Sixth ISCA Workshop on Speech Synthesis
Bonn, Germany
August 22-24, 2007

Workshop Proceedings

Edited by Petra Wagner, Julia Abresch, Stefan Breuer, and Wolfgang Hess

Rheinische Friedrich-Wilhelms-Universität Bonn
Institut für Kommunikationswissenschaften
Abteilung Sprache und Kommunikation
Poppelsdorfer Allee 47, D-53115 Bonn

Sixth ISCA Workshop on Speech Synthesis
Bonn, Germany
August 22-24, 2007

Table of Contents

Keynote 1 (Wednesday, August 22, 10:10)

Various Topics (Wednesday, August 22, 11:00)

Expressive Speech Synthesis (Wednesday, August 22, 14:00)

Poster Session 1 (Wednesday, August 22, 16:05)

Voice Conversion (Thursday, August 23, 9:00)

Speech Synthesis by HMM (Thursday, August 23, 11:00)

Tone and Tone Accent Languages (Thursday, August 23, 14:00)

Poster Session 2 (Thursday, August 23, 915:40)

Prosody Modelling (Thursday, August 23, 16:50)

Inventory Construction (Friday, August 24, 9:00)

Keynote 2 (Friday, August 24, 11:00)

Applications (Friday, August 24, 11:50)

Systems (Friday, August 24, 14:00)

Poster Session 3 (Friday, August 24, 15:40)

Evaluation (Friday, August 24, 16:50)

Sixth ISCA Workshop on Speech Synthesis
Bonn, Germany
August 22-24, 2007

Workshop Program at a Glance

Wednesday, August 22, 2007

10:00-10:10	Opening
10:10-11:00	Keynote 1: Articulatory Synthesis
11:00-12:40	Oral Session 1: Various Topics
12:40-14:00	Lunch Break
14:00-16:05	Oral Session 2: Expressive Speech Synthesis
16:05-17:15	Poster Session 1 and Coffee Break
18:00	Boat Trip on the Rhine River

Thursday, August 23, 2007

09:00-10:40	Oral Session 3: Voice Conversion
10:10-11:00	Coffee Break
11:00-12:40	Oral Session 4: Speech Synthesis by HMM
12:40-14:00	Lunch Break
14:00-15:40	Oral Session 5: Tone and Tone Accent Languages
15:40-16:50	Poster Session 2 and Coffee Break
16:50-18:30	Oral Session 6: Prosody Modelling

Friday, August 24, 2007

09:00-10:40	Oral Session 7: Inventory Construction
10:10-11:00	Coffee Break
11:00-11:50	Keynote 2: The Blizzard Challenge
11:50-12:40	Oral Session 8: Applications
12:40-14:00	Lunch Break
14:00-15:40	Oral Session 9: Systems
15:40-16:50	Poster Session 2 and Coffee Break
16:50-18:05	Oral Session 10: Evaluation
18:05-18:15	Closing

Sixth ISCA Workshop on Speech Synthesis
Bonn, Germany
August 22-24, 2007

Welcome Address by the Chairman of SSW6, Prof. Wolfgang Hess, Bonn, Germany

Welcome to the Sixth ISCA Workshop on Speech Synthesis!

This series of workshops has now been held over the past 17 years. It started with the workshop in Autrans, France (1990), followed by the ones at Mohonk, NY, USA (1994), Jenolan Caves, Australia (1998), Pitlochry, Scotland, UK (2001), and Pittsburgh, PA, USA (2004).

Jonathan Allen once said: “Speech synthesis is not a big problem, but a large collection of small detail problems”. It is therefore not astonishing that many of the more recent developments in speech synthesis, such as nonuniform unit selection, trainable systems, or articulatory synthesis, are based on long-standing ideas and concepts.

The quality of spoken output again has substantially improved over the last years, yet it still has its limitations with respect to naturalness and even with respect to intelligibility and comprehensibility. In 2005, the Blizzard Challenge was started to establish a common platform for comparing and evaluating corpus-based synthesis systems. We will hear more about its outcome at the workshop.

The last years have also seen a rising interest in parametric methods, in particular in connection with trainable systems, such as hidden Markov models. They are extremely flexible with respect to voice variation; they need little training data to adapt to a new speaker or a different voice quality, and are thus good research tools in voice conversion or expressive speech synthesis. So these systems provide the flexibility that is missing in methods directly based on natural speech signals. Another principle to be mentioned is articulatory synthesis whose quality has also been greatly improved.

Let me also welcome you to the University of Bonn which will celebrate its 200th birthday 11 years from now. With more than 25000 students and 7 faculties it is one of Germany’s major universities. It covers a wide range of areas from the humanities to natural and life sciences, from law to medicine, from economy to agriculture, from mathematics and computer science to language, speech, literature, and culture.

Again, a warm welcome to SSW6, and enjoy your stay at this workshop and at the University of Bonn!

On behalf of the Organizing Committee,
Wolfgang Hess
Chairman, SSW6

Welcome Address by the President of SynSIG, Prof. Thierry Dutoit, Mons, Belgium

At an international conference on speech processing, a speech scientist once held up a tube of toothpaste (whose brand was ”Signal”) and, squeezing it in front of the audience, coined the phrase ”This is speech synthesis; speech recognition is the art of pushing the toothpaste back into the tube.”

One could turn this very simplistic view the other way round: users are generally much more tolerant of speech recognition errors than they are willing to listen to unnatural speech. There is magic in a speech recognizer that transcribes continuous radio speech into text with a word accuracy as low as 50%; in contrast, even a perfectly intelligible speech synthesizer is only moderately tolerated by users if it delivers nothing more than ”robot voices”. Delivering both intelligibility and naturalness has been the holy grail of speech synthesis research for the past 30 years. More recently, expressivity has been added as an objective of speech synthesis.

Add to this the engineering costs (computational cost, memory cost, design cost for making another synthetic voice or another language) which have to be taken into account, and you’ll start to have an idea of the challenges underlying text-to-speech synthesis.

The ISCA Special Interest Group on Speech Synthesis (or SynSIG) was created in 1998 to help face these challenges with the best chances of success. It promotes activities related to advancing the science of speech synthesis, including: white papers, surveys, special issues in international journals, the development of a speech synthesis portal (http://www.synsig.org), promotion of the annual Blizzard Challenge speech synthesis evaluations, and ... Speech Synthesis Workshops (SSWs), now held every three years. SSWs are a unique occasion for meeting each other. They contribute to establishing a feeling that we are all participating in a joint effort towards intelligible, natural, and expressive synthetic speech.

If you are not a member of SynSIG yet, visit our web page and join our (moderated, low traffic) mailing list!

Enjoy SSW6!

On behalf on the SynSIG board,
Thierry Dutoit, President

Welcome Address by the President of ISCA, Prof. Julia B. Hirschberg, New York, NY, USA

As president of ISCA (and one of the organizers of the Second ESCA/IEEE Workshop on Speech Synthesis held in 1994) I warmly welcome you the sixth workshop in one of ISCA’s most successful and enduring workshop series. Since the first ESCA Workshop on Speech Synthesis held in Autrans in 1990 and organized by Christian Benoît and Gérard Bailly, these meetings have played an invaluable role in keeping the members of the Speech Synthesis community in touch with one another and in encouraging innovation in both science and technology. On behalf of the ISCA Board, I thank Wolfgang Hess, Workshop Chair, and the members of his team at the Institute of Communication Sciences (IfK) in the University of Bonn, and also the ISCA Special Interest Group on Speech Synthesis (SynSIG) for organizing this workshop. I have no doubt that it will be a tremendous success.

Julia Hirschberg
President,
International Speech Communication Association (ISCA)

Sixth ISCA Workshop on Speech Synthesis
Bonn, Germany
August 22-24, 2007

Acknowledgements

General Chair

Wolfgang Hess, Bonn, Germany

Program Chair

Petra Wagner and Wolfgang Hess, Bonn, Germany

Scientific and Review Committee

Alan Black, Pittsburgh, PA, USA
Alexander Kain, Beaverton, OR, USA
Alistair Conkie, AT&T Labs, USA
Andrew Breen, Nuance Communications, UK
Ann Syrdal, AT&T Labs, USA
Antje Schweizer, Stuttgart, Germany
Antonio Bonafonte, Barcelona, Spain
Arthur Toth, Pittsburgh, PA, USA
B. Yegnanarayama, Madras, India
Baris Bozkurt, Izmir, Turkey
Bernd Möbius, Stuttgart, Germany
Christophe d'Alessandro, Orsay, France
David Talkin, Google, USA
Esther Klabbers, Beaverton, OR, USA
Gérard Bailly, Grenoble, France
Grazyna Demenko, Poznan, Poland
Harald Höge, Munich, Germany
John Kominek, Pittsburgh, PA, USA
Julia Abresch, Bonn, Germany
Junichi Yamagishi, Edinburgh, UK
Keiichi Tokuda, Nagoya, Japan
Korin Richmond, Edinburgh, UK
Louis Pols, Amsterdam, The Netherlands
Marc Schröder, DFKI, Germany
Martine Grice, Cologne, Germany
Matthew Aylett, Edinburgh, UK
Nick Campbell, ATR, Kyoto, Japan
Noam Amir, Tel Aviv, Israel
Oliver Jokisch, Dresden, Germany
Peter Birkholz, Rostock, Germany
Petra Wagner, Bonn, Germany
Ralf Benzmüller, Bochum, Germany
Raul Fernandez, Yorktown Heights, NY, USA
Rüdiger Hoffmann, Dresden, Germany
Simon King, Edinburgh, UK
Stefan Breuer, Bonn, Germany
Thierry Dutoit, Mons, Belgium
Tomoki Toda, Nara, Japan
Vincent Pagel, Acapela Group, Belgium
Volker Strom, Edinburgh, UK
Wolfgang Hess, Bonn, Germany

Local Organization

Julia Abresch
Stefan Breuer
Wolfgang Hess
Gisela von Neffe
Christoph Reinhard
Petra Wagner

Web Master

Mareike Ahrens, Bonn, Germany

Proceedings

Julia Abresch
Stefan Breuer
Wolfgang Hess
Petra Wagner

Photos

Frank Luerweg, Ulrike E. Klopp, Bonn, Germany

SSW6 Logo

Nicolai Sandow, Bonn, Germany

Cover

department digital, Bonn, Germany

Sixth ISCA Workshop on Speech Synthesis
Bonn, Germany
August 22-24, 2007

Abstracts

Keynote Session 1

Wednesday, August 22, 10:10-11:00

Perspectives for Articulatory Speech Synthesis

Bernd J. Kröger

Department of Phoniatrics, Pedaudiology, and Communication Disorders, Medical Faculty of Aachen University, Aachen, Germany

Articulatory speech synthesis currently has two perspectives. (i) Technical perspective: Due to progress in common computer hardware (general increase in computation rate) and software (usability of compilers and simulation software) it is now possible to develop comprehensive phonetic models of speech production reaching nearly real-time for the calculation of acoustic speech signals. Furthermore the phonetic knowledge increased to a degree that these production models now are capable of accomplishing a good up to high acoustic quality. Limitations are mainly the control modules. In this paper we argue for a self-learning input dependent gestural control model for articulatory speech synthesis. (ii) Theoretical perspective: A comprehensive articulatory speech synthesis system capable of producing high quality acoustic output necessarily incorporates a lot of knowledge on all phonetic aspects of speech production: articulatory sound targets, typical articulatory movement strategies for realizing sounds or syllables (e.g. coarticulation), a general concept for temporal coordination of speech relevant articulatory movements (i.e. speech gestures) etc. In this paper an example for such a system will be given and a suggestion for the still open question on strategies for control concepts for high-quality articulatory speech synthesis will be proposed.

Pages: 391-391

Session: Various Topics

Wednesday, August 22, 11:00-12:40

Learning Optimal Audiovisual Phasing for an HMM-based Control Model for Facial Animation

Oxana Govokhina (1,2), Gérard Bailly (1), Gaspard Breton (2)

(1) GIPSA-Lab Dept. Speech & Cognition, CNRS/INPG/UJF & Univ. Stendhal, Grenoble, France
(2) France Telecom R&D, Cesson-Sévigné, France

We propose here an HMM-based trajectory formation system that predicts articulatory trajectories of a talking face from phonetic input. In order to add flexibility to the acoustic/ gestural alignment and take into account anticipatory gestures, a phasing model has been developed that predicts the delays between the acoustic boundaries of allophones to be synthesized and the gestural boundaries of HMM triphones. The HMM triphones and the phasing model are trained simultaneously using an iterative analysis-synthesis loop. Convergence is obtained within a few iterations. We demonstrate here that the phasing model improves significantly the prediction error and captures subtle context-dependent anticipatory phenomena.

Pages: 1-4

Control Concepts for Articulatory Speech Synthesis

Peter Birkholz (1), Ingmar Steiner (2), Stefan Breuer (3)

(1) Institute for Computer Science, University of Rostock, Germany
(2) Department of Computational Linguistics and Phonetics, Saarland University, Germany
(3)Institute of Communication Sciences (IfK), University of Bonn, Germany

We present two concepts for the generation of gestural scores to control an articulatory speech synthesizer. Gestural scores are the common input to the synthesizer and constitute an organized pattern of articulatory gestures. The first concept generates the gestures for an utterance using the phonetic transcriptions, phone durations, and intonation commands predicted by the Bonn Open Synthesis System (BOSS) from an arbitrary input text. This concept extends the synthesizer to a text-to-speech synthesis system. The idea of the second concept is to use timing information extracted from Electromagnetic Articulography signals to generate the articulatory gestures. Therefore, it is a concept for the re-synthesis of natural utterances. Finally, application prospects for the presented synthesizer are discussed.

Pages: 5-10

Spectral Control in Concatenative Speech Synthesis

Alexander B. Kain, Qi Miao, Jan P. H. van Santen

Center for Spoken Language Understanding (CSLU), OGI School of Science & Engineering, Oregon Health & Science University (OHSU), Beaverton, OR, USA

We report on research in which we increased the degree of spectral control in concatenative synthesis by controlling the formant frequencies of the synthetic speech, as well as the energies in four spectral bands. In addition, we eliminated ”points” of concatenation in favor of ”regions” of concatenation, by *cross-fading* between the end and the beginning of two speech segments that are part of a concatenation operation. We hypothesized that these approaches would decrease the frequency and severity of audible discontinuities in the synthetic speech and thus also increase the perceived quality of the speech. A listening test determined that stimuli created with the proposed methods resulted in significantly increased quality.

Pages: 11-16

Feature Transformation Applied to the Detection of Discontinuities in Concatenated Speech

Barry Kirkpatrick, Darragh O'Brien, Ronán Scaife

Speech Group, Research Institute for Networks and Communications Engineering, Faculty of Engineering and Computing, Dublin City University, Dublin, Ireland

The quality of concatenated speech depends on the degree of mismatch between successive units. Defining a perceptually salient join cost to represent the degree of mismatch has proven to be a difficult task. Such a join cost is critical in unit selection synthesis to ensure that the optimum sequence of speech units is selected from the units available in the speech inventory. In this study the problem of defining a join cost is extended to include a feature transformation stage. Two feature transformations are considered, principal component analysis and a neural networkbased approach. Each transformation was investigated for its ability to improve the detection of discontinuities in concatenated speech for a given feature set. The results indicate that a feature transformation combining principal component analysis as a preprocessing stage to a neural network-based transformation can increase the rate of detection of discontinuities. The neural network was trained using perceptual data obtained from a subjective listening test indicating if a join is continuous or discontinuous. The highest scoring measure based on this strategy provided a correlation with perceptual results of 0.8859 compared with a value of 0.7576 over the baseline MFCC measure on the same test data set.

Pages: 17-21

Session: Expressive Speech Synthesis

Wednesday, August 22, 14:00-16:05

Towards Conversational Speech Synthesis: Lessons Learned from the Expressive Speech Processing Project

Nick Campbell

NiCT/ATR-SLC, National Institute of Information and Comunications Technology & ATR Spoken Language Communication Research Labs, Keihanna Science City, Kyoto, Japan

This paper discusses some ideas for the requirements and methods of conversational speech synthesis, based on experience gained from the collection and analysis of a very large corpus of conversational speech in a variety of real-life everyday contexts. It shows that because variation in voice quality plays a significant part in the transmission of interpersonal and affect-related social information, this feature should be given priority in future speech synthesis research. Several solutions to this problem are proposed.

Pages: 22-27

Communicative Speech Synthesis with XIMERA: A First Step

Shinsuke Sakai (1,2), Jinfu Ni (1,2),
Ranniery Maia (1,2), Keiichi Tokuda (1,3),
Minoru Tsuzaki (1,4), Tomoki Toda (1,5),
Hisashi Kawai (2,6), Satoshi Nakamura (1,2)

(1) National Inst. of Inform. and Comm. Tech. (NiCT), Japan
(2) ATR Spoken Language Comm. Labs, Japan
(3) Nagoya Institute of Technology, Japan
(4) Kyoto City University of Arts, Japan
(5) Nara Institute of Science and Technology, Japan
(6) KDDI Research and Development Labs, Japan

This paper presents a corpus-based approach to communicative speech synthesis. We chose ”good news” style and ”bad news” style for our initial attempt to synthesize speech that has appropriate expressiveness desired in human-human or human-machine dialog. We utilized 10-hour ”neutral” style speech corpus as well as smaller corpora with good news and bad news styles, each consisting of two to three hours of speech from the same speaker. We trained target HMM models with each style and synthesized speech with unit databases containing speech with the relevant style as well as neutral speech. From the listening tests, we found out that intended communicative styles were comprehended by listeners and that considerably high mean opinion score on naturalness was achieved with rather small, style-specific corpora.

Pages: 28-33

Automatic Exploration of Corpus-Specific Properties for Expressive Text-to-Speech: A Case Study in Emphasis

Raul Fernandez, Bhuvana Ramabhadran

IBM TJ Watson Research Center, Yorktown Heights, NY 10598, USA

In this paper we explore an approach to expressive text-tospeech synthesis in which pre-existing expression-specific corpora are complemented with automatically generated labels to augment the search space of units the engine can exploit to increase its expressiveness. We motivate this data-discovery approach as an alternative to an approach guided by data collection, in order to harness the full usefulness of the expressiveness already contained in a synthesis corpus. We illustrate the approach with a case study that uses emphasis as its intended expression, describe algorithms for the automatic discovery of such instances in the database and how to make use of them during synthesis, and, finally, evaluate the benefits of the proposal to demonstrate the feasibility of the approach.

Pages: 34-39

Modeling and Perceiving of (Un-)Certainty in Articulatory Speech Synthesis

Charlotte Wollermann (1), Eva Lasarcyk (2)

(1) Institute of Communication Sciences, University of Bonn, Germany
(2) Institute of Phonetics, Saarland University, Germany

This paper deals with the role of paralinguistic expression in articulatory speech synthesis. We describe two experiments which investigate the perception of certain vs. uncertain utterances producrd by articulatory speech synthesis, using the system developed in [1].

Experiment 1 tests to what extent subjects are able to identify certainty and uncertainty as intended paralinguistic expressions in the acoustical signal by the varying acoustic cues intonation and delay. Further on, we investigate if (un)certainty influences the intelligibility of the synthetic utterances. Results show that the utterances are identified as intended with respect to (un)certainty. Regarding intelligibility, hardly any influence is measurable.

Experiment 2 looks more in detail into the perception of uncertainty by using several levels. Therefore, not only intonation and delay are varied as acoustical cues but also fillers. Results show that our intended different levels of uncertainty indeed evoked different degrees of perceived uncertainty.

Reference. [1] Birkholz, P. (2005). *3-D Artikulatorische Sprachsynthese* (Logos, Berlin)

Pages: 40-45

Perceptual Annotation of Expressive Speech

Lijuan Wang (1), Min Chu (1), Yaya Peng (2), Yong Zhao (1), Frank K. Soong (1)

(1) Microsoft Research Asia, Beijing, China
(2) Department of Linguistics & Modern Languages, The Chinese University of Hong Kong, China

A six-dimensioned label set for annotating expressiveness of speech samples is proposed. Unlike conventional emotional annotation labels that require annotators to make rather difficult judgments on speakers' emotional (high-level) status, the new annotation set of six low-level labels, i.e., ”pitch”, ”vocal effort”, ”voice age”, ”loudness”, ”speaking rate”, and ”speaking manner” can be more easily labeled by non-experts. 800 expressive utterances were annotated by four annotators with the proposed labels. The labeling also shows a good consistency (71%) among the annotators. The proposed six labels capture the different styles (expressiveness) well in the audio-book. The difference between styles, measured by the intensity of styles along the six labels, is highly correlated (0.85) with the perceptual distance obtained from a subjective AB test. A compact classification and regression tree (CART) is built to automatically group sentences of similar expressiveness into several ”pure” speaking styles. The interpretation of each speaking style can be explicitly understood from the CART structure.

Pages: 46-51

Poster Session 1

Wednesday, August 22, 16:05-17:15

Joint Analysis of Speech Frames for Synthesis Based on Lossy Tube Models

Karl Schnell, Arild Lacroix

Institute of Applied Physics, Goethe-University Frankfurt, Germany

This paper discusses a model-based synthesis approach focused on the estimation of model parameters. For the treated approach, tube models are used for analysis and synthesis of speech units. In comparison to the standard lossless tube model, an extended tube model is used which includes the frequency dependent vocal tract losses. The parameters of the tube models are estimated by minimizing the spectral error between the tube model and a speech segment. For the analysis of speech units, the time evolution of the parameters is taken into account. For that purpose, the speech segments are analyzed jointly which ensures smooth parameter trajectories. The investigations show that, especially for extended tube models, the joint analysis of frames improves the quality of the synthesized speech signals. Additionally, the differences of the results obtained by the standard and the extended tube model are discussed.

Pages: 52-57

Are Rule-based Syllabification Methods Adequate for Languages with Low Syllabic Complexity? The Case of Italian

Connie R. Adsett (1), Yannick Marchand (2)

(1) Institute for Biodiagnostics (Atlantic), National Research Council Canada, Halifax, Nova Scotia, Canada
(2) Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada

Syllabification information is a valuable component in speech synthesis systems. Linguistic rule-based methods have been assumed to be the best technique for determining the syllabification of unknown words. This has recently been shown to be incorrect for the English language where data-driven algorithms have been shown to outperform rule-based methods. It may be possible, however, that data-driven methods are only better for languages with complex syllable structures. In this paper, three rule-based automatic syllabification systems are compared and two data-driven (Syllabification by Analogy and the Look-Up Procedure) on a language with lower syllabic complexity - Italian. Using a leave-one-out procedure on 44,720 words, the best data-driven algorithm (Syllabification by Analogy) achieved 97.70% word accuracy while the best rule-based method correctly syllabified 89.77% words. These results show that data-driven methods can also outperform rule-based methods on Italian syllabification, indicating that these may be the best approaches to the syllabification component of speech synthesis systems.

Pages: 58-63

Spoken Language Conversion with Accent Morphing

Mark Huckvale, Kayoko Yanagisawa

Department of Phonetics and Linguistics, University College London, UK

Spoken language conversion is the challenge of using synthesis systems to generate utterances in the voice of a speaker but in a language unknown to the speaker. Previous approaches have been based on voice conversion and voice adaptation technologies applied to the output of a foreign language TTS system. This inevitably reduces the quality and intelligibility of the output, since the source speaker will not be a good source of phonetic material in the new language. This article contrasts previous work with a new approach that uses two synthesis systems: one in the source speaker's voice, one in the voice of a native speaker of the target language. Audio morphing technology is then exploited to correct the foreign accent of the source speaker, while at the same time trying to maintain his or her identity. In this paper we construct a spoken language conversion system using accent morphing and evaluate its performance in terms of intelligibility. Encouraging results tell us more about the challenges of spoken language conversion.

Pages: 64-70

Comparative Investigation of Peak Alignment in Polish and German Unit Selection Corpora

Grazyna Demenko (1), Agnieszka Wagner (1), Matthias Jilka (2), Bernd Möbius (3)

(1) Dept. of Linguistics, Adam Mickiewicz University, Poznan, Poland
(2) Dept. of English Linguistics, University of Stuttgart, Germany
(3) Institute of Natural Language Processing, University of Stuttgart, Germany

This paper presents a comparative study on the temporal alignment of pitch peaks of H*L accents in Polish and German. Speech material used in the study came from the unit selection synthesis corpora of the Polish voice module of the BOSS system and the IMS German Festival TTS system. The major factors investigated were concerned with the influence of syllable structure on the one hand, as well as phrasal and tonal environment on the other hand. For the analysis of Polish falling accents, the effects of accent type, phrase type, and word position were also taken into account. Results show that in both languages, pitch peak placement is consistently affected by onset and coda type and by the tonal context (H or

L tonal target preceding or following). Also, the position of the accent in the phrase is found to have a significant influence. Additionally, the results also reveal the difference between the two Polish falling pitch accents (static and dynamic).

Pages: 71-76

Optimization of Polish Segmental Duration Prediction with CART

Katarzyna Klessa (1), Marcin Szymanski (2),
Stefan Breuer (3), Grazyna Demenko (1)

(1) Institute of Linguistics, Dept. of Phonetics, Adam Mickiewicz University, Poznan, Poland
(2) Poznan University of Technology, Poland
(3) Institute of Communication Sciences, University of Bonn, Germany

This paper describes results of the investigation of Polish segmental duration for the purpose of speech synthesis. The experiment is a continuation of the previous work of the same authors [1] aiming at improving the outcome of the duration prediction mechanism to enhance the overall quality of synthesized speech.

Reference. [1] Breuer, S., Francuzik, K., Demenko, G., Szymanski, M. (2006), Analysis of Polish Duration with CART, *Proceedings of Speech Prosody, Dresden*

Pages: 77-80

Utilization of an HMM-Based Feature Generation Module in 5-ms-Segment Concatenative Speech Synthesis

Toshio Hirai (1) Junichi Yamagishi (2)
Seiichi Tenpaku (2)

(1) Arcadia, Inc., Osaka, Japan
(2) The Centre for Speech Technology Research, University of Edinburgh, UK

If a concatenative speech synthesis system uses more short speech segments, it increases the potential to generate natural speech because the concatenation variation becomes greater. Recently, a synthesis approach was proposed in which very short (5 ms) segments are used. In this paper, an implementation of an HMM-based feature generation module into a very short segment concatenative synthesis system that has the advantage of modularity and a synthesis experiment are described.

Pages: 81-84

Clustering Algorithm for F_0 Curves Based on Hidden Markov Models

Damien Lolive, Nelly Barbot, Olivier Boeffard

IRISA / University of Rennes 1 - ENSSAT, Lannion, France

This article describes a new unsupervised methodology to learn F_0 classes using HMM on a syllable basis. A F_0 class is represented by a HMM with three emitting states. The unsupervised clustering algorithm relies on an iterative gaussian splitting and EM retraining process. First, a single class is learnt on a training corpus (8000 syllables) and it is then divided by perturbing gaussian means of successive levels. At each step, the mean RMS error is evaluated on a validation corpus (3000 syllables). The algorithm stops automatically when the error becomes stable or increases. The syllabic structure of a sentence is the reference level we have taken for F_0 modelling even if the methodology can be applied to other structures. Clustering quality is evaluated in terms of cross-validation using a mean of RMS errors between F_0 contours on a test corpus and the estimated HMM trajectories. The results show a pretty good quality of the classes (mean RMS error around 4Hz).

Pages: 85-89

Building a Better Indian English Voice Using ”More Data”

Rohit Kumar, Rashmi Gangadharaiah, Sharath Rao,
Kishore Prahallad, Carolyn P. Rosé, Alan W. Black

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

We report our experiments towards improving an existing publicly available Indian English voice using additional data. The additional data was used to create new duration and pronunciation models as well as to convert the existing voice to create a more Indian sounding voice. Two experiments along the above lines are reported. In the first experiment, we found that changing the pronunciation models has the potential to improve an existing Indian English voice. We conducted a second experiment to validate this finding. The second experiment shows the potential value in carefully investigating the separate effects of the different components of a pronunciation model in order to understand their unique contributions to improving an Indian English voice.

Pages: 90-94

Creating German Unit Selection Voices for the MARY TTS Platform from the BITS Corpora

Marc Schröder, Anna Hunecke

DFKI GmbH, Saarbrücken, Germany

The present paper reports on the creation of German unit selection voices from corpora which had been recorded and annotated previously in the BITS project. We describe the unit selection mechanism of our MARY TTS platform, as well as the tools for creating a synthesis voice from a speech corpus, and their application to the creation of German unit selection voices from the BITS corpora. Because of reservations concerning the mismatch of phonetic chains predicted by the German TTS components in MARY and the manually corrected database labels, we compared voices based on the manually corrected labels with voices based on automatic forced alignment labelling. We compute the diphone coverage for both types of voices and show that it is a reasonable approximation of the German diphone set. A preliminary evaluation confirms the expectations: while the manually corrected versions show a higher segmental accuracy, the automatically labelled versions sound more fluent.

Pages: 95-100

Session: Voice Conversion

Thursday, August 23, 9:00-10:40

Regression Approaches to Voice Quality Control Based on One-to-Many Eigenvoice Conversion

Kumi Ohta, Yamato Ohtani, Tomoki Toda,
Hiroshi Saruwatari, Kiyohiro Shikano

Graduate School of Information Science, Nara Institute of Science and Technology, Japan

This paper proposes techniques for flexibly controlling voice quality of converted speech from a particular source speaker based on one-to-many eigenvoice conversion (EVC). EVC realizes a voice quality control based on the manipulation of a small number of parameters, i.e., weights for eigenvectors, of an eigenvoice Gaussian mixture model (EV-GMM), which

is trained with multiple parallel data sets consisting of a single source speaker and many pre-stored target speakers. However, it is difficult to control intuitively the desired voice quality with those parameters because each eigenvector doesn't usually represent a specific physical meaning. In order to cope with this problem, we propose regression approaches to the EVC-based voice quality controller. The tractable voice quality control of the converted speech is achieved with a low-dimensional voice quality control vector capturing specific voice characteristics. We conducted experimental verifications of each of the proposed approaches.

Pages: 101-106

An Evaluation of Many-to-One Voice Conversion Algorithms with Pre-Stored Speaker Data Sets

Daisuke Tani, Yamato Ohtani, Tomoki Toda,
Hiroshi Saruwatari, Kiyohiro Shikano

Graduate School of Information Science, Nara Institute of Science and Technology, Japan

This paper describes an evaluation of many-to-one voice conversion (VC) algorithms converting an arbitrary speaker's voice into a particular target speaker's voice. These algorithms effectively generate a conversion model for a new source speaker using multiple parallel data sets of many pre-stored source speakers and the single target speaker. We conducted experimental evaluations for demonstrating the conversion performance of each of the many-to-one VC algorithms, including not only the conventional algorithms based on a speaker independent GMM and on eigenvoice conversion (EVC), but also new algorithms based on speaker selection and on EVC with speaker adaptive training (SAT). As a result, it is shown that an adaptation process of the conversion model improves significantly conversion performance, and the algorithm based on speaker selection works well even when using a very limited amount of adaptation data.

Pages: 107-112

Towards an Improved Modeling of the Glottal Source in Statistical Parametric Speech Synthesis

Joao P. Cabral, Steve Renals, Korin Richmond,
Junichi Yamagishi

The Centre for Speech Technology Research, University of Edinburgh, UK

This paper proposes the use of the Liljencrants-Fant model (LFmodel) to represent the glottal source signal in HMM-based speech synthesis systems. These systems generally use a pulse train to model the periodicity of the excitation signal of voiced speech. However, this model produces a strong and uniform harmonic structure throughout the spectrum of the excitation which makes the synthetic speech sound buzzy. The use of a mixed band excitation and phase manipulation reduces this effect but it can result in degradation of the speech quality if the noise component is not weighted carefully. In turn, the LFwaveform has a decaying spectrum at higher frequencies, which is more similar to the real glottal source excitation signal.

We conducted a perceptual experiment to test the hypothesis that the LF-model can perform as well as or better than the pulse train in a HMM-based speech synthesizer. In the synthesis, we used the mean values of the LF-parameters, calculated by measurements of the recorded speech. The result of this study is important not only regarding the improvement in speech quality of these type of systems, but also because the LF-model can be used to model many characteristics of the glottal source, such as voice quality, which are important for voice transformation and generation of expressive speech.

Pages: 113-118

GMM-based Speech Transformation Systems under Data Reduction

Larbi Mesbahi, Vincent Barreaud, Olivier Boeffard

IRISA / University of Rennes 1 - ENSSAT, Lannion, France

The purpose of this paper is to study the behavior of voice conversion systems based on Gaussian mixture model (GMM) when reducing the size of the training data corpus. Our first objective is to locate the threshold of degradation on the training corpus from which the error of conversion becomes too important. Secondly, we seek to observe the behavior of these conversion systems with regard to this threshold, in order to establish a relation between the size of training data corpus and the complexity of each method of transformation. We observed that the threshold is beyond 50 sentences (ARCTIC corpus), whatever the conversion system. For this corpus, the conversion error of the best approach increases only by 1.77 % compared to the complete training corpus which contains 210 utterances.

Pages: 119-124

Session: Speech Synthesis by HMM

Thursday, August 23, 11:00-12:40

Improved Average-Voice-based Speech Synthesis using Gender-Mixed Modeling and a Parameter Generation Algorithm considering GV

Junichi Yamagishi (1), Takao Kobayashi (2),
Steve Renals (1), Simon King (1), Heiga Zen (3),
Tomoki Toda (4), Keiichi Tokuda (3)

(1) University of Edinburgh, UK; (2) Tokyo Institute of Technology, Japan; (3) Nagoya Institute of Technology, Japan; (4) Nara Institute of Science and Technology, Japan

For constructing a speech synthesis system which can achieve diverse voices, we have been developing a speaker independent approach of HMM-based speech synthesis in which statistical average voice models are adapted to a target speaker using a small amount of speech data. In this paper, we incorporate a high-quality speech vocoding method STRAIGHT and a parameter generation algorithm with global variance into the system for improving quality of synthetic speech. Furthermore, we introduce a feature-space speaker adaptive training algorithm and a gender mixed modeling technique for conducting further normalization of the average voice model. We build an English text-to-speech system using these techniques and show the performance of the system.

Pages: 125-130

An Excitation Model for HMM-Based Speech Synthesis Based on Residual Modeling

Ranniery Maia (1), Tomoki Toda (1,2),
Heiga Zen (3), Yoshihiko Nankaku (3),
Keiichi Tokuda (1,3)

(1) National Inst. of Inform. and Comm. Tech. (NiCT) / ATR Spoken Language Comm. Labs, Japan
(2) Nara Institute of Science and Technology, Japan
(3) Nagoya Institute of Technology, Japan

This paper describes a trainable excitation approach to eliminate the unnaturalness of HMM-based speech synthesizers.

During the waveform generation part, mixed excitation is constructed by state-dependent filtering of pulse trains and white noise sequences. In the training part, filters and pulse trains are jointly optimized through a procedure which resembles analysis-bysynthesis speech coding algorithms, where likelihood maximization of residual signals (derived from the same database which is used to train the HMM-based synthesizer) is pursued. Preliminary results show that the novel excitation model in question eliminates the unnaturalness of synthesized speech, being comparable in quality to the the best approaches thus far reported to eradicate the buzziness of HMM-based synthesizers.

Pages: 131-136

An HMM-based Bilingual (Mandarin-English) TTS

Hui Liang (1), Yao Qian (2) Frank K. Soong (2)

(1) School of Information Security Engineering, Shanghai Jiaotong University, China
(2) Microsoft Research Asia, Beijing, China

We propose to build an HMM-based, Mandarin and English, bilingual TTS system. Starting with a simple baseline of two TTS systems built separately from Mandarin and English databases recorded by the same speaker, we construct a new, mixed-language TTS by designing language specific and independent questions to facilitate phone sharing across the two languages. With shared phones, the new system has a smaller footprint than the baseline system. The synthesis quality is either the same for non-mixed, Mandarin or English synthesis as the baseline or much better for mixed-language synthesis. The higher quality of mixed-language synthesis is confirmed by preference scores of 59.5% vs 40.5%, obtained in a subjective listening test. A preliminary Mandarin synthesis experiment was also performed by using the model parameters in the leaf nodes of English decision tree where Kullback-Leibler divergence is used to establish the nearest neighbor based mapping between leaf nodes in the decision trees of the two languages. A subjective transcription test shows a character accuracy of 93.9%.

Pages: 137-142

Data-Driven Approach to Rapid Prototyping Xhosa Speech Synthesis

Justus C. Roux, Albert S. Visagie

Centre for Language and Speech Technology, Stellenbosch University, South Africa

This paper presents work in progress towards building a Xhosa speech synthesizer. HTS is being used for this purpose due to certain desirable properties. As a minority language, linguistic resources for Xhosa are limited despite a variety of impressionistic phonetic studies, prompting a minimalist approach and a preference for data-driven methods. Xhosa is an agglutinative language, and is also held to be a tonal language, which therefore requires morphological analysis and tonal information in order to generate intelligible speech. By taking into account more recent findings on the nature of Xhosa prosody, it appears that a minimalist approach that excludes tone information is possible. We implement the system using HTS. Such a data-driven TTS system is a useful tool to test various syntactic and other features in text that influence Xhosa prosody.

Pages: 143-147

Session: Tone and Tone Accent Languages

Thursday, August 23, 14:00-15:40

CRF-based Statistical Learning of Japanese Accent Sandhi for Developing Japanese Text-to-Speech Synthesis Systems

Nobuaki Minematsu (1), Ryo Kuroiwa (2), Keikichi Hirose (2) Michiko Watanabe (1)

(1) Graduate School of Frontier Sciences; (2) Graduate School of Information Science and Technology; University of Tokyo, Japan

In Japanese, every content word has its own H/L pitch pattern when it is uttered isolatedly, called accent type. In a TTS system, this lexical information is usually stored in a dictionary and it is referred to for prosody generation. When converting a written sentence to speech, however, this lexical H/L pattern is often changed according to the context, known as word accent sandhi. This accent change is troublesome for speech synthesis researchers because it is difficult even for native speakers to describe explicitly what kind of mechanism is working for the change although young Japanese learn the mechanism without trouble. For developing a good Japanese TTS system, this implicit and phonological knowledge has to be built in the system. In our previous study [1], we developed a rule-based module for the accent sandhi but it is true that it produced an unignorable number of errors. In this paper, the development of a corpusbased module is described using Conditional Random Fields (CRFs) to predict the change. Although the new module shows the better performance for the prediction than the previous rulebased module, the new module is tuned further by integrating the rule-based knowledge acquired in the previous study.

Reference. [1] N. Minematsu, R. Kita, and K. Hirose (2003), ”Automatic estimation of accentual attribute values of words for accent sandhi rules of Japanese text-to-speech conversion,” Trans. IEICE, vol. E86-D, no.3, pp.550-557

Pages: 148-153

Two-Step Generation of Mandarin F_0 Contours Based on Tone Nucleus and Superpositional Models

Qinghua Sun (1), Keikichi Hirose (2), Nobuaki Minematsu (3)

(1) Graduate School of Engineering; (2) Graduate School of Information Science and Technology; (3) Graduate School of Frontier Sciences; University of Tokyo, Japan

A 2-step scheme was developed in our method for synthesizing sentence fundamental frequency (F_0) contours of Mandarin speech. The method is based on representing a sentence logarithmic F_0 contour as a superposition of tone components on phrase components as in the case of generation process model (F_0 model). The tone components are realized by concatenating tone nucleus F_0 patterns generated by a corpus-based method, while the phrase components are generated by rules under the F_0 model framework. In the 2-step scheme, the phrase components are first generated and their information is added to the inputs for the prediction of tone nucleus F_0 patterns. Result of listening tests on synthetic speech with the synthesized F_0 contours verified the validity of the developed scheme. For comparison, we also generated F_0 contours without decomposing them into tone and phrase components as most existing methods did. Although from the viewpoint of naturalness of synthetic speech, the result did not show clear advantage of the proposed method, from the

viewpoint of flexibility the advantage came clear: by manipulating phrase components in the proposed method, a better focus control was realized.

Pages: 154-159

Design of Tree-based Context Clustering for an HMM-based Thai Speech Synthesis System

Suphattharachai Chomphan, Takao Kobayashi

Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama, Japan

This paper proposes an approach to improving the correctness of tone of the synthesized speech which is generated by an HMM-based Thai speech synthesis system. In the tree-based context clustering process, tone groups and tone types are used to design four different structures of decision tree including a single binary tree structure, a simple tone-separated tree structure, a constancy-based-tone-separated tree structure, and a trend-based-tone-separated tree structure. A subjective evaluation of tone correctness is conducted by using tone perception of eight Thai listeners. The simple tone-separated tree structure gives the highest level of tone correctness, while the single binary tree structure gives the lowest level of tone correctness. Moreover, the additional contextual tone information which is applied to all structures of the decision tree achieves a significant improvement of tone correctness. Finally, the evaluation of syllable duration distortion among the four structures shows that the constancy-based-toneseparated and the trend-based-tone-separated tree structures can alleviate the distortions that appear when using the simple tone-separated tree structure.

Pages: 160-165

Development of a BOSS Unit Selection Module for Tone Languages

Arne Bachmann, Stefan Breuer

Institute of Communication Sciences (IfK), University of Bonn, Germany

The Bonn Open Synthesis System (BOSS) is a toolkit for the efficient development of speech synthesis applications. To facilitate adaptation to tone languages, we added support for tone contour quantization and prediction. Now it is possible to integrate syllable and word tone templates into the system and predict as well as select them efficiently. The simple model presented here is trained automatically and works independently of the morphophonemic rules specific to a certain tone language. Its feasibility is exemplified for the African language *Ibibio*.

Pages: 166-171

Poster Session 2

Thursday, August 23, 15:40-16:45

Unit-Selection Text-to-Speech Synthesis using an Asynchronous Interpolation Model

Alexander B. Kain, Jan P. H. van Santen

Center for Spoken Language Understanding (CSLU), OGI School of Science & Engineering at OHSU, Beaverton, OR, USA; and BioSpeech, Inc., Lake Oswego, OR, USA

We describe the Asynchronous Interpolation Model, which represents speech as a composition of several different types of feature streams that are computed using asynchronous interpolation of neighboring basis vectors, according to transition weights. When applied to the acoustic inventory of a concatenative Text-to-Speech synthesizer, the model eliminates concatenation errors and affords opportunities for high rates of compression and voice transformation. We propose a particular instance of the model that uses formant frequency values and formant-normalized complex spectra as two types of streams, in conjunction with a unit-selection synthesizer. During analysis, basis vectors and transition weights were estimated automatically, using three different labeling schemes and dynamic programming methods. An evaluation of the intelligibility and quality of the synthesized speech showed significant improvements over a standard, size-matched compression scheme. The proposed method was also able to convincingly transform speaker characteristics through replacement of basis vectors.

Pages: 172-177

Modelling Voiceless Speech Segments by means of an Additive Procedure based on the Computation of Formant Sinusoids

Ingo Hertrich, Hermann Ackermann

Department of General Neurology, University of Tübingen, Germany

A previously developed vowel synthesis algorithm implements formants as sinusoids, amplitude- and phase-modulated by the fundamental frequency (Hertrich and Ackermann, 1999, Journal of the Acoustical Society of America, 106, 2988- 2990). The present study extends this approach to the modelling of the acoustic characteristics of aperiodic speech segments. To these ends, a voiceless signal component is generated by adding at each sample point a random parameter onto the formants' phase progression. Voiceless stop consonants then can be modelled, e.g., by combining a release burst, i.e., an interval in which the formant sinusoids abruptly increase and gradually decrease in amplitude, with formantshaped noise components, representing inter-articulator frication, aspiration, and breathy vowel onset.

Pages: 178-181

Using Articulatory Position Data in Voice Transformation

Arthur R. Toth, Alan W. Black

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

Articulatory position data is information about the location of various articulators in the vocal tract. One form of it has been made freely available in the MOCHA database [1]. This data is interesting in that it provides direct information on the production of speech, but there is the question of whether it actually provides information beyond what can be derived from the audio signal, which is much easier to collect. Although there has been some success in improving small-scale speech recognition and in demonstrating mappings between articulatory positions and spectral features of the audio signal, there are many problems to which this data has not been applied. This work investigates the possibility of using articulatory position data to improve voice transformation, which is the process of making speech from one person sound as if it had been spoken by another. After further investigation, it appears to be difficult to use articulatory position data to improve voice transformation using state-of-the-art voice transformation techniques as we only had a few positive results across a range of experiments. To achieve these results, it was necessary to modify our baseline voice transformation approach and/or consider features derived from the articulatory positions.

Reference. [1] Wrench, A. (1999), "The MOCHA-TIMIT articulatory database," Queen Margaret University College,

Edinburgh,
http://www.cstr.ed.ac.uk/artic/mocha.html

Pages: 182-187

Text Processing for Text-to-Speech Systems in Indian Languages

Anand Arokia Raj (1), Tanuja Sarkar (1), Satish Chandra Pammi (1), Santhosh Yuvaraj (1), Mohit Bansal (2), Kishore Prahallad (1,3), Alan W. Black (3)

(1) International Institute of Information Technology, Hyderabad, India
(2) Indian Institute of Technology, Kanpur, India
(3) Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA

To build a natural sounding speech synthesis system, it is essential that the text processing component produce an appropriate sequence of phonemic units corresponding to an arbitrary input text. In this paper we discuss our efforts in addressing the issues of Font-to-Akshara mapping, pronunciation rules for Aksharas, text normalization in the context of building text-to-speech systems in Indian languages.

Pages: 188-193

Flexible Harmonic/Stochastic Speech Synthesis

Daniel Erro, Asunción Moreno, Antonio Bonafonte

Department of Signal Theory and Communications, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

In this paper, our flexible harmonic/stochastic waveform generator for a speech synthesis system is presented. The speech is modeled as the superposition of two components: a harmonic component and a stochastic or aperiodic component. The purpose of this representation is to provide a framework with maximum flexibility for all kind of speech transformations. In contrast to other similar systems found in the literature, like HNM, our system can operate using constant frame rate instead of a pitch-synchronous scheme. Thus, the analysis process is simplified, while the phase coherence is guaranteed by the new prosodic modification and concatenation procedures that have been designed for this scheme. As the system was created for voice conversion applications, in this work, as a previous step, we validate its performance in a speech synthesis context by comparing it to the well-known TD-PSOLA technique, using four different voices and different synthesis database sizes. The opinions of the listeners indicate that the methods and algorithms described are preferred rather than PSOLA, and thus are suitable for high-quality speech synthesis and for further voice transformations.

Pages: 194-199

Prosody Modelling in Czech Text-to-Speech Synthesis

Jan Romportl, Jirí Kala

Department of Cybernetics, University of West Bohemia, Pilsen, Czech Republic

This paper describes data-driven modelling of all three basic prosodic features - fundamental frequency, intensity and segmental duration - in the Czech text-to-speech system ARTIC. The fundamental frequency is generated by a model based on concatenation of automatically acquired intonational patterns. Intensity of synthesised speech is modelled by experimentally created rules which are in conformity with phonetics studies. Phoneme duration modelling has not been previously solved in ARTIC and this paper presents the first solution to this problem using a CART-based approach.

Pages: 200-205

Measuring Attribute Dissimilarity with HMM KL-Divergence for Speech Synthesis

Yong Zhao (1), Chengsuo Zhang (2), Frank K. Soong (1), Min Chu (1), Xi Xiao (2)

(1) Speech Group, Microsoft Research Asia, China
(2) Department of Electronic Engineering, Tsinghua University, China

This paper proposes to use KLD between context-dependent HMMs as target cost in unit selection TTS systems. We train context-dependent HMMs to characterize the contextual attributes of units, and calculate Kullback-Leibler Divergence (KLD) between the corresponding models. We demonstrate that the KLD measure provides a statistically meaningful way to analyze the underlining relations among elements of attributes. With the aid of multidimensional scaling, a set of attributes, including phonetic, prosodic and numerical contexts, are examined by graphically representing elements of the attribute as points on a low-dimensional space, where the distances among points agree with the KLDs among the elements. The KLD between multi-space probability distribution HMMs is derived. A perceptual experiment shows that the TTT system defined with the KLD-based target cost sounds slightly better than one with the manually-tuned.

Pages: 206-210

Lagrangian Relaxation for Optimal Corpus Design

Jonathan Chevelu, Nelly Barbot, Olivier Boeffard, Arnaud Delhay

IRISA / University of Rennes 1 - ENSSAT, Lannion, France

This article is interested in the problem of the linguistic content of a speech corpus. Depending on the target task (speech recognition, speech synthesis, etc) we try to control the phonological and linguistic content of the corpus by collecting an optimal set of sentences which make it possible to cover a preset description of phonological attributes (prosodic tags, allophones, syllables, etc) under the constraint of a minimal overall duration. This goal is classically achieved by greedy algorithms which however do not guarantee the optimality of the desired cover. We propose to call upon the principle of lagrangian relaxation where a set covering problem is solved by iterating between a primal and a dual spaces. We propose to evaluate our proposed methodology against a standard greedy algorithm in order to estimate an optimal phone and diphone covering in French. Our results show that our algorithm based on a lagrangian relaxation principle gives a 10% better solution than a standard greedy algorithm and especially enables to locate the absolute quality of the proposed solution by giving a lower bound to the set covering problem. According to our experiments, our best solution is only 0.8% far from the lower bound of the phone and diphone covering problem.

Pages: 211-216

Approaches for Adaptive Database Reduction for Text-to-Speech Synthesis

Aleksandra Krul (1), Géraldine Damnati (1), François Yvon (2), Cédric Boidin (1), Thierry Moudenc (1)

(1) France Télécom R&D Division, TECH/SSTP, Lannion, France
(2) GET/ENST and CNRS/LTCI, Paris, France

This paper raises the issue of speech database reduction adapted to a specific domain for Text-To-Speech (TTS) synthesis application. We evaluate several methods: a database pruning technique based on the statistical behaviour of the unit selection algorithm and a novel method based on the Kullback- Leibler divergence. The aim of the former method is to eliminate the least selected units during the synthesis of a domain specific training corpus. The aim of the latter approach is to build a reduced database whose unit distribution approximates a given target distribution. We compare the reduced databases. Finally we evaluate these methods on several objective measures given by the unit selection algorithm.

Pages: 217-222

Statistical Analysis of Filled Pauses' Rhythm for Disfluent Speech Synthesis

Jordi Adell (1), Antonio Bonafonte (1), David Escudero (2)

(1) Dpt. of Signal Theory and Comunications, Universitat Politècnica de Catalunya, Spain
(2) Dpt. Computer Science, Universidad de Valladolid, Spain

Given that state of the art speech synthesis systems have already reached a high naturalness level, it is time to move to talking speech from the actual read speech framework. For this purpose it is thus necessary to investigate how disfluencies can be included in speech synthesis and even increase its naturalness. This paper builds on a previously presented work and focuses on finding a local model of filled pauses rhythm. A statistical study of rhythm effects around filled pauses is presented and based on the correlation between rhythm variables, a regression model is proposed to predict filled pauses duration and prepausal lengthening.

Pages: 223-227

Quantitative Analysis of F_0 Contours of Emotional Speech of Mandarin

Wentao Gu, Tan Lee

Department of Electronic Engineering, the Chinese University of Hong Kong, China

The F_0 characteristics of Mandarin speech in four basic emotions (anger, fear, joy, and sadness) as well as in neutral reading are compared quantitatively. Two approaches are employed: analysis of surface features from time-normalized F_0 contours, and analysis-by-synthesis of time-intact F_0 contours based on the command-response model, which turns out to be also applicable to emotional speech. For surface F_0 features, the height and range of F_0, the local tonal variation, and the sentential F_0 declination are all investigated. In model-based analysis, the parameters for both phrase and tone commands are compared systematically. The study shows that those surface F_0 phenomena can be explained better by the model-based approach, which can later be used in F_0 generation for emotional speech synthesis.

Pages: 228-233

Session: Prosody Modelling

Thursday, August 23, 16:50-18:30

Maximum-Likelihood Dynamic Intonation Model for Concatenative Text-to-Speech System

Slava Shechtman

IBM Research Laboratory, Haifa, Israel

In this work we present a Maximum Likelihood (ML) joint pitch curve modeling, inspired by HMM TTS synthesis concept. This model provides an optimal solution for the coarse target intonation curve (3 points per syllable) and incorporates both static and dynamic pitch values for better utterance intonation modeling. The coarse intonation curve may be optionally combined with the original pitch extracted from the concatenated units, by a technique named *microprosody preservation*, which is also described. The latter is intended for reducing pitch modification ratio and improving sound naturalness for large-scale concatenative TTS systems. The proposed model was successfully applied on IBM's trainable concatenative TTS system improving the subjective intonation quality.

Pages: 234-239

Data-Driven Extraction of Intonation Contour Classes

Uwe D. Reichel

Institute of Phonetics and Speech Processing, University of Munich, Germany

In this paper we introduce the first steps towards a new data-driven method for extraction of intonation events that does not require any prerequisite prosodic labelling. Provided with data segmented on the syllable constituent level it derives local and global contour classes by stylisation and subsequent clustering of the stylisation parameter vectors. Local contour classes correspond to pitch movements connected to one or several syllables and determine the local F_0 shape. Global classes are connected to intonation phrases and determine the F_0 register. Local classes initially are derived for syllabic segments, which are then concatenated incrementally by means of statistical language modelling of co-occurrence patterns.

Due to its generality the method is in principle language independent and potentially capable to deal also with other aspects of prosody than intonation.

Pages: 240-245

Word Accentuation Prediction using a Neural Net Classifier

Taniya Mishra, Emily Tucker Prud'hommeaux, Jan P. H. van Santen

Center for Spoken Language Understanding, OGI School of Science & Engineering at OHSU, Beaverton, OR, USA

Automatic prediction of pitch accent assignment is an important but challenging task in text-to-speech synthesis (TTS). Early work in accent prediction relied on simple word-class distinctions, but recently more sophisticated inductive learning models using multiple features have been applied to the problem. For our neural network accent classifier, we developed a corpus that was labeled according to judgments of accent assignment appropriateness in synthesized speech rather than the usual ToBI annotation guidelines. Because the resulting training set was imbalanced, the baseline neural network we developed for this task had a very high accuracy rate (84%) but performed only slightly better than chance according to our ROC analysis. Balancing our training data using downsizing, oversampling, and cost-based post-processing yielded significant improvement in this informative measure. We anticipate that balance adjustments and the inclusion of more complex features will lead to further improvement.

Pages: 246-251

Issues of Optionality in Pitch Accent Placement

Leonardo Badino, Robert A. J. Clark

Centre for Speech Technology Research, University of Edinburgh, Scotland, UK

When comparing the prosodic realization of different English speakers reading the same text, a significant disagreement is usually found amongst the pitch accent patterns of the speakers. Assuming that such disagreement is due to a partial optionality of pitch accent placement, it has been recently proposed to evaluate pitch accent predictors by comparing them with multispeaker reference data. In this paper we face the issue of pitch accent optionality at different levels. At first we propose a simple mathematical definition of intra-speaker optionality which allows us to introduce a function for evaluating pitch accent predictors which we show being more accurate and robust than those used in previous works. Subsequently we compare a pitch accent predictor trained on single speaker data with a predictor trained on multi-speaker data in order to point out the large overlapping between intra-speaker and inter-speaker optionality. Finally, we show our successful results in predicting intra-speaker optionality and we suggest how this achievement could be exploited to improve the performances of a unit selection text-to speech synthesis (TTS) system.

Pages: 252-257

Session: Inventory Construction

Friday, August 24, 9:00-10:40

Single Speaker Segmentation and Inventory Selection Using Dynamic Time Warping Self Organization and Joint Multigram Mapping

Matthew P. Aylett, Simon King

Centre of Speech Technology Research, University of Edinburgh, Edinburgh, UK

In speech synthesis the inventory of units is decided by inspection and on the basis of phonological and phonetic expertise. The ephone (or emergent phone) project at CSTR is investigating how self organisation techniques can be applied to build an inventory based on collected acoustic data together with the constraints of a synthesis lexicon. In this paper we will describe a prototype inventory creation method using dynamic time warping (DTW) for acoustic clustering and a joint multigram approach for relating a series of symbols that represent the speech to these emerged units. We initially examined two symbol sets: 1) A baseline of standard phones 2) Orthographic symbols. The success of the approach is evaluated by comparing word boundaries generated by the emergent phones against those created using state-of-the-art HMM segmentation. Initial results suggest the DTW segmentation can match word boundaries with a root mean square error (RMSE) of 35ms. Results from mapping units onto phones resulted in a higher RMSE of 103ms. This error was increased when multiple multigram types were added and when the default unit clustering was altered from 40 (our baseline) to 10. Results for orthographic matching had a higher RMSE of 125ms. To conclude we discuss future work that we believe can reduce this error rate to a level sufficient for the techniques to be applied to a unit selection synthesis system.

Pages: 258-263

How (Not) to Select Your Voice Corpus: Random Selection vs. Phonologically Balanced

Tanya Lambert, Norbert Braunschweiler, Sabine Buchholz

Speech Technology Group, Cambridge Research Laboratory; Toshiba Research Europe Ltd., Cambridge, UK

This paper compares the effect of two different voice corpus selection methods on the overall quality of unit selection-based text-to-speech (TTS) voices resulting from training on these corpora. The first selection method aims to maximize the coverage of stressed as well as unstressed diphones (phonologically balanced: *Phonbal*) while the second method simply selects sentences at random (*Random*). We show that, as expected, the *Phonbal* method results in better phonetic and phonological coverage for the training as well as unseen test sentences. However, we also provide evidence from an objective evaluation and a subjective listening test that the *Random* method results in an overall better voice quality when only automatic corpus annotation tools (such as forced alignment) are used, and potentially even with manual annotation. This result has general implications for the fast creation of TTS voices.

Pages: 264-269

Unit Selection Synthesis Using Long Non-Uniform Units and Phonemic Identity Matching

Lukas Latacz, Yuk On Kong, Werner Verhelst

Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, Brussels, Belgium

This paper investigates two ways of improving synthesis quality: to maximise the length of selected units or to capitalise on phonemic context. For the former, it compares a synthesiser using a novel way of target specification and unit search with a standard unit selection synthesiser. For the latter, weights for phonemic context are set differently according to the distance of the phoneme concerned from the target diphone, and according to the class (consonant/vowel) to which the phoneme in question belongs. Both ways lead to improvements, at least when the speech database is small in size.

Pages: 270-275

Evaluation of Various Unit Types in the Unit Selection Approach for the Czech Language using the Festival System

Martin Gruber, Daniel Tihelka, Jindrich Matousek

Department of Cybernetics, University of West Bohemia, Pilsen, Czech Republic

The present paper focuses on the utilization of concatenative speech synthesis, aiming to determine and compare the influence on the synthesized speech quality when various unit types are used in the unit selection approach. There are several unit types which can be used for this purpose. This work deals with those most widely used, i.e. halfphones, diphones, phones, triphones and syllables. Speech was synthesized using these unit types and the outcome was listened to a by number of listeners, whose task was to evaluate the quality of synthetic speech. The result of the listening test performed for the Czech language is presented. However, it can be assumed that the results would be probably equal for other languages with similar structure, as we made no language-dependent modification in the Festival system. No research of a similar character has been conducted yet, so this unique evaluation should suggest what unit types are appropriate for general TTS systems.

Pages: 276-281

Keynote Session 2

Friday, August 24, 11:00-11:50

The Blizzard Challenge: Evaluating Corpus-based Speech Synthesis Techniques

Alan W. Black

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

The Blizzard Challenge was started in 2005 as a way to evaluate different corpus speech synthesis techniques on a common data set. It has been noted that it is very hard to evaluate different speech synthesis techniques when different size and quality databases are used to build a voice. To remove the variable of database size and speaker quality, we proposed a common database that all participants would use. The Challenge itself is for participants to take the given database (or databases) and build a voice using their voice building software. After a short time, a set of test sentences are released that are to be synthesized by each participants' system. The synthesized utterances are collected together and a web-based listening test is set up. Two types of listening tests are carried out, a simple MOS based test, and a set of understandability tests where the listener is asked to type in what they hear.

Three sets of listeners are used: speech experts (provided from the participants' groups), volunteers (collect by web advertising), and paid undergraduate native speakers. Each year the results have been presented at a workshop where participants present descriptions of their systems, and final results are given.

The challenge has brought together groups from academia and industry from around the world. Both established groups, and new groups have been represented. The results have been both interesting and unexpected.

But we see the Challenge as a long term evolving event. Modifications in the basic structure are being considered each year. For example: how to test if speaker identity is preserved in voice conversion based systems; how can we test multi-sentence synthesis; what about multi-lingual databases; and who is going to run it.

No individual results will be presented in this talk, but overall trends will be given as well as discussion of future directions for Blizzard.

A more detailed description of the motivation and details of the challenge is described in [1]. All the presentations including anonymized results are also available on line at http://festvox.org/blizzard/ .

Reference. [1] Black, A., and Tokuda, K., (2005) Blizzard Challenge 2005: Evaluating corpus-based speech synthesis on common datasets
Interspeech 2005, Lisbon, Portugal.

Pages: 392-392

Session: Applications

Friday, August 24, 11:50-12:40

Assessing the Adequate Treatment of Fast Speech in Unit Selection Speech Synthesis Systems for the Visually Impaired

Donata Moers, Petra Wagner, Stefan Breuer

Institut für Kommunikationswissenschaften, Abteilung Sprachliche Kommunikation, Rheinische Friedrich-Wilhelms-Universität
Bonn, Germany

This paper describes work in progress concerning the adequate modeling of fast speech in unit selection speech synthesis systems mostly having in mind blind and visually impaired users. Initially, a survey of the main phonetic characteristics of fast speech will be given. From this, certain conclusions concerning an adequate modeling of fast speech in unit selection synthesis will be drawn. Subsequently, a questionnaire assessing synthetic speech related preferences of visually impaired users will be presented. The last section deals with future experiments aiming at a definition of criteria for the development of synthesis corpora modeling fast speech within the unit selection paradigm.

Pages: 282-287

Making Speech Synthesis More Accessible to Older People

Maria Wolters (1), Pauline Campbell (2), Christine DePlacido (2) Amy Liddell (2), David Owens (2)

(1) Centre for Speech Technology Research, University of Edinburgh, Edinburgh, UK
(2) Audiology Division, Queen Margaret University, Edinburgh, UK

In this paper, we report on an experiment that tested users' ability to understand the content of spoken auditory reminders. Users heard meeting reminders and medication reminders spoken in both a natural and a synthetic voice. Our results show that older users can understand synthetic speech as well as younger users provided that the prompt texts are well-designed, using familiar words and contextual cues. As soon as unfamiliar and complex words are introduced, users' hearing affects how well they can understand the synthetic voice, even if their hearing would pass common screening tests for speech synthesis experiments. Although hearing thresholds correlate best with users' performance, central auditory processing may also influence performance, especially when complex errors are made.

Pages: 288-293

Session: Systems

Friday, August 24, 14:00-15:40

The HMM-based Speech Synthesis System (HTS) Version 2.0

Heiga Zen (1), Takashi Nose (2), Junichi Yamagishi (2,3), Shinji Sako (1,4) Takashi Masuko (2), Alan W. Black (5), Keiichi Tokuda (1)

(1) Nagoya Institute of Technology, Japan
(2) Tokyo Institute of Technology, Japan
(3) University of Edinburgh, UK
(4) Tokyo University, Japan
(5) Carnegie Mellon University, Pittsburgh, PA, USA

A statistical parametric speech synthesis system based on hidden Markov models (HMMs) has grown in popularity over the last few years. This system simultaneously models spectrum, excitation, and duration of speech using context-dependent HMMs and generates speech waveforms from the HMMs themselves. Since December 2002, we have publicly released an open-source software toolkit named HMM-based speech synthesis system (HTS) to provide a research and development platform for the speech synthesis community. In December 2006, HTS version 2.0 was released. This version includes a number of new features which are useful for both speech synthesis researchers and developers. This paper describes HTS version 2.0 in detail, as well as future release plans.

Pages: 294-299

eCIRCUS: Building Voices for Autonomous Speaking Agents

Christian Weiss (1), Luis C. Oliveira (1), Sergio Paulo (1), Carlos Mendes (1) Luis Figueira (1) Marco Vala (2), Pedro Sequeira (2), Ana Paiva (2), Thurid Vogt (3), Elisabeth Andre (3)

(1) INESC-ID/IST, Spoken Language Systems Laboratory, Lisbon, Portugal
(2) INESC-ID/IST, GAIPS, Lisbon, Portugal
(3) Institute of Computer Science, University of Augsburg, Germany

This paper describes our work integrating automatic speech generation into a virtual environment where autonomous agents are enabled to interact by natural spoken language. The application intents to address bullying problems for children aged 9-12 in the UK and Germany by presenting improvised dramas and by asking the user to act as an ”invisible friend” of the victimised character. As we are addressing an elementary school environment one specification of the resulting voice was building agecorresponding young school kids voices. The second specification addresses building a low-resource speech generation system which is capable to run on older school computers but is still fast enough in response time to guaranty a fluent conversation between the agents. Third requirement was integrating the speech-module with the agents. We focus on the speech generation system itself, pointing out possible implementation issues in building non-controlled speech interaction in virtual environments Furthermore we describe the problems arising in building unit-selection based child’s’ voice TTS and shows alternative methods to child’s voice recording by deploying voice transformation methods.

Pages: 300-303

Unit Selection Synthesis in the SmartWeb Project

Martin Barbisch, Grzegorz Dogil, Bernd Möbius, Bettina Säuberlich, Antje Schweitzer

Institute for Natural Language Processing, University of Stuttgart, Germany

This paper describes three aspects of the unit selection synthesis used in the SmartWeb dialog system. The synthesis module has been implemented in the IMS German Festival speech synthesis system. First, we compare a unit selection strategy developed in the course of the project to a strategy developed earlier. Second, we discuss our experiences with F_0 smoothing and amplitude modeling, which were both devised to reduce audible discontinuities. However, the results are inconclusive so far. Finally, we sketch a simple mechanism that addresses the problem of language disambiguation for proper names.

Pages: 304-309

Building a Finnish Unit Selection TTS system

Hanna Silen, Elina Helander Konsta Koppinen, Moncef Gabbouj

Institute of Signal Processing, Tampere University of Technology, Finland

Speech synthesis based on unit selection can produce far more natural speech than conventional diphone-based methods. Unit selection based text-to-speech synthesizers have been built for many different languages. In this paper, we describe the development of TUT VOICE, the first Finnish unit selection synthesis engine for academic research. The system includes database construction, synthesis engine implementation and optimization for Finnish.

Pages: 310-315

Poster Session 3

Friday, August 24, 15:40-16:45

Evaluating Automatic Syllabification Algorithms for English

Yannick Marchand (1,2), Connie R. Adsett (1,2), Robert I. Damper (1,3)

(1) Institute for Biodiagnostics (Atlantic), National Research Council Canada, Halifax, Nova Scotia, Canada
(2) Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada
(3) School of Electronics and Computer Science University of Southampton, UK

Automatic syllabification of words is challenging, not least because the syllable is difficult to define precisely. This task is important for word modelling in the composition process of concatenative synthesis as well as in automatic speech recognition. There are two broad approaches to perform automatic syllabification: rule-based and data-driven. The rule-based method effectively embodies some theoretical position regarding the syllable, whereas the data-driven paradigm infers new’ syllabifications from examples assumed to be correctly-syllabified already. This paper compares the performance of the two basic approaches. However, it is difficult to determine a correct syllabification in all cases and so to establish the quality of the gold standard’ corpus used either to quantitatively evaluate the output of an automatic algorithm or as the example-set on which data-driven methods crucially depend. Thus, three lexical databases of pre-syllabified words were used. Two of these lexicons hold the same 18,016 words with their corresponding syllabifications coming from independent sources, whereas the third corresponds to the 13,594 words that share the same syllabifications according to these two sources. As well as one rule-based approach (Fisher’s implementation of Kahn’s syllabification theory), three data-driven techniques are evaluated: a look-up procedure, an exemplar-based generalization technique, and syllabification by analogy (SbA). The results on the three databases show consistent and robust patterns: the datadriven techniques outperform the rule-based system in word and juncture accuracies by a very significant margin and best results are obtained with SbA.

Pages: 316-321

Voice Building from Insufficient Data - Classroom Experiences with Web-Based Language Development Tools

John Kominek, Tanja Schultz, Alan W. Black

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA

To make the goal of building voices in new languages easier and more accessible to non-experts, the combined tasks of phoneme set definition, text selection, prompt recording, lexicon building, and voice creation in Festival are now integrated behind a web-based development environment. This environment has been exercised in a semester-long laboratory course taught at Carnegie Mellon University. Here we report on the students’ efforts in building voices for the languages of Bulgarian, English, German, Hindi, Konkani, Mandarin, and Vietnamese. In some cases intelligible synthe-

sizers were built from as little as ten minutes of recorded speech.

Pages: 322-327

SVM Based Feature Extraction in Speech Synthesis

Peter Cahill, Jan Macek, Julie Carson-Berndsen

School of Computer Science and Informatics, University College Dublin, Ireland

Annotations of speech recordings are a fundamental part of any unit selection speech synthesiser. However, obtaining flawless annotations is an almost impossible task. Manual techniques can achieve themost accurate annotations, provided that enough time is available to analyse every phone individually. Automatic annotation techniques are a lot faster than manual, doing the task in a much more reasonable time frame, but such annotations contain a considerable amount of error. In this paper a technique is introduced that can quite accurately ensure a degree of articulatory-acoustic similarity between annotated units. The synthesiser will encourage the use of units that have been identified to have appropriate articulatory-acoustic parameters, but will not limit the domain of the speech database. This helps to identify where joins can be performed best and also identifies which annotations should be avoided at the phone level.

Pages: 328-332

Spectral Conversion based on Statistical Models Including Time-Sequence Matching

Yoshihiko Nankaku (1), Kenichi Nakamura (1), Tomoki Toda (2), Keiichi Tokuda (1)

(1) Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, Aichi, Japan;
(2) Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, Japan

This paper proposes a spectral conversion technique based on a new statistical model which includes time-sequence matching. In conventional GMM-based approaches, the Dynamic Programming (DP) matching between source and target feature sequences is performed prior to the training of GMMs. Although a similarity measure of two frames, e.g., the Euclid distance is typically adopted, this might be inappropriate for converting the spectral features. The likelihood function of the proposed model can directly deal with two different length sequences, in which a frame alignment of source and target feature sequences is represented by discrete hidden variables. In the proposed algorithm, the maximum likelihood criterion is consistently applied to the training of model parameters, sequence matching and spectral conversion. In the subjective preference test, the proposed method is superior than the conventional GMM-based method.

Pages: 333-338

Analysis of Affective Speech Recordings using the Superpositional Intonation Model

Esther Klabbers, Taniya Mishra, Jan P. H. van Santen

Center for Spoken Language Understanding, OGI School of Science & Engineering at OHSU, Beaverton, OR, USA

This paper presents an analysis of affective sentences spoken by a single speaker. The corpus was analyzed in terms of different acoustic and prosodic features, including features derived from the decomposition of pitch contours into phrase and accent curves. It was found that sentences spoken with a sad affect were most easily distinguishable from other affects as they were characterized by a lower F_0, lower phrase and accent curves, lower overall energy and a higher spectral tilt. Fearful was also relatively easy to distinguish from angry and happy as it exhibited flatter phrase curves and lower accent curves. Angry and happy were more difficult to distinguish from each other, but angry was shown to exhibit a higher spectral tilt and a lower speaking rate. The analysis results provide informative clues for synthesizing affective speech using our proposed recombinant synthesis method.

Pages: 339-344

Calliphony: A Real-Time Intonation Controller for Expressive Speech Synthesis

Sylvain Le Beux, Albert Rilliard, Christophe d'Alessandro

LIMSI-CNRS, Orsay, France

Intonation synthesis using a hand-controlled interface is a new approach for effective synthesis of expressive prosody. A system for prosodic real time modification is described. The user is controlling prosody in real time by drawing contours on a graphic tablet while listening to the modified speech. This system, a pen controlled speech instrument, can be applied to text to speech synthesis along two lines. A first application is synthetic speech post-processing. The synthetic speech produced by a TTS system can be very effectively tuned by hands for expressive synthesis. A second application is database enrichment. Several prosodic styles can be applied to the sentences in the database without the need of recording new sentences. These two applications are sketched in the paper.

Pages: 345-350

Epoch Synchronous Non-Overlap-Add (ESNOLA) Method-Based Concatenative Speech Synthesis System for Bangla

Shyamal Kumar Das Mandal, Asoke Kumar Datta

Centre for Development of Advanced Computing (C-DAC), Kolkata, India

In the last decade there has been a shift towards development of speech synthesizer using concatenative synthesis technique instead of parametric synthesis. There are a number of different methodologies for concatenative synthesis like TDPSOLA, PSOLA, and MBROLA. This paper, describes a concatenative speech synthesis system based on Epoch Synchronous Non Over Lapp Add (ESNOLA) technique, for standard colloquial Bengali, which uses the partnemes as the smallest signal units for concatenation. The system provided full control for prosody and intonation.

Pages: 351-355

Syllable-Based Thai Duration Model using Multi-Level Linear Regression and Syllable Accommodation

Chatchawarn Hansakunbuntheung (1), Hiroaki Kato (2), Yoshinori Sagisaka (1)

(1) GITI/Language and Speech Science Research Laboratory, Waseda University, Tokyo, Japan
(2) NICT/ATR Cognitive Information Science Labs, Kyoto, Japan

This paper proposes a syllable-based Thai duration model using multi-level linear regression and syllable accommodation. To build a timing model reflecting control characteristics directly, we introduce two analysis results on hierarchical control characteristics. First analysis result showed that syllable is highly correlated to higher-phone-level timing controls, while phone differences by themselves do not affect higher control and contribute to local timing control only.

Second one on the syllable accomodation showed that phone duration highly depends on local phone factors. These analysis results support a syllable-based hierarchical model proposed in this paper. Duration prediction experiments of 5-fold cross validation showed 46.73 and 32.37 ms in RMS error, and, 0.905 and 0.811 in correlation between measured and predicted duration at syllable and phone levels, respectively. The comparison of predicted precision showed that the proposed syllable-based multi-level duration model better performed than a conventional single-level phone duration model.

Pages: 356-361

Linguistic and Mixed Excitation Improvements on a HMM-based speech synthesis for Castilian Spanish

Xavier Gonzalvo, Joan Claudi Socoró, Ignasi Iriondo, Carlos Monzo, Elisa Martínez

GPMM - Grup de Recerca en Processament Multimodal, Enginyeria i Arquitectura La Salle. Universitat Ramon Llull, Barcelona, Spain

Hidden Markov Models based text-to-speech (HMM-TTS) synthesis is one of the techniques for generating speech from trained statistical models where spectrum and prosody of basic speech units are modelled altogether. This paper presents the advances in our Spanish HMM-TTS and a perceptual test is conducted to compare it with an extended PSOLA-based concatenative (E-PSOLA) system. The improvements have been performed on phonetic information and contextual factors according to the Castilian Spanish language and speech generation using a mixed excitation (ME) technique. The results show the preference of the new HMM-TTS system in front of the previous system and a better MOS in comparison with a real E-PSOLA in terms of acceptability, intelligibility and stability.

Pages: 362-367

Inventory of Intonation Contours for Text-to-Speech Synthesis

Tetyana Lyudovyk, Valentyna Robeiko

International Research/Training Center for Information Technologies and Systems, Kyiv, Ukraine

This paper presents an intonation model which determines intonation contours over intonation phrases. The model is described by four elements: communicative type of an intonation phrase; number of accent groups in it; position of the nuclear accent group in it; and set of target intonation points. Individualization of the model is based on semiautomatic analysis of speaker database. The model was implemented in unit selection TTS system for Ukrainian.

Pages: 368-373

Session: Evaluation

Friday, August 24, 16:50-18:10

Analysis Methods for Assessing TTS Intelligibility

H. Timothy Bunnell, Jason Lilley

Center for Pediatric Auditory and Speech Sciences, Nemours Biomedical Research, USA & Department of Linguistics & Cognitive Sciences, University of Delaware, USA

Semantically unpredictable (SU) sentences are often used to assess intelligibility of TTS systems, but analyses of listener responses to SU sentences can be a labor-intensive process. In this paper we compare several approaches to the analysis of data from an SUS task. Data from a study comparing five TTS systems were analyzed in a variety of ways ranging from string edit measures based on carefully hand-corrected phonetically transcribed responses to largely uncorrected words- or sentences-correct measures. Results suggest that a simple sentences-correct measure is adequate when only rank order information is of interest. However, the sentences-correct measure masks the magnitude of differences between systems and should be avoided when it is important to gage how large the difference in intelligibility is between systems. In preparing response data for analysis, careful human interpretation of listener response data can lead to higher intelligibility measures overall, but does not interact with TTS system or other factors and consequently does not lead to different conclusions when comparing multiple TTS systems. This suggests that largely automated scoring procedures are feasible.

Pages: 374-379

Understandable Production of Massive Synthesis

Brian Langner, Alan W. Black

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

This paper explores *massive synthesis*, or synthesis of sufficiently large amounts of content such that its evaluation is challenging. We discuss various applications where massive synthesis may apply, and their related issues. We also outline factors related to those applications that affect the perceived quality and intelligibility of the speech output, and discuss modifications of those factors that can improve the understandability of the resulting synthetic speech. There is a discussion of the challenges of evaluating this work, and of the different possible metrics that may be appropriate. Finally, we show in a simple evaluation that our modifications improve the perceived quality of the synthesis.

Pages: 380-384

The Online Evaluation of Speech Synthesis using Eye Movements

Charlotte van Hooijdonk, Edwin Commandeur, Reinier Cozijn, Emiel Krahmer, Erwin Marsi

Department of Communication & Information Sciences, Tilburg University, The Netherlands

This paper describes an eye tracking experiment to study the processing of diphone synthesis, unit selection synthesis, and human speech taking segmental and suprasegmental speech quality into account. The results showed that both factors influenced the processing of human and synthetic speech, and confirmed that eye tracking is a promising albeit time consuming research method to evaluate synthetic speech.

Pages: 385-390

Sixth ISCA Workshop on Speech Synthesis
Bonn, Germany
August 22-24, 2007

Author Index

Learning Optimal Audiovisual Phasing for an HMM-based Control Model for Facial Animation

Oxana Govokhina[1,2], *Gérard Bailly*[1] *and Gaspard Breton*[2]

[1] GIPSA-Lab Dpt. Speech & Cognition CNRS/INPG/UJF/Stendhal 38041 Grenoble - France
{Oxana.Govokhina,Gerard.Bailly}@gipsa-lab.inpg.fr
[2] France TelecomR&D, 4 rue du Clos Courtel, BP 59 35512 Cesson-Sévigné - France
Gaspard.Breton@orange-ftgroup.com

Abstract

We propose here an HMM-based trajectory formation system that predicts articulatory trajectories of a talking face from phonetic input. In order to add flexibility to the acoustic/gestural alignment and take into account anticipatory gestures, a phasing model has been developed that predicts the delays between the acoustic boundaries of allophones to be synthesized and the gestural boundaries of HMM triphones. The HMM triphones and the phasing model are trained simultaneously using an iterative analysis-synthesis loop. Convergence is obtained within a few iterations. We demonstrate here that the phasing model improves significantly the prediction error and captures subtle context-dependent anticipatory phenomena.

1. Introduction

Embodied conversational agents – virtual characters as well as anthropoid robots – should be able to compute facial movements from symbolic input in order to engage in conversation with human partners. This symbolic input minimally consists in the phonetic string with phoneme durations. It can be enriched with more phonological information, facial expressions, or paralinguistic information that has an impact on speech articulation (mental or emotional state). A trajectory formation model has thus to be built that computes articulatory parameters from such a symbolic specification of the speech task. These articulatory parameters will then drive the plant (the shape and appearance models of a talking face or the control model of the robot).

Human interlocutors are sensitive to discrepancies between the visible and audible consequences of articulation [1, 2] and have strong expectations on articulatory variability [3] resulting from the under-specification of articulatory targets and planning. The effective modeling of coarticulation in speech is therefore a challenging issue for trajectory formation systems.

Audiovisual speech synthesizers should therefore cope not only with the modeling of adequate inter-articulatory coordination but also with the correct synchronization of audible and visible articulation [4]. Central to all speech synthesizers using rules, stored segments or trajectory formation models to generate speech from phonological input is the choice of speech landmarks. In most systems acoustic boundaries between phones are used as such landmarks for prosody characterization or generation. We question here the relevance of these landmarks for the generation of gestural scores.

2. State-of the art

Several strategies can be proposed to build audiovisual text-to-speech synthesis [5]. The most straightforward solution simply consists in driving a trajectory formation model from the phoneme string and phoneme durations computed by an existing text-to-speech system. The trajectory formation model then uses acoustic phoneme boundaries to anchor the gestural score and the coarticulation model if necessary. Coarticulation is usually predicted using rules [6] or by exploiting an explicit coarticulation model [7, 8] that anchor the positions and spans of the phoneme-specific gestural targets. Interestingly, Kaburagi and Honda [9] have proposed to add dynamic features in the specification of gestural targets in order to cope with inter-gestural phasing relations.

Data-driven trajectory formation systems have also been proposed to automatically capture regularities of the context-dependent gestural realization of phoneme-sized segments [10]. Concatenative audiovisual speech synthesis encapsulates coarticulation effects by storing multimodal segments. The problem of possible asynchronies is thus pushed in the segmentation and smoothing of boundaries and eventually in the compression/expansionof segments if required. Although HMMs are intrinsically generation engines that are tuned to emit a set of training observations, they have been used only recently for speech synthesis and particularly as trajectory formation systems [11, 12]. HMMs can in fact capture inter-gestural phasing relations thanks to the state-dependent static and dynamic probability density functions characterizing the sub-phonemic observations. Although HMM structures have been proposed [13] to take into account larger audiovisual asynchronies, the benefit for audiovisual recognition scores is highly discussed [14]. We should also mention a third possibility that consists in computing articulation directly from speech signals. Proposals range from frame-based linear [15] or nonlinear models to GMM-based or HMM-based mapping models that take as input a large speech window surrounding the current analysis frame [11]. The key problem is here to determine the span of coarticulation and hope that the mapping model will learn context-dependent phasing patterns from training data.

We study here an HMM-based trajectory formation system and claim that audiovisual asynchrony has an impact on its performance. A phasing model has thus been developed that predicts the delays between the acoustic boundaries of allophones to be synthesized and the gestural boundaries of HMM triphones that are proposed by unconstrained HMM alignment.

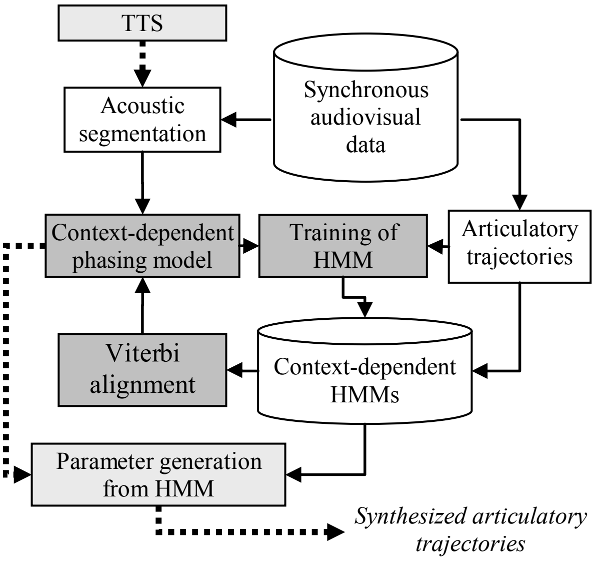

TTS
Acoustic segmentation
Synchronous audiovisual data
Context-dependent phasing model
Training of HMM
Articulatory trajectories
Viterbi alignment
Context-dependent HMMs
Parameter generation from HMM
Synthesized articulatory trajectories

Figure 1. Training consists in iteratively refining the context-dependent phasing model and HMMs (plain lines and dark blocks). The phasing model computes the average delay between acoustic boundaries and HMM boundaries obtained by aligning current context-dependent HMMs with training utterances. Synthesis simply consists in forced alignment of selected HMMs with boundaries predicted by the phasing model (dotted lines and light blocks).

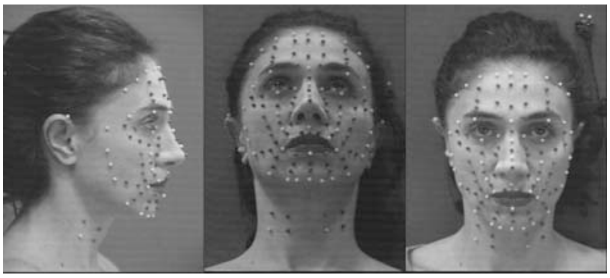


Figure 2. 125 colored beads have been glued on the subject's face along Langer's lines so that to cue geometric deformations caused by main articulatory movements when speaking.

3. Data and articulatory model

In order to be able to compare up-to-date data-driven methods for audiovisual synthesis, a main corpus of 697 sentences pronounced by a female speaker was recorded. Using a greedy algorithm, the phonetic content of these sentences was designed in order to maximize statistical coverage of triphones (differentiated also with respect to syllabic and word boundaries).

We used the motion capture technique developed at ICP [16, 17] that consists in collecting precise 3D data on selected visemes. 3D movements of facial fleshpoints (see Figure 2) are acquired using photogrammetry and hand-fitted generic models. Visemes are selected by an analysis-by-synthesis technique [18] that combines robust automatic tracking with semi-automatic correction.

Our shape models are built using a so-called guided Principal Component Analysis (PCA) where a priori knowledge is introduced during the linear decomposition. We in fact compute and iteratively subtract predictors using carefully chosen data subsets [19]. For speech movements, this methodology enables us to extract six components directly related to jaw, proper lip movements and clear movements of the throat linked with underlying movements of the larynx and hyoid bone. The resulting articulatory model also includes components for head movements and basic facial expressions but only components related to speech articulation are considered here.

We use here only the first 230 sentences for training and 10 sentences for testing. The average modeling error for training frames is less than half a millimeter for beads located on the lower face.

4. The trajectory formation system

The principle of speech synthesis by HMM was first introduced by Donovan for acoustic speech synthesis [20]. This was extended to audiovisual speech by the HTS working group [21]. The HMM-trajectory synthesis technique comprises training and synthesis parts.

4.1. Basic principles

An HMM and a duration model for each state are first learned for each segment of the training set. The input data for the HMM training is a set of observation vectors. The observation vectors consist of static and dynamic parameters, i.e. the values of articulatory parameters and their temporal derivatives. The HMM parameter estimation is based on ML (Maximum-Likelihood) criterion [22]. The ML estimation is achieved using a particular EM (Expectation Maximization) algorithm known as the Baum-Welch recursion algorithm. Usually, for each phoneme in context, a 3-state left-to-right model with single Gaussian diagonal output distributions. The state durations of each HMM are usually modeled as single Gaussian distributions. A second training step may also be added to factor out similar output distributions among the entire set of states (state tying).

The synthesis is performed as follows. The phonetic string to be synthesized is first chunked into segments and a sequence of HMM states is built by concatenating the corresponding segmental HMMs. State durations for the HMM sequence are determined so that the output probabilities of the state durations are maximized (thus usually by z-scoring) From the HMM sequence with the proper state durations assigned, a sequence of observation parameters is generated using a specific ML-based parameter generation algorithm [12].

4.2. Comments

This trajectory formation system exploits the dynamic parameters both in training and synthesis: the generated trajectory reflects both the means and covariances of the output distributions of a number of frames before and after each of the frames. By this way, this algorithm may incorporate implicitly part of short-term coarticulation patterns and inter-articulatory asynchrony. Larger coarticulation effects can also be captured since triphones intrinsically depend on adjacent phonetic context.

Note however that these coarticulation effects are anchored to acoustic boundaries that are imposed as synchronization events between the duration model and the HMM sequence. Intuitively we can suppose that context-dependent HMM can easily cope with this constraint. We show here that adding a context-dependent phasing model helps the trajectory formation system to better adjust to observed trajectories.

4.3. Adding and learning a phasing model

We propose to add a phasing model to the standard HMM-based trajectory formation system (see Figure 1) that consists in learning the time lag between acoustic and gestural units i.e. between acoustic boundaries delimiting allophones and gestural boundaries delimiting pieces of the articulatory score observed/generated by the context-dependent HMM sequence.

We test here a very simple phasing model: a unique time lag is associated with each context-dependent HMM. This lag is computed as the mean delay between acoustic boundaries and unconstrained alignment of triphones with articulatory trajectories of training utterances.

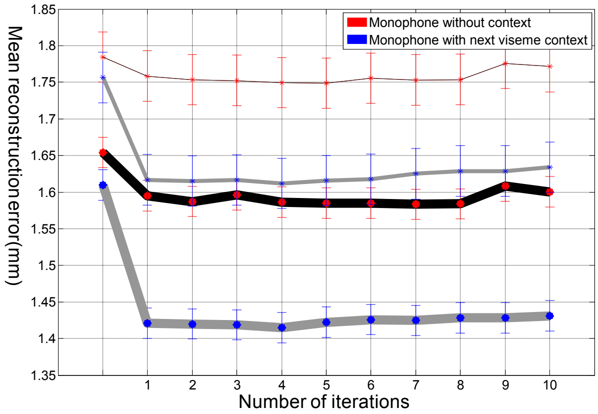


Figure 3: Mean reconstruction error as a function of number of iterations for context independent (black) and context-dependent phone HMMs (light gray). Results for training vs. test utterances are displayed respectively with thick vs. thin lines. Convergence is very fast and the phasing model benefits even more from contextual information.

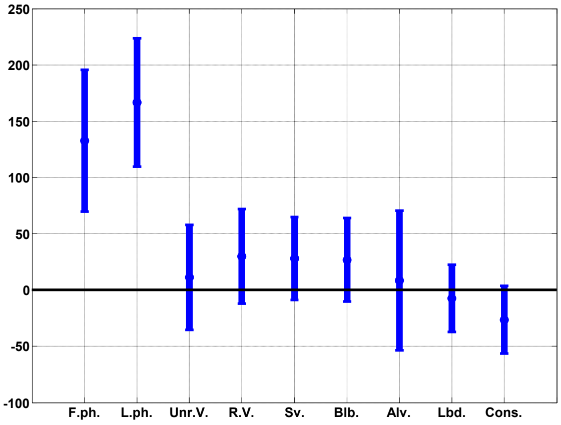


Figure 4: Average duration (ms) increase/decrease of the gestural segment with reference to its acoustic duration according to position and phoneme category. From left to right: first and final segment of the utterance, unrounded, rounded vowels, semivowels, bilabials, alveolars, labiodentals and remaining consonants.

5. Results

Figure 3 shows the significant decrease of prediction error when the phasing model is introduced in the HMM-based trajectory formation model. The convergence is obtained within 2 iterations: regularization constraints guarantying minimum durations of segments should be applied at least one time to avoid degeneration of the model.

Figure 4 shows that most gestural expansions occur at initial and final positions in the utterance (capturing prephonatory gestures and termination of phonation). Slow vocalic gestures generally expand whereas rapid consonantal gestures shrink: this is completely in accordance to the well-known numerical model of coarticulation proposed by Öhman [23] that superposes and blends vocalic and consonantal tongue gestures. The trajectory formation model places boundaries between segments so that dynamic information contained by observation probabilities of flanking HMM states best capture the variations of gestural speeds at the boundaries. Figure 5 gives an example of the necessary compromise between speech and duration: the large rounding gesture due to the semi-vowel [ɥ] is adequately predicted by the proposed system because the phasing model expands the duration of the gesture compared to the observed acoustic duration of the sound.

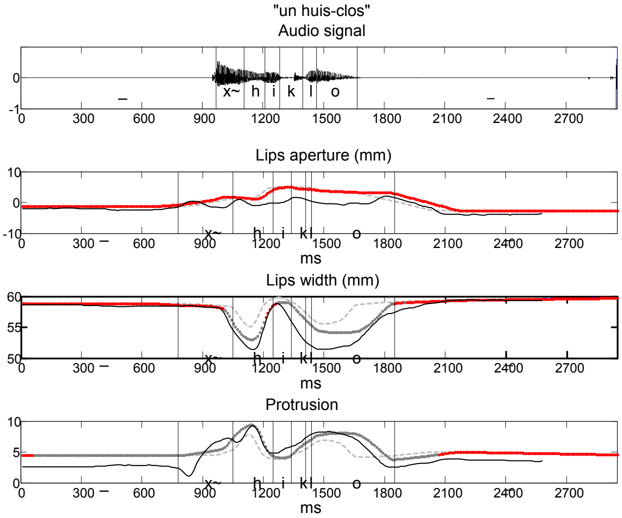


Figure 5. Comparing prediction of lip geometry by context-dependent HMMs trained either using acoustic (light gray) or gestural boundaries (dark gray) with original test data (black). The utterance is: ”un huis clos” [œ̃ɥiklo]. Note the expansion of initial and final movements (enabling the large final rounding movement) as well as the expansion of the semivowel [ɥ] with the following [i] shifted forward in time.

6. Conclusions

We have demonstrated here that the prediction accuracy of an HMM-based trajectory formation system can be greatly improved by modeling the phasing relations between acoustic and gestural boundaries. The phasing model is learned using an analysis-synthesis loop that uses constrained and unconstrained HMM alignments with the original data. We have shown that this scheme improves significantly the prediction error and captures subtle context-dependent anticipatory phenomena.

The interest of such an HMM-based trajectory formation system is double: (a) it provides accurate and smooth articulatory trajectories that can be used straightforwardly to control the articulation of a talking face or used as a skeleton to anchor multimodal concatenative synthesis [see notably the TDA proposal in 24]; (b) it also provides gestural segmentation as a by-product of the phasing model. These gestural boundaries can be used to segment original data for multimodal concatenative synthesis. This segmentation can also be used for asynchronous audiovisual speech recognition.

References

[1] N. F. Dixon and L. Spitz, "The detection of audiovisual desynchrony," *Perception*, vol. 9, pp. 719-721, 1980.

[2] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, pp. 746-748, 1976.

[3] D. H. Whalen, "Coarticulation is largely planned," *Journal of Phonetics*, vol. 18, pp. 3-35, 1990.

[4] K. W. Grant, V. van Wassenhove, and D. Poeppel, "Discrimination of auditory-visual synchrony," presented at Audio Visual Speech Processing, St Jorioz, France, 2003.

[5] G. Bailly, M. Bérar, F. Elisei, and M. Odisio, "Audiovisual speech synthesis," *International Journal of Speech Technology*, vol. 6, pp. 331-346, 2003.

[6] J. Beskow, "Rule-based Visual Speech Synthesis," presented at Eurospeech, Madrid, Spain, 1995.

[7] M. M. Cohen and D. W. Massaro, "Modeling coarticulation in synthetic visual speech," in *Models and Techniques in Computer Animation*, D. Thalmann and N. Magnenat-Thalmann, Eds. Tokyo: Springer-Verlag, 1993, pp. 141-155.

[8] G. Bailly, G. Gibert, and M. Odisio, "Evaluation of movement generation systems using the point-light technique," presented at IEEE Workshop on Speech Synthesis, Santa Monica, CA, 2002.

[9] T. Kaburagi and M. Honda, "A model of articulator trajectory formation based on the motor tasks of vocal-tract shapes," *Journal of the Acoustical Society of America*, vol. 99, pp. 3154-3170, 1996.

[10] C. Weiss, "Framework for data-driven video-realistic audio-visual speech synthesis," presented at Int. Conf. on Language Resources and Evaluation, Lisbon, 2004.

[11] M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "Visual speech synthesis based on parameter generation from HMM: speech-driven and text-and-speech-driven approaches," presented at Auditory-visual Speech Processing Workshop, Terrigal, Sydney, Australia, 1998.

[12] H. Zen, K. Tokuda, and T. Kitamura, "An introduction of trajectory model into HMM-based speech synthesis," presented at ISCA Speech Synthesis Workshop, Pittsburgh, PE, 2004.

[13] G. Gravier, G. Potamianos, and C. Neti, "Asynchrony modeling for audio-visual speech recognition," presented at Human Language Technology Conference, San Diego, CA, 2002.

[14] T. J. Hazen, "Visual model structures and synchrony constraints for audio-visual speech recognition," *IEEE Trans. on Speech and Audio Processing*, 2005.

[15] T. Kuratate, K. G. Munhall, P. E. Rubin, E. Vatikiotis-Bateson, and H. Yehia, "Audio-visual synthesis of talking faces from speech production correlates," presented at EuroSpeech, 1999.

[16] L. Revéret, G. Bailly, and P. Badin, "MOTHER: a new generation of talking heads providing a flexible articulatory control for video-realistic speech animation," presented at International Conference on Speech and Language Processing, Beijing, China, 2000.

[17] F. Elisei, M. Odisio, G. Bailly, and P. Badin, "Creating and controlling video-realistic talking heads," presented at Auditory-Visual Speech Processing Workshop, Scheelsminde, Denmark, 2001.

[18] G. Bailly, F. Elisei, P. Badin, and C. Savariaux, "Degrees of freedom of facial movements in face-to-face conversational speech," presented at International Workshop on Multimodal Corpora, Genoa - Italy, 2006.

[19] P. Badin, G. Bailly, L. Revéret, M. Baciu, C. Segebarth, and C. Savariaux, "Three-dimensional linear articulatory modeling of tongue, lips and face based on MRI and video images," *Journal of Phonetics*, vol. 30, pp. 533-553, 2002.

[20] R. Donovan, "Trainable speech synthesis," in *Univ. Eng. Dept.* Cambridge, UK: University of Cambridge, 1996, pp. 164.

[21] M. Tamura, S. Kondo, T. Masuko, and T. Kobayashi, "Text-to-audio-visual speech synthesis based on parameter generation from HMM," presented at EUROSPEECH, Budapest, Hungary, 1999.

[22] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," presented at IEEE International Conference on Acoustics, Speech, and Signal Processing, Istanbul, Turkey, 2000.

[23] S. E. G. Öhman, "Numerical model of coarticulation," *Journal of the Acoustical Society of America*, vol. 41, pp. 310-320, 1967.

[24] O. Govokhina, G. Bailly, G. Breton, and P. Bagshaw, "TDA: A new trainable trajectory formation system for facial animation," presented at InterSpeech, Pittsburgh, PE, 2006.

Control Concepts for Articulatory Speech Synthesis

Peter Birkholz[1], *Ingmar Steiner*[2], *Stefan Breuer*[3]

[1]Institute for Computer Science, University of Rostock, Germany
[2]Department of Computational Linguistics and Phonetics, Saarland University, Germany
[3]Institute of Communication Sciences (IfK), University of Bonn, Germany

piet@informatik.uni-rostock.de, steiner@coli.uni-saarland.de, breuer@ifk.uni-bonn.de

Abstract

We present two concepts for the generation of gestural scores to control an articulatory speech synthesizer. Gestural scores are the common input to the synthesizer and constitute an organized pattern of articulatory gestures. The first concept generates the gestures for an utterance using the phonetic transcriptions, phone durations, and intonation commands predicted by the Bonn Open Synthesis System (BOSS) from an arbitrary input text. This concept extends the synthesizer to a text-to-speech synthesis system. The idea of the second concept is to use timing information extracted from Electromagnetic Articulography signals to generate the articulatory gestures. Therefore, it is a concept for the re-synthesis of natural utterances. Finally, application prospects for the presented synthesizer are discussed.

1. Introduction

Articulatory speech synthesis is the most rigorous way of synthesizing speech, as it constitutes a simulation of the mechanisms underlying real speech production. Compared to other approaches in speech synthesis, it has the potential to synthesize speech with any voice and in any language with the most natural quality. Further advantages of articulatory speech synthesis are discussed by Shadle and Damper [17]. However, despite its potential, it is still a difficult task to actually achieve an average speech quality for one specific voice and language with an articulatory speech synthesizer. The problem are the high demands on the models for the various aspects of speech production. One of these aspects is the generation of speech movements, i.e., the control of the model articulators. In this paper, we present (i) a novel control model based on articulatory gestures and (ii) propose two concepts for the high-level prediction of the gestural parameters. The control model was implemented as part of an articulatory speech synthesizer based on a 3D model of the vocal tract and a comprehensive aeroacoustic simulation method [3, 4, 5]. The goal of the proposed high-level concepts is to specify the articulatory gestures in the form of a *gestural score* needed for the generation of the speech movements from different sources of input.

The idea of the first concept is to generate speech from text using the open source software platform BOSS (Bonn Open Synthesis System) [8]. BOSS was originally developed as a unit-selection speech synthesis system comprising modules for phonetic transcription, phone duration prediction, intonation generation, and the actual unit-selection step. In this study, we present a way to transform the output of the modules for phonetic transcription and phone duration prediction into the gestural score for the articulatory synthesizer.

The idea of the second concept is to use timing information extracted from Electromagnetic Articulography (EMA) signals to create the artificial gestural scores. Since EMA signals reflect the articulatory movements of real speakers, this is a concept for the *re*synthesis of speech. In other words, the second concept is an attempt to copy the speech of a speaker recorded by an EMA device, primarily with respect to gestural timing.

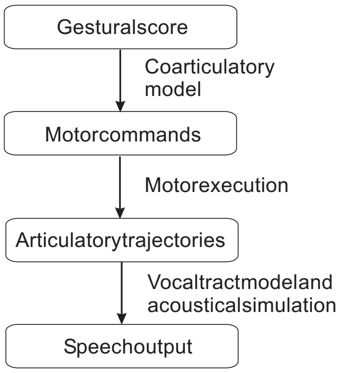


Figure 1: Flow diagram of the articulatory synthesizer.

The speech generation chain of the articulatory synthesizer is depicted in Figure 1. As mentioned above, the input to the synthesizer is a gestural score. It can be regarded as a representation of the intended utterance in terms of gestures for the glottal and the supraglottal articulators. As in the framework of articulatory phonology by Browman and Goldstein [10] and the gestural control model by Kröger [15], we regard gestures as characterizations of discrete articulatory events that unfold during speech production in terms of goal-oriented articulatory movements. However, the actual characterization of these events differs from the aforementioned approaches and will be discussed later. After a gestural score has been specified, it is transformed into sequences of *motor commands* – one sequence for each parameter of the glottis and the vocal tract model. The execution of the motor commands, i.e. the generation of the actual articulatory trajectories, is simulated by means of third order linear systems. These systems were designed to produce smooth movements similar to those observed in EMA signals. The movements are directly generated in terms of time-varying parameter values for the vocal tract and the glottis. They determine the shape of the vocal tract and the state of the glottis which are the input to the aeroacoustic simulation generating the speech output.

This article is organized as follows. In Section 2, the components in Figure 1 will be described in more detail, in particular the models for the vocal tract and the glottis, the specification

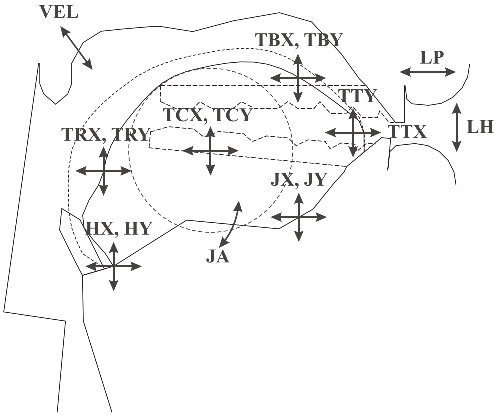


Figure 2: Schematic overview of the parameters of the vocal tract model and the articulatory structures that they control.

of gestural scores, and their transformation into speech movements. Section 3 presents the concepts for the high level control of the synthesizer, i.e. the generation of gestural scores from text using BOSS on one hand, and from timing information extracted from EMA tracks on the other hand. In Section 3.3 we discuss application prospects for the presented synthesizer. Conclusions are drawn in Section 4.

2. Articulatory speech synthesizer

2.1. Models for the vocal tract and the glottis

Vocal tract model. The vocal tract model of the synthesizer is a three-dimensional wire frame representation of the surfaces of the articulators and the vocal tract walls of a male speaker [3, 4]. The shape and position of all movable stuctures is a function of 23 adjustable parameters. Figure 2 shows the midsagittal section of the 3D vocal tract model along with the most important parameters. The arrows indicate how the corresponding parameters influence the articulation. Most of these parameters come in pairs and define the position of certain structures directly in Cartesian coordinates in a fixed frame of reference. For example, the point defined by the parameters (TCX, TCY) specifies the position of the tongue body (represented by a circle), (TTX, TTY) defines the position of the tongue tip, and (JX, JY) the position of the jaw. Therefore, the temporal change of these parameters should be comparable to the movement of pellets glued to the tongue or mandible in real articulations, as measured by EMA devices. The parameter values that best represent the ideal articulatory target shapes for German vowels and consonants have recently been determined by means of magnetic resonance images (MRI) [4]. The articulatory targets for consonants represent the vocal tract shape at the time of the maximum constriction, uttered without a specific phonetic context. However, it is well known that the actual articulatory realization of consonants strongly depends on the phonetic context. Only a few articulators (or parts of them) are really involved in the formation of the consonantal constriction while others are subject to coarticulation with adjacent phones. For example, the [g] in [igi] is realized differently from the [g] in [ugu]. In both cases, the tongue body is raised to make a palatal closure, but it is clearly more anterior in the context of the front vowel [i] than in the context of the back vowel [u]. In our synthesizer, such coarticulatory differences are handled by means of a dominance model. This model specifies a dominance value or “degree of importance” for each vocal tract parameter of each consonant. A high dominance means that a certain parameter is important for the formation of the consonantal constriction, and a low dominance value means that it is not important and therefore subject to coarticulation. In the above example for the consonant [g], the parameter TCY for the height of the tongue body has a high dominance, but TCX for its horizontal position a low dominance. The actual target parameter value $x_{c|v}[i]$ of a parameter i for a consonant c in the context of a vowel v at the moment of maximum constriction/closure is expressed as

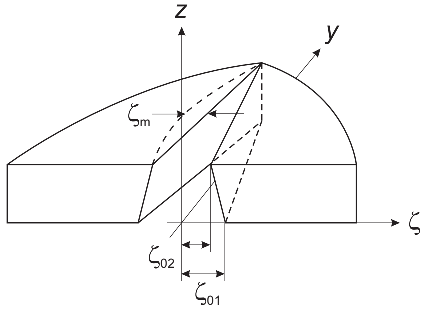


Figure 3: Model for the glottis based on Titze [19].

$$x_{c|v}[i] = x_v[i] + w_c[i] \cdot (x_c[i] - x_v[i]), \qquad (1)$$

where $w_c[i]$ is the weight (dominance) for parameter i, and $x_c[i]$ and $x_v[i]$ are the parameter values of the ideal targets for the consonant and vowel. The optimal dominance values for all parameters of all consonants have been determined in a previous study [4]. It was also shown that this simple dominance model is capable of reproducing the major coarticulatory differences in the realization of consonants.

Vocal fold model. For the voiced excitation of the synthesizer, we implemented a parametric model of the glottal geometry based on the proposal by Titze [19]. A schematic representation of the model is shown in Figure 3. The vocal fold parameters are the degree of abduction at the posterior end of the folds at the lower and upper edge (ζ_{01} and ζ_{02}), the fundamental frequency F_0, the phase difference between the upper and lower edge, and the subglottal pressure. Based on these parameters, the model generates the time-varying cross-sectional areas at the glottal inlet and outlet opening. We extended Titze’s original model to account for a smooth diminishment of the oscillation amplitude with increasing abduction [2] and for a parametrization of glottal leakage similar to [11].

Combination of the models. The geometric models of the vocal folds and the vocal tract are transformed into a combined area function. This area function, supplemented with the area functions of the subglottal system and the nasal cavity, serve as input to a time domain simulation of the flow and acoustics in the vocal system, producing the actual speech output [2, 1].

2.2. From gestural scores to speech movements

The intermediate representation layer for an utterance in the synthesizer is a gestural score. It defines an utterance in terms of an organized pattern of articulatory gestures. The specification and execution of these gestures differs, however, from previously proposed gestural control concepts (e.g., Browman and Goldstein [10], and Kröger [15]).

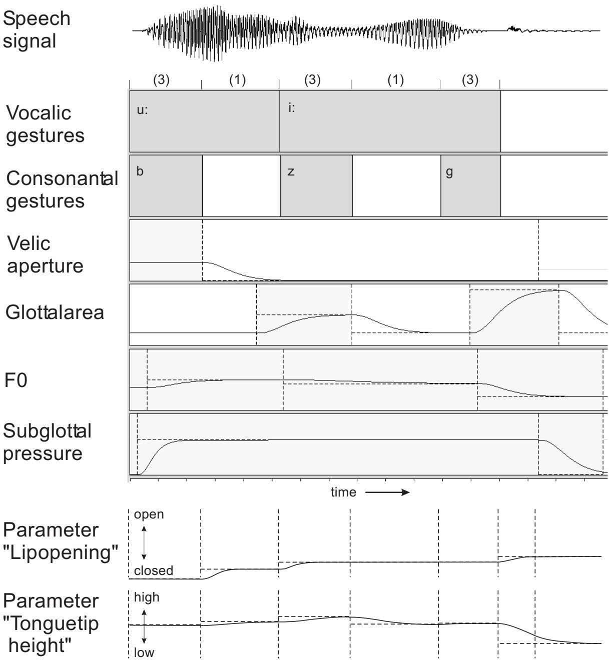


Figure 4: Gestural score for the utterance [muːziːk] with the generated speech waveform (top) and the resulting targets and their execution for two of the vocal tract parameters (bottom).

Figure 4 shows a gestural score for the utterance [muːziːk]. This example will illustrate the following explanations of the model. We differentiate between six types of gestures. Each row in Figure 4 contains the gestures of one type. The gestures in the first two rows are *vocalic* and *consonantal* gestures. Together with the *velic* gestures (third row) they determine the parameters of the vocal tract model, i.e., the supralaryngeal articulation. The gestures in the remaining rows control the glottal rest area (degree of abduction), the F_0, and the subglottal pressure. They determine the parameters of the model of the vocal folds, i.e., the laryngeal articulation. Each gesture has a certain temporal activation interval (defined by the vertical boundary lines) and is associated with a target for one or more vocal tract parameters or laryngeal parameters.

Let us first turn towards the supraglottal articulation. In Figure 4, the first vocalic gesture is associated with the target configuration for the vowel [uː], and the second one is associated with the vowel [iː]. The fixed target configurations were determined a priori for each vowel, as discussed in Section 2.1. The consonantal gestures in Figure 4 are associated with the consonants [b], [z] and [g]. We must point out that the target configuration for consonants with the same place of articulation are represented by only one configuration for each group. The groups {[b],[p],[m]}, {[d],[t],[n]}, and {[g],[k],[ŋ]} are represented by the target configurations for [b], [d], and [g], respectively. The voiceless plosives and the nasals are assumed to differ from the voiced plosives only in the state of the velum and the glottal area, which can be controlled individually in the gestural scores. Also the supraglottal articulation of voiced and voiceless fricatives with the same place of articulation is represented by only the voiced cognates. In Figure 4, the intervals for [b], [z], and [g] overlap with the intervals for the vowels [uː] and [iː]. This means that these consonants are coarticulated with the corresponding vowels. All vocalic and consonantal gestures are associated with an articulatory effort parameter. This effort translates into the transition speed towards the associated targets during the execution of the gestures.

But how are the vocalic and consonantal gestures executed, i.e., how are they transformed into the time-varying vocal tract parameter functions? First, a sequence of motor commands is generated for each parameter. In the context of this control model, a motor command is defined as target value for a vocal tract parameter within a defined time interval. Below the gestural score in Figure 4, these sequences of target values are shown for the lip opening LH and the vertical tongue tip position TTY by means of horizontal dashed lines. An individual motor command is generated for each combination of a vocalic and a consonantal gesture. The motor command boundaries are indicated by vertical dotted lines. The actual target value associated with a motor command for a vocal tract parameter depends on the underlying gestures. We differentiate between three cases: (1) The target value is that for a vowel. (2) The target is that for an isolated consonant. (3) The target is that for a consonant coarticulated with a vowel calculated according to Equation (1).

In Figure 4, we have only the cases (1) and (3), which are marked accordingly on top of the gestural score. In this way, a sequence of motor commands is calculated for each vocal tract parameter. The only exception is the parameter for the velic aperture, which is controlled separately by the velic gestures. These gestures directly correspond to the motor commands for the parameter *VEL* (cf. Figure 2).

The execution of the motor commands is modeled by means of a critically damped dynamical third order linear system with the transfer function

$$H(s) = 1/(1+\tau s)^3, \qquad (2)$$

where s is the complex frequency and τ is a time constant to be described later. The input to the system is the sequence of targets for a certain parameter. The system's output is the time dependent function value for that parameter. For the parameters LH and TTY, the resulting functions are drawn as solid lines below the gestural score in Figure 4. Note that the systems behave in such a way that the vocal tract parameters successively approximate the target values associated with the motor commands. In other words, they implement the original articulatory gestures as goal-oriented movements. The parameter τ in Equation (2) is a measure for the speed of target approximation. A small value for τ corresponds to a fast movement, and vice versa. The τ parameters for the individual motor commands are derived from the articulatory effort parameters for the vocalic and consonantal gestures. Therefore, τ can vary for adjacent motor commands.

As stated before, the parameter for the velic aperture of the vocal tract model is controlled independently from the other supraglottal parameters by means of velic gestures. The velic gestures directly define the target positions for motor commands, which are executed in the same way as described above. Similarily, the gestural targets for the glottal rest area, F_0, and the subglottal pressure defined in the gestural score are directly mapped on motor commands for the corresponding parameters of the model of the vocal folds.

A more detailed description of the gestural control model and the underlying ideas can be found in [6].

3. High level control concepts

3.1. Bonn Open Synthesis System (BOSS)

The Bonn Open Synthesis System (BOSS) [8] is a developer framework for the design of unit selection speech synthesis applications in C++. Its main goal is to relieve researchers in the field of speech synthesis of the need to implement their own systems from scratch. It is available under the GPL open source license from the IfK website [9]. BOSS is designed to be used as a client/server application over a network. Most of the symbolic preprocessing, the selection of units and their concatenation and manipulation are performed by the server while the client software is responsible for text normalization and tokenization and for encoding this information into the XML vocabulary understood by the server. By this choice of design, BOSS can be flexibly employed for either CTS or TTS, depending on what type of client is used. The core class of the BOSS server, also called the module scheduler, processes the client-generated information sentence by sentence. Required modules are loaded dynamically upon initialization of the scheduler class. The names and calling order of module libraries are defined in a configuration file, so that a developer who wishes to adapt BOSS to a new language or application is not required to change the source code of the server software. For the application described in this paper, we used the German transcription module, the CART [7] duration prediction module and the Fujisaki-based [13] intonation module delivered with the BOSS distribution. In summary, these modules provide the phonetic transcription (structured into syllables and phones) of a German input text with a duration specification for each phone, and Fujisaki-based intonation commands for each syllable. In the following, we will discuss a proposal how to translate this information into a gestural score for the articulatory synthesizer.

The major problem in this context is to translate the phone durations given by BOSS into activation intervals of the gestures, especially of the vocalic, consonantal, velic and glottal gestures. BOSS predicts the phone durations corresponding to the conventional way of phone segmentation, i.e. the beginning and the end of phones is associated with striking landmarks in the auditory signal or the spectrogram. In this sense, the consonant [t], for example, starts where the acoustic signal energy suddenly drops due to the apico-alveolar closure and ends after the aspiration phase following the release of the closure. In general, these acoustical landmarks can be assigned to special *articulatory* events that are also reflected in the gestural scores. Furthermore, each class of phones exhibits typical patterns of temporal coordination of the involved articulatory gestures, such as the coordination between the constriction forming gesture (consonantal gesture) and the glottal abduction gesture for voiceless plosives. These patterns are sometimes called "phasing rules" [10, 15]. The phasing rules, together with the associations between acoustical landmarks and time instants in the gestural scores allow to calculate phone durations from gestural scores, and vice versa, to create gestural constellations for phones of a given class and with a given duration.

Figure 5 illustrates the phasing rules and the correspondence between gestural constellations and the resulting speech waveform for plosives, fricatives, and nasals. The consonants in these examples were embedded into the context [iːCaː]. First of all, the consonantal gestures were always aligned to be coarticulated with the vowel of the second syllable, according to Xu [20]. The time intervals of consonantal closure (or critical constriction in the case of [s]) are marked by vertical dashed lines. Typically, these intervals start 30–60 ms after the onset of the consonantal gestures. This is the time the constriction forming articulators need to reach their target positions. The ends of the constriction/closure intervals are typically very shortly after the offset of the consonantal gestures, where the articulators start moving towards their targets for the following vowels. For [iːdaː], [iːsaː], and [iːnaː] (and the corresponding classes of consonants), the constriction intervals directly correspond to the phone durations according to the BOSS predictions. However, for voiceless aspirated plosives as the [t] in [iːtaː], BOSS does not predict the constriction duration, but the duration from the onset of the closure to the end of the burst and aspiration phase. In the gestural score, this end point is roughly where the glottal aperture is reduced to 50% of its maximal area.

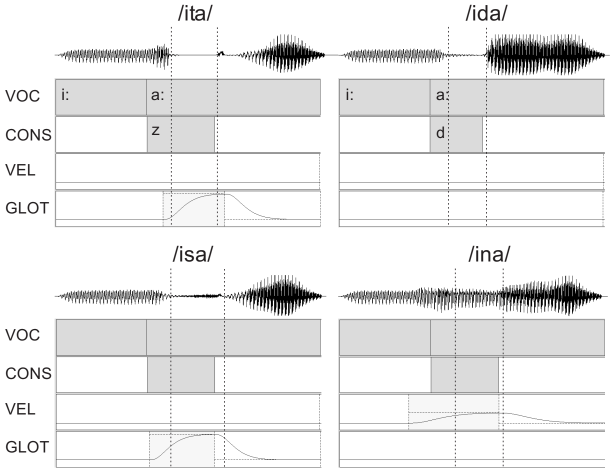


Figure 5: Gestural constellations for voiced and voiceless plosives, voiceless fricatives, and nasals in the context [iːCaː]. *VOC*=vocalic gestures, *CONS*=consonantal gestures, *VEL*=velic gestures, and *GLOT*=glottal gestures. The vertical dotted lines indicate the beginnings and ends of the consonantal closure/constriction intervals. The gestures for subglottal pressure and F_0 are not shown.

The velic and glottal gestures in Figure 5 illustrate appropriate phasing rules for the different classes of consonants. Voiced plosives need neither a velic nor a glottal gesture. For voiceless aspirated plosives, glottal abduction should approximately start at the beginning of the closure interval [18]. To get a fair amount of aspiration, glottal adduction should start approximately by the end of the oral closure interval. For voiceless fricatives, the glottal gesture should start and end roughly simultaneously with the consonantal gesture to produce good synthetic results. Nasals need a lowering of the velum by means of a velic gesture. Preliminary synthesis results suggest that the onset and offset of the velic aperture is not very critical. For [iːnaː] in Figure 5, we made the velic gesture start shortly before the corresponding consonantal gesture and end simultaneously with it. Similar rules can easily be established for voiced fricatives, laterals, glottal consonants, and the generation of consonant clusters. The duration of vowels and diphthongs is determined by the borders of the adjacent consonants.

This section was mainly meant to illustrate basic ideas for the rule-based creation of gestural scores from a given phonetic transcription and phone durations. A quantitative implementation of these rules is in progress, and first speech examples will be presented at the conference. To improve the naturalness of the synthetic utterances, a prototypical transformation from

BOSS intonation commands to gestures for F_0 control will also be implemented.

3.2. Speech resynthesis based on EMA data

The duration of predicted parameters (both segmental and suprasegmental) using conventional TTS "preprocessing" is based on observations of acoustic landmarks in speech. In articulatory synthesis, we must predict the movements of the articulators which cause these landmarks, after a certain delay. To analyze and directly implement this delay in an articulatory synthesizer, we must first study the actual movements of the articulators during speech production. One possibility of doing this is through Electromagnetic Articulography (EMA).

For the analysis of articulatory parameters during actual speech production, we were given access to two EMA corpora ([12], [14]). The first of these contains recordings of a female German speaker uttering /CVCVCVCV/ sequences, with all combinations of a set of 9 consonants and 15 vowels of German, in two conditions (EMA sensors: jaw, lower and upper lip, tongue tip, blade and dorsum). The second corpus consists of recordings of 7 German speakers (1 female, 6 male) uttering /CVC/ syllables embedded in a carrier phrase, with all combinations of 3 consonants and 14 vowels, in two conditions, as well as reading a list of 108 German sentences (EMA sensors: jaw, lower lip, tongue tip, blade, dorsum, and back).

The aim of an intermediate study is to resynthesize the utterances of the recorded speakers, comparing the trajectories of the articulatory parameters. Since the virtual vocal tract is modeled upon that of one speaker and the natural data obtained from another, a direct comparison of raw articulator movements does not make sense. Rather, the *timing* of the simulated EMA trajectories produced by the synthesizer is modeled on the temporal structure of articulatory gestures performed by the original speaker, and thereby indirectly on his speech rhythm.

While it could in theory be possible to directly transfer the EMA trajectories to the virtual articulators (normalized for differences in anatomy) and produce similar, if not identical utterances, such a low-level approach is not the goal of an articulatory synthesizer with high-level control mechanisms. Rather, the purpose of this resynthesis is twofold: to test the parametric fidelity of the synthesizer; and to analyze the observed delay from gestural onsets to the acoustic landmarks traditionally regarded as the beginning of the corresponding segment in the synthesis output.

For a preliminary comparison of natural and synthetic articulatory trajectories, the word *Methanol* [meta'noːl] was resynthesized, using EMA parameters of one of the male speakers. The resynthesis process involved two steps: first, identifying intervals in which the relevant EMA trajectories approached the respective target values; and second, providing this timing information to the synthesizer in the form of a gestural score. Additionally, the F_0 contour was extracted from the acoustic signal and included in the gestural score in a smoothed form. The resulting synthesis output is presented alongside the original recording in Figure 6. The relative height and arrangement over time of the peaks and valleys in these curves displays an encouraging similarity. One should keep in mind that our aim was not to produce an exact copy of the trajectories, but to combine the gestural targets of the virtual vocal tract with timing derived from EMA data, creating the desired perceptual impression.

In addition to gestural timing, it is conceivable to extract measures of articulatory effort from the EMA trajectories and include these in the gestural score, since the synthesizer allows fine control over this parameter.

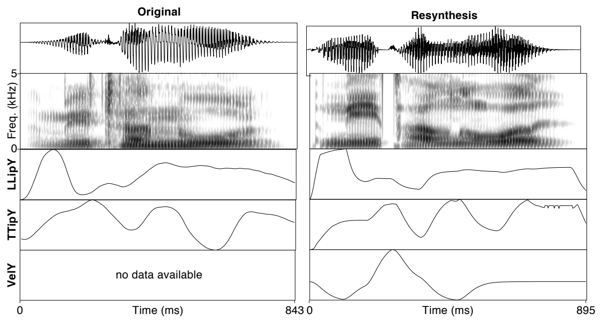


Figure 6: Gestural constellations for original (left) and resynthesized (right) version of the word *Methanol* [meta'noːl]. Below the spectrograms are the normalized trajectories of the parameters corresponding to height of the lower lip (*LLipY*), tongue tip (*TTipY*), and velum (*VelY*).

3.3. Application prospects

Combining high-level articulatory control with natural-sounding synthesis breaks out of the widely-accepted compromise that naturalness and parametric flexibility are inversely correlated in speech synthesis and cannot both be satisfied at once. This opens up many new opportunities for a variety of applications for the presented system. A few immediate prospects are outlined below, but listing all the possibilities would be well outside the scope of this paper.

Considerable naturalness can already be achieved with unit-selection and similar synthesis approaches (especially in a limited domain), but at the cost of prosodic control. In fact, many unit-selection synthesis platforms currently choose to abandon explicit prosody modeling altogether and therefore lack control over parameters such as F_0. Those that do allow F_0 target specification (either through the unit selection algorithm itself or subsequent signal manipulation) may introduce significant artifacts in an unpredictable way, depending on whether or not suitable units can be found in the unit-selection corpus.

Expressive speech synthesis. One possible area of application for an articulatory synthesizer with full flexibility and high naturalness is of course expressive (a.k.a. "emotional") speech synthesis (cf. [16] for a detailed survey). This expanding field of speech synthesis relies heavily on flexible control over prosodic and/or paralinguistic parameters, mainly F_0, but also voice quality, among others. For this reason, expressive speech synthesis has largely been unable to make use of the progress in unit-selection approaches, being forced to rely instead on less natural-sounding, but more flexible diphone concatenation or formant synthesis.

Certain other relevant parameters, such as voice register, articulatory effort, lip spreading, etc. can only be controlled with elaborate effort, if at all, using the synthesis methods mentioned above. The system presented here, however, is ideally suited to such tasks and can be extended to provide high-level control over precisely such parameters.

Multilingual speech synthesis. With a certain amount of adjustment, the presented system could easily be adapted to new languages, the phoneset being, after all, a set of gestural "macros". The resulting synthesis output would be in the same voice as long as the vocal tract characteristics remain unchanged. This would allow true multilingual synthesis without depending on necessarily distinct native speaker recordings.

Voice morphing. On the other hand, vocal tract characteristics could be deliberately modified to create a different voice. This allows control over gender, age, timbre, as well as a multitude of other extralinguistic parameters. Since all synthesis output is rendered to an acoustic signal only once, no degradation of quality occurs, as is inevitable with voice morphing techniques and similar signal processing. The presented system provides full control over numerous physiological properties of the synthesis voice, permitting finely detailed voice design for e.g. artificial agents in dialog systems.

Prosody research. Phonetic research in prosody would benefit greatly from an instrument allowing at leisure the synthesis of natural-sounding, prosodically fully-flexible speech. This would provide the means to e.g. implement and test autosegmental phonological models, generate high-quality stimuli for experiments, and much more. Currently, many synthetic stimuli created for prosody experiments suffer from limited naturalness, depending on the synthesis technique used to produce them, for the same reasons as outlined above under *expressive speech synthesis.* Whereas in a (commercial) TTS system, intelligibility takes precedence over naturalness, in prosodic experiments, a lack of naturalness may distract test subjects and affect their responses, skewing the results of the study.

Nevertheless, it must be acknowledged that the computational complexity of articulatory synthesis as implemented in the presented system currently prevents synthesis in realtime on an average desktop PC. It is our belief, however, that realtime synthesis will become realistic in the very near future, owing to advances in processing power as well as code optimization.

4. Conclusions

We have presented two concepts for the high-level control of an articulatory speech synthesizer. First, we outlined rules for the transformation of phonetic transcriptions and phone durations predicted by the Bonn Open Synthesis System (BOSS) into gestural scores, extending the synthesizer to a text-to-speech system. Second, we demonstrated the generation of gestural scores based on EMA signals. Our preliminary results suggest that both ways lead to well intelligible synthetic speech.

For future research, it is conceivable to train BOSS to directly predict gestural parameters, e.g. gestural durations, instead of phone durations in the conventional sense, as it currently does. This would considerably simplify the rules for the generation of gestural scores, but would require a corresponding segmentation of the original EMA data.

5. Acknowledgments

This research was partially funded by the German Research Foundation (DFG) with the grant JA 1476/1-1. We would like to thank Sascha Fagel and Phil Hoole for making their EMA data available to us.

6. References

[1] P. Birkholz and D. Jackèl, “Influence of temporal discretization schemes on formant frequencies and bandwidths in time domain simulations of the vocal tract system,” in *Interspeech 2004-ICSLP*, Jeju, Korea, pp. 1125–1128, 2004.

[2] P. Birkholz, “3D-Artikulatorische Sprachsynthese,” Ph.D. dissertation, University of Rostock, 2005.

[3] P. Birkholz, D. Jackèl, and B. J. Kröger, “Construction and control of a three-dimensional vocal tract model,” in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*, Toulouse, France, pp. 873–876, 2006.

[4] P. Birkholz and B. J. Kröger, “Vocal tract model adaptation using magnetic resonance imaging,” in *7th International Seminar on Speech Production (ISSP'06)*, Ubatuba, Brazil, pp. 493–500, 2006.

[5] P. Birkholz, D. Jackèl, and B. J. Kröger, “Simulation of losses due to turbulence in the time-varying vocal system,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1218–1226, 2007.

[6] P. Birkholz, “Control of an articulatory speech synthesizer based on dynamic approximation of spatial articulatory targets,” *submitted to Interspeech 2007 - Eurospeech*, Antwerp, Belgium, 2007.

[7] L. Breiman, J. Friedman, R. Olshen and C. Stone, *Classification and Regression Trees*, Wadsworth International, Belmont, CA, 1984.

[8] S. Breuer, P. Wagner, J. Abresch, J. Bröggelwirth, H. Rohde and K. Stöber *Bonn Open Synthesis System (BOSS) 3 Documentation and User Manual,* `http://www.ikp.uni-bonn.de/boss/BOSS_Documentation.pdf`, 2005.

[9] `http://www.ikp.uni-bonn.de/boss`

[10] C. P. Browman and L. Goldstein, “Articulatory phonology: An overview,” *Phonetica*, vol. 49, pp. 155–180, 1992.

[11] B. Cranen and J. Schroeter, “Modeling a leaky glottis,” *Journal of Phonetics*, vol. 23, pp. 165–177, 1995.

[12] S. Fagel, *Audiovisuelle Sprachsynthese: Systementwicklung und -bewertung.* Logos Verlag, Berlin, 2004

[13] H. Mixdorff, “Intonation Patterns of German - Model-based Quantitative Analysis and Synthesis of F_0 contours,” Ph.D. dissertation, TU Dresden, 1998

[14] P. Hoole and C. Mooshammer, “Articulatory analysis of the German vowel system”, In: Auer, P., Gilles, P. & Spiekermann, H. (eds.), *Silbenschnitt und Tonakzente.* Niemeyer, Tübingen, pp. 129–152, 2002.

[15] B. J. Kröger, *Ein phonetisches Modell der Sprachproduktion.* Niemeyer, Tübingen, 1998.

[16] M. Schröder, “Approaches to emotional expressivity in synthetic speech,” in K. Izdebski (ed.), *Emotions in the Human Voice*, vol. 3, 2007.

[17] C. H. Shadle and R. I. Damper, “Prospects for articulatory synthesis: A position paper,” in *Fourth ISCA Tutorial and Research Workshop on Speech Synthesis*, Pitlochry, Scotland, pp. 121–126, 2001.

[18] K. N. Stevens, *Acoustic Phonetics.* MIT Press, Boston, 1998.

[19] I. R. Titze, “Parameterization of the glottal area, glottal flow, and vocal fold contact area,” *Journal of the Acoustical Society of America*, vol. 75, no. 2, pp. 570–580, 1984.

[20] Y. Xu and F. Liu, “Tonal alignment, syllable structure and coarticulation: Toward an integrated model,” *Italian Journal of Linguistics (in press)*, 2007.

Spectral Control in Concatenative Speech Synthesis

Alexander Kain, Qi Miao, Jan P. H. van Santen

Center for Spoken Language Understanding (CSLU)
OGI School of Science & Engineering
Oregon Health & Science University (OHSU)
20000 NW Walker Road, Beaverton, OR 97006, USA

Abstract

We report on research in which we increased the degree of spectral control in concatenative synthesis by controlling the formant frequencies of the synthetic speech, as well as the energies in four spectral bands. In addition, we eliminated "points" of concatenation in favor of "regions" of concatenation, by *cross-fading* between the end and the beginning of two speech segments that are part of a concatenation operation. We hypothesized that these approaches would decrease the frequency and severity of audible discontinuities in the synthetic speech and thus also increase the perceived quality of the speech. A listening test determined that stimuli created with the proposed methods resulted in significantly increased quality.

1. Introduction

In the process of generating audible speech from a textual representation, a text-to-speech (TTS) system first converts text into a linguistic representation, which is then used to generate an appropriate acoustic waveform. This second step is achieved by using a speech synthesis model that describes the relationship between linguistic units and acoustic features. These speech synthesis models vary in their complexity. The first intelligible synthesizers used an approach called *formant synthesis*, which utilizes relatively simple models of the glottal source and vocal tract. Model parameters can be generated either by rule [1] or from a database [2]. Most aspects of speech are controllable, including the degree of articulation and characteristics of the speaker. The resulting speech is highly intelligible, but is often judged as not very natural. In an effort to increase naturalness without decreasing flexibility, researchers have increased the complexity of the speech synthesis model to take into account more physiological and physical details about the speech production process; this approach is called *articulatory synthesis* [3]. Unfortunately, it is proving difficult, in practice, to generate the high-dimensional parameter trajectories necessary to drive articulatory synthesis models, because the relationships between linguistic units and parameter trajectories are complicated and cannot be learned easily. Both formant and articulatory synthesis are examples of *parametric synthesis*.

The most successful TTS approach to-date is called *concatenative synthesis*; in this approach, natural speech utterances of a single speaker must first be recorded and stored in an *acoustic inventory*. During synthesis, individual portions of speech are retrieved from the inventory, optionally modified, and then concatenated in the desired sequence. In the *unit-selection* approach, the relationship between linguistic units from text-processing and acoustic units of the acoustic inventory is established by means of a search, which, given a sequence of target linguistic units, optimizes (1) the fit between chosen linguistic units and the target linguistic units, also known as *target cost*, and (2) the fit between the chosen consecutive units, usually in the acoustic domain, known as *concatenation cost*. Intelligibility and naturalness are very high in the concatenative synthesis approach [4]. However, output speech is limited by the contents of the acoustic inventory (not just the linguistic content, but also the emotional state of the speaker, degree of articulation, etc.), and inevitable concatenation errors can lead to audible discontinuities. To overcome the problems of limited content and discontinuities, researchers either significantly increase the size of the database to include more variability, or introduce additional modeling to modify and thus control the natural speech signal. In the latter case, models that include prosodic control of pitch and duration are common [5]. In addition to prosodic modifications, researchers have also proposed spectral modifications, for example smoothing spectral balance discontinuities at concatenation points, expressed as energies in four bands [6], smoothing formant discontinuities [7, 8, 9], and controlling the degree of articulation [10].

It is our long-term goal to combine parametric and concatenative synthesis methods to achieve highly flexible and natural speech, by researching data-driven speech models and high-quality speech modification algorithms. In this paper, we report on experiments involving modification of formant frequencies, spectral balance, and time-domain waveforms, using speech units selected by the concatenative approach. We eliminated "points" of concatenation in favor of "regions" of concatenation, by *cross-fading* (i. e. fading out one signal while fading in another) in various domains between the end and the beginning of two speech segments adjoining a concatenation. We hypothesized that this approach would decrease the frequency and severity of audible discontinuities in the synthetic speech and thus increase the perceived quality of the speech.

Section 2 introduces the methods to analyze and construct formant frequency trajectories, and to implement the necessary changes to the speech signal. Section 3 describes a perceptual test designed to validate our hypothesis, and we conclude in Section 4.

2. Methods

The key goal of our proposed approach is to decrease the negative effects of unnatural discontinuities between two speech segments that are part of a concatenation operation. We aimed to achieve this decrease by explicitly controlling the first three formant frequencies and the energies in four spectral bands

This work was supported by NSF grant #0313383 "Objective Methods for Predicting and Optimizing Synthetic Speech Quality".

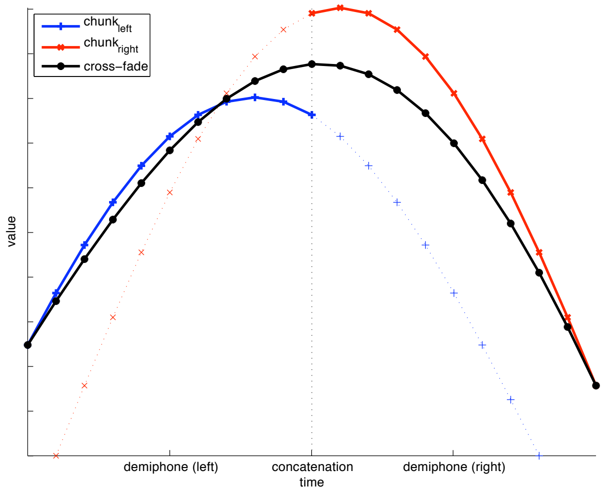


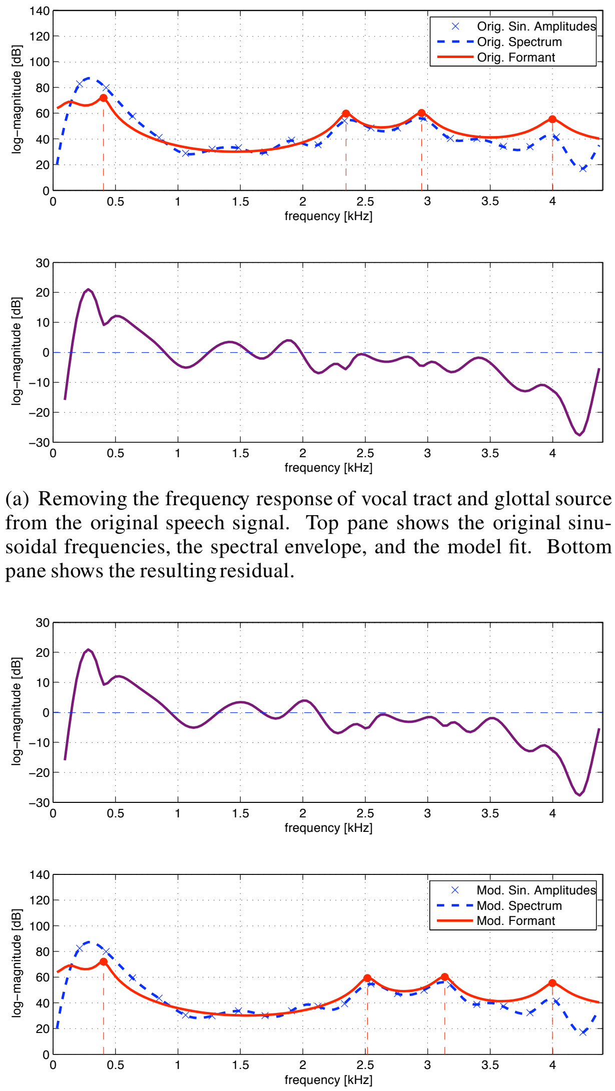
Figure 1: Cross-fading across a region of concatenation, using a fictitious 1-dimensional example feature. Without cross-fading, the final trajectory would be the concatenation of the solid left half-curve with the solid right half-curve, resulting in a large discontinuity. With cross-fading, the following demiphone of the left chunk and the previous demiphone of the right chunk are combined, resulting in a smooth final trajectory as indicated by the continuous curve.

of the synthetic speech. Although a predictive model of how these features, given linguistic targets, may evolve over time exists [11], we initially chose a cross-fading approach of natural formant frequencies and natural four-band spectral energies in the acoustic inventory to achieve smooth synthetic trajectories. We then modified the selected speech segments in accordance with the cross-faded feature trajectories. Even after controlling for formant frequencies and spectral balance, we expected remaining, unaccounted-for differences in the two speech segments to be joined. Therefore, a final time-domain cross-fade was employed, which cross-faded the waveforms of the two modified segments.

2.1. Acoustic Inventory and Feature Analysis

Our acoustic inventory consisted of the CSLU's TLL diphone database, used in related previous experiments [12]. Typically, a diphone database only contains "chunks" (contiguous speech segments containing one or more speech units) of the type $a_R - b_L$, where a_R is the demiphone corresponding to the right-hand side of a phoneme a, and b_L is the demiphone corresponding to the left-hand side of the following phoneme b. To accommodate cross-fading in the formant frequency, spectral band, and time domains, we extended our analysis one demiphone to the left and one to the right, analyzing and storing chunks of the type $a_L - a_R - b_L - b_R$, equivalent to two full phonemes for each possible phoneme combination.

For each of the 1733 two-phoneme chunks in the acoustic inventory, we automatically extracted formant frequencies, as well as amplitudes and phases of harmonic sinusoids. We calculated energies in four discrete spectral bands (0–800 Hz, 800–2500 Hz, 2500–3500 Hz, and 3500-8000 Hz) by integrating the corresponding harmonic amplitudes [13, 6]. Formant frequency trajectories in vowel regions (the focus of the perceptual experiment in Section 3) were manually verified and corrected when necessary, using a standard labeling tool in conjunction with a pen input device.



(a) Removing the frequency response of vocal tract and glottal source from the original speech signal. Top pane shows the original sinusoidal frequencies, the spectral envelope, and the model fit. Bottom pane shows the resulting residual.

(b) Creating a new spectrum. Top pane shows the warped residual. Bottom pane shows the frequency response of the new model, as well as the recombination of that model with the warped residual.

Figure 2: Formant frequency modification.

2.2. Feature Trajectory Construction

As mentioned previously, we aim to reduce concatenation errors by constructing smooth feature trajectories in the formant frequency and spectral balance domains, and then modifying the natural speech signal accordingly. The construction of the feature trajectory was implemented by cross-fading the acoustic features of each speech frame across the entire phoneme that is involved in the concatenation operation (we ignored atypical concatenations at phoneme boundaries). Specifically, we considered the demiphone that followed the previous chunk, and the demiphone that preceded the following chunk, giving us a double set of features over the entire phoneme region (features were stretched or compressed by linear interpolation to match durations). The desired smooth feature tra-

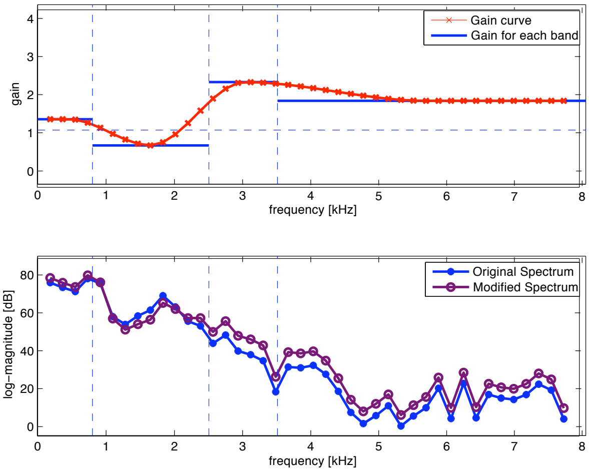


Figure 3: Spectral band modification. Top pane shows the desired amplitude gains for each individual band. To avoid discontinuities, a smooth gain curve is calculated. The bottom pane shows the original and modified sinusoidal harmonics and spectral envelopes.

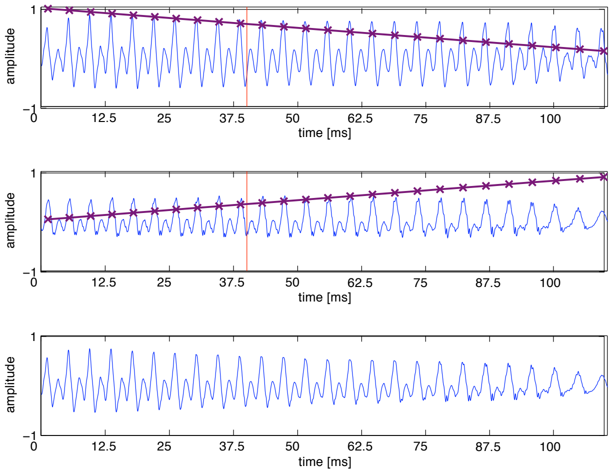


Figure 4: Time-domain cross-fade. Top pane shows left speech segment and middle pane shows right speech segment. Lines with ×markers represent cross-fade weights α and $1 - \alpha$. The traditional cutpoint is displayed as vertical lines. The bottom pane shows the cross-faded waveform.

jectories $\mathbf{s}(t)$ were calculated by applying the equation $\mathbf{s}(t) = \alpha(t)\cdot\mathbf{r}(t)+(1-\alpha(t))\cdot\mathbf{l}(t)$, where $\mathbf{l}(t)$ and $\mathbf{r}(t)$ are feature vectors at time $t = 1 \ldots N$ of the last demiphone of the left chunk and the first demiphone of the right chunk, respectively, N denotes the total number of datapoints in the cross-fade region, and α is the cross-fade function given by $\alpha(t) = t/(N + 1)$.

Figure 1 illustrates the concept using a fictitious trajectory. We implemented both formant domain cross-fading on the first three formant frequencies, and spectral balance cross-fading, using the energies in four spectral bands.

The approach just described has some parallels with a "fusion unit" strategy researched previously [14]; however, the differences are that our proposed approach modifies formant frequencies instead of line spectral frequencies, does not require a fusion unit, and operates on features directly, instead of on their derivatives.

2.3. Speech Modification and Synthesis

Speech was synthesized using a pitch-synchronous, frame-by-frame, overlap-add, harmonic sinusoidal system. During synthesis, both left and right natural segments from the acoustic inventory are modified in accordance with the smooth feature trajectories constructed as described in the previous section, first in the formant frequency (FFXF) and then in the spectral band domains (SBXF). Finally the two modified speech segments are cross-faded in the time-domain (TDXF), to smooth any remaining acoustic differences.

2.3.1. Formant Modification

The modification of formants has attracted attention by many researchers. Most studies focus on the so-called "pole interaction" problem, which refers to the problem of correctly associating formants with the roots of linear prediction coefficients (LPC). Once formants are identified, modification is carried out by changing LPC poles' angles and radii [15, 16], or direct modification of line spectral pairs [17, 18], usually followed by LPC synthesis. Researchers also proposed modifications in the mel-cepstrum domain using the STRAIGHT analysis and synthesis method [19]. Finally, another modification approach is based on a joint all-pole and sinusoidal model, wherein residual harmonics are warped in accordance with changes to the all-pole model, leading to improved speech quality [14].

In our work, we used a variation on the last approach. The pole-interaction problem did not exist in our case since reliable formant information was available. Figure 2 shows an example of increasing F2 and F3 formant frequencies. In a first step, we constructed an estimate of the frequency response of the speech signal by linearly combining the effects of the vocal tract and the glottal source. The vocal tract was modeled as an all-pole formant filter using the original, manually verified formants F1–F3 information from Section 2.1; in addition, we added higher formants with constant frequency and bandwidths. The glottal source was modeled using a frequency domain representation of a standard glottal flow model [18], with global glottal source parameters that were tuned for the TLL voice. Next, we subtracted the resulting frequency response from an upsampled and smoothed envelope of the individual harmonic sinusoids, as illustrated by Figure 2(a). Then, we frequency warped the resulting residual in accordance with the desired formant frequency changes, and recombined the modified residual with a new formant filter that reflects the desired changes to formant frequencies and bandwidths. Finally, we sampled the new spectral envelope at harmonic intervals to obtain the new sinusoidal parameters, as illustrated by Figure 2(b).

2.3.2. Spectral Band Modification

After formant modification, we calculate the 4-band spectral energies of the modified spectrum, compare the energy values to the desired cross-faded spectral band trajectories, and compute the required gains. For each frame, sinusoidal amplitudes are multiplied with the resulting gain function, after appropriate smoothing to avoid energy discontinuities at the band edges (see Figure 3).

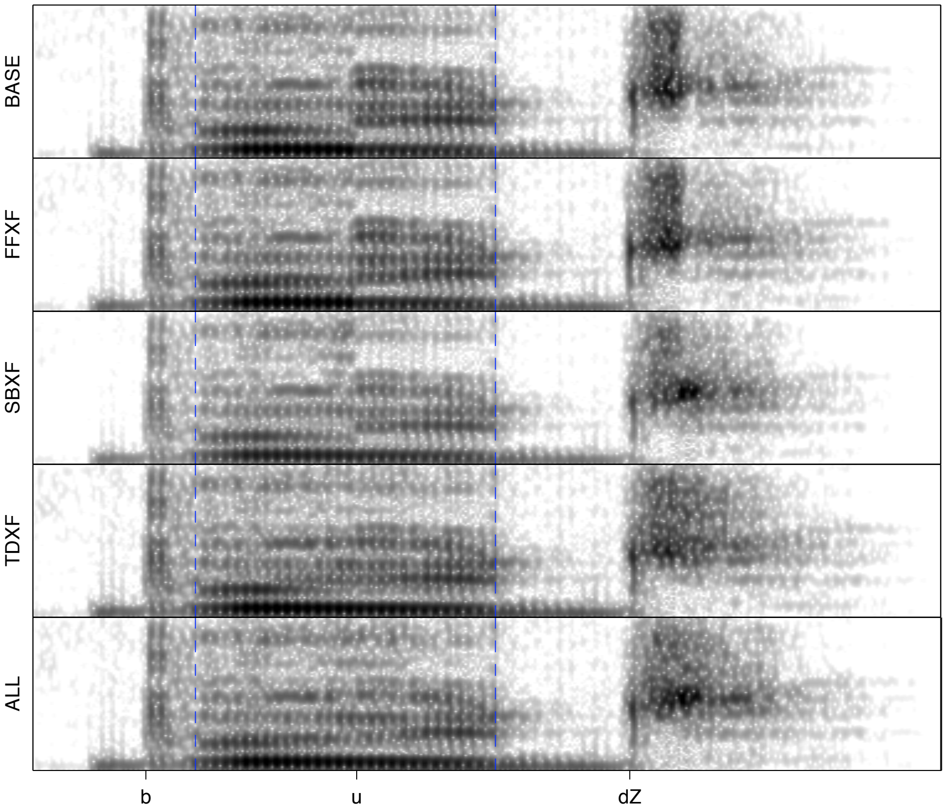


Figure 5: Spectrograms of the word /b u dZ/ in the five conditions. Dashed lines denote phoneme boundaries.

2.3.3. Time-domain Cross-fade

Despite best efforts to explicitly control speech parameters, in our case formant frequencies and spectral band energies, there are likely to be aspects of speech that remain unmodeled. During concatenation, a mismatch of those aspects may be heard as audible discontinuities. To address this problem, we used a time-domain cross-fade approach to make a smooth transition from one (already modified by the methods described above) chunk to the next. This approach required the synthesizer to produce parallel frames of speech with identical features, but from two distinct chunks, during regions of concatenation. We then linearly interpolated between these (pitch-synchronous) segments in the time domain, according to a cross-fade function similar to the one in Section 2.2 (see Figure 4). It should be noted that the TDXF approach implements global energy smoothing inherently.

3. Perceptual Experiment

3.1. Stimuli and Administration

To test the expected quality improvements over a baseline system (BASE), we ran a comparative mean opinion score (CMOS) listening test, using just one of the proposed approaches in isolation (FFXF, SBXF, and TDXF) or all of them jointly (ALL). Stimuli consisted of six vowels (two diphthongs /aI/ and /aU/, and four tense vowels /i:/, /@/, /u/ and /A/) in a consonant-vowel-consonant (CVC) context.

For each of the six vowels, we were interested in the interactions between given formant frequency or spectral band distances, and the approaches designed to smooth them. Therefore, we selected stimuli based on two distance types at the concatenation points, namely the formant distance, D_{FF}, and the spectral balance distance, D_{SB}. For each possible vowel concatenation in a $C_1 - V - C_2$ context in the acoustic inventory, we calculated the distances by applying equations

$$D_{FF}(V_L, V_R) = \sqrt{\sum_{k=1}^{3}(FF_{k,V_L} - FF_{k,V_R})^2}$$

and

$$D_{SB}(V_L, V_R) = \sqrt{\sum_{k=1}^{4}(SB_{k,V_L} - SB_{k,V_R})^2}$$

where V_L represents the left half of a vowel in a $C_1 - V$ context, V_R represents the right half of a vowel in a $V - C_2$ context, FF_{k,V_L} and FF_{k,V_L} represent the k^{th} formant frequencies (in Bark) at the concatenation point of V_L and V_R, and SB_{k,V_L} and SB_{k,V_R} represent the energies in the k^{th} spectral band (in dB) at the concatenation point of V_L and V_R.

After determining both D_{FF} and D_{SB} distances for all possible vowel concatenations, we normalized their values, and selected concatenations at the extremes of these distances, using the Euclidean distance to the four corners of the square spanned by the candidate data. This resulted in four stimulus types: large D_{FF} and large D_{SB}, large D_{FF} and small D_{SB}, small D_{FF} and large D_{SB}, and finally small D_{FF} and small D_{SB}, using the top and bottom 50% of the data for large and small, respectively. We repeated this process for all six vowels, using two concatenations per distance type, resulting in 48 (2 concatenations × 4 types × 6 vowels) different CVC words, some of them nonsensical.

Listener	FFXF	SBXF	TDXF	ALL
1	+0.04	+0.17	+0.40	+0.50
2	+0.40	+0.63	+0.77	+1.15
3	+0.08	+0.33	+0.67	+0.58
4	+0.10	+0.27	+0.38	+0.65
5	-0.08	+0.06	+0.31	+0.23
6	+0.10	+0.15	+0.58	+0.23
7	0.00	+0.27	+0.38	+0.42
8	+0.19	+0.54	+0.60	+0.67
Mean	+0.10	+0.30	+0.51	+0.56
SD	0.13	0.18	0.16	0.28

Table 1: Comparative mean opinion scores for the modified conditions, as compared to the BASE condition. Scores are averaged over all vowels with results shown for individual listeners, as well as the mean and standard deviation for averaged listener responses.

The selected CVC words were generated by an implementation of the proposed approaches in Section 2. We synthesized the selected CVC words under five different conditions: (1) no modifications were applied (BASE), (2) only formant frequency trajectories were cross-faded (FFXF), (3) only spectral band energy trajectories were cross-faded (SBXF), (4) only time-domain cross-fading was applied (TDXF), and (5) all cross-fading operations were applied (ALL). Note that the BASE condition performed a very short version of TDXF as part of the standard procedure of overlap-adding synthesis speech frames.

Each CVC word consisted of four chunks from the acoustic inventory (pause-C → C-V → V-C → C-pause), requiring three concatenation operations. Smoothing operations took place in all three concatenations, except that FFXF was not used when consonants were involved that lacked reliable formant information (such as unvoiced fricatives). We set vowel durations to their median values, as calculated from the acoustic inventory (130 ms for /i:/, 185 ms for /@/, 125 ms for /u/, 175 ms for /A/, 175 ms for /aI/, and 170 for /aU/). We used a naturally falling pitch contour with an average of 220 Hz for each CVC word.

Figure 5 shows spectrograms of the word /b u dZ/[1] in all five conditions. The following observations can be made: the vowel and the final consonant are quite discontinuous in the BASE condition. The FFXF condition "connects" the formants of the vowel smoothly (especially F2), but large energy differences remain. The SBXF condition smooths the energy transition in the vowel (this can be seen clearly for F3 and F4), but formant discontinuities remain; however, this condition smooths the final consonant very successfully. The TDXF condition can be seen to smooth the vowel transition by fading one speech unit out as it is fading another unit in; however, formants do not truly connect this way, and at the middle of the cross-fade there are formant "duplicates" (as can be seen by the presence of two F2 tracks towards the middle of the vowel). For the final consonant, TDXF performs an adequate smoothing. Finally, the ALL condition connects formants, equalizes the energy in the four spectral bands, and cross-fades any remaining discrepancies.

The final test stimuli contained pairs of identical CVC words in two different conditions, with a 200 ms separating pause. We compared all 4 modified conditions against the BASE condition, but ignored ordering effects, which resulted in 4 possible condition pairs and a total of 192 stimuli (48 CVC words × 4 condition pairs).

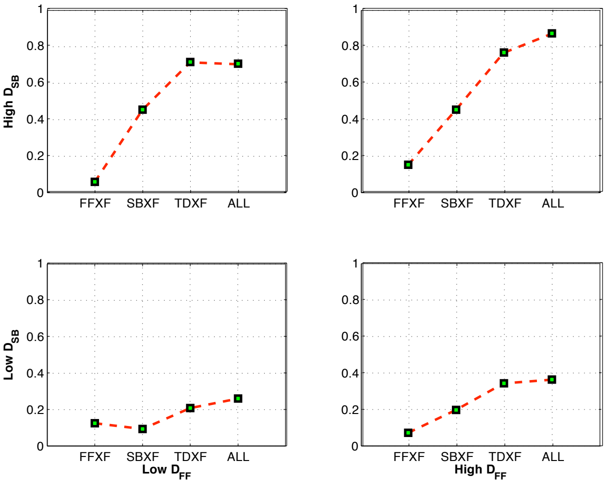


Figure 6: Comparative mean opinion scores for the four modified conditions as compared to the BASE condition, separated into the four stimulus types described in Section 3.1. Scores are averaged over all vowels and listeners.

We recruited 8 normal-hearing (self-reported) listeners, whose native language was American English. Listeners heard stimuli over circumaural headphones. Upon hearing the two words, they were asked to compare them based on quality and processing artifacts, using a scale of -2 (A is much better than B), -1 (A is slightly better than B), 0 (A and B are about the same), +1 (B is slightly better than A), and +2 (B is much better than A). The order of the conditions in a stimulus pair was randomized.

3.2. Results and Discussion

The CMOS values (preference scores) were first transformed to take into account the order of presentation. Table 1 shows the preference scores averaged over all words and listeners. We observed that all individual modifications improved quality, with FFXF yielding the least amount of improvement, followed by SBXF and then TDXF. The combined ALL condition led to the highest overall score. Individual t-tests (one-sided) showed significant ($p \leq 0.05$) differences between the following condition pairs: BASE-FFXF ($p = 0.04$), BASE-SBXF ($p = 0.002$), BASE-TDXF ($p < 0.001$), and BASE-ALL ($p < 0.001$). However, TDXF-ALL ($p = 0.31$) did not show significant differences.

Figure 6 illustrates the relationship between quality scores and conditions, when separated by the four stimulus types defined in Section 3.1. We observed that the ordering of conditions remained mostly invariant across all types. However, we noted that the SBXF condition resulted in a relatively low score for stimulus types for which D_{FF} and D_{SB} was small, and that the ALL condition did not improve upon the TDXF condition for two of the four stimulus types.

To further investigate the relationships between distances and scores of various conditions, we performed a linear regression with D_{FF}, D_{SB}, and $D_{FF} + D_{SB}$ as independent variables and scores Q for various conditions as dependent variable. Table 2 shows correlation coefficients for four relationships of interest, for all available data, and for data for which either D_{FF} or D_{SB} was large or small, respectively. For all data, correla-

[1] An atypical concatenation inside the frication of the /dZ/ unit was forced for purposes of illustration.

Correlation Coefficient	All	$D_{FF} \uparrow$	$D_{SB} \uparrow$	$D_{FF} \downarrow$	$D_{SB} \downarrow$
$D_{FF} \rightarrow Q_{\text{FFXF}}$	0.11	0.18	0.14	-0.17	0.06
$D_{SB} \rightarrow Q_{\text{SBXF}}$	0.52⋆	0.48⋆	0.38	0.55⋆	0.46⋆
$D_{FF} + D_{SB} \rightarrow Q_{\text{TDXF}}$	0.50⋆	0.50⋆	0.17	0.54⋆	0.36
$D_{FF} + D_{SB} \rightarrow Q_{\text{ALL}}$	0.52⋆	0.45⋆	0.23	0.60⋆	0.34

Table 2: Correlations between distances and scores, for all data and large (↑) and small (↓) distances. Starred correlations are significant.

tion coefficients were significant at $r = 0.5$, with the exception of predicting Q_{FFXF} from D_{FF}. The latter relationship was not significant for any stimulus types. Predicting Q_{SBXF} from D_{SB} resulted in significantly positive coefficients, except for when D_{SB} was large. Predicting Q_{TDXF} and Q_{ALL} from $D_{FF} + D_{SB}$ resulted in significantly positive coefficients for all data, and for data with large or small D_{FF}; however, when using data with large or small D_{SB}, coefficients were smaller, and not significant.

4. Conclusion

We proposed two approaches that increase the degree of spectral control in concatenative speech synthesizers, by controlling formant frequencies and energies in four spectral bands. We used the proposed methods (FFXF and SBXF) and one additional time-domain cross-fading technique (TDXF) to smoothly connect from one unit of the acoustic inventory to the next. A comparative mean opinion score listening test showed that all three methods significantly improved perceived quality, to varying degrees. Using all three methods in combination (ALL) was not significantly different from using TDXF alone. We speculate that this is so because (1) even though formants are not continuous in frequency, the human auditory system resolves cross-faded formants with small frequency differences into smoothly varying formants, and (2) a global energy smoothing takes place simultaneously. However, TDXF cannot implement other types of spectral changes, such as controlling the degree of articulation or modeling reduction phenomena due to changes in phoneme duration.

Even though we considered the whole phoneme region for cross-fading in this work, the approach could also be used for smaller regions centered around the point of concatenation.

In the future, we plan on exploring additional capabilities. One example is the application of formant parameters predicted by an explicit model that considers input parameters such as phoneme durations and degree of articulation. Another example is the transformation of formants by a mapping function for voice transformation.

5. References

[1] D. Klatt, "Review of text-to-speech conversion for English," *JASA*, vol. 82, no. 3, pp. 737–793, Sept. 1987.

[2] R. H. Manell, "Formant diphone parameter extraction utilising a labelled single-speaker database," in *ICSLP*, Sydney, Australia, 1998.

[3] C. Shadle and R. Damper, "Prospects for articulatory synthesis: A position paper," in *Proc. of the fourth ISCA Tutorial and Research Workshop*, Perthshire, Scotland, 2001.

[4] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal, "The AT&T Next-Gen TTS System," in *Proc. Joint Meeting of ASA, EAA and DEGA*, 1999.

[5] P. Taylor, A. Black, and R. Caley, "The Architecture of the Festival Speech Synthesis System," in *Proc. of the third ESCA workshop on speech synthesis*, Jenolan Caves, Australia, 1998.

[6] Q. Miao, X. Niu, E. Klabbers, and J. van Santen, "Effects of prosodic factors on spectral balance: analysis and synthesis," in *Speech prosody*, Dresden, Germany, 2006.

[7] H. Mizuno, M. Abe, and T. Hirowaka, "Waveform-based speech synthesis approach with a formant frequency modification," in *ICASSP*, 1993, pp. 195–198.

[8] D. T. Chappell and J. H. L. Hansen, "A comparison of spectral smoothing methods for segment concatenation based speech synthesis," *Speech Communication*, vol. 36, no. 3, pp. 343–373, 2002.

[9] P. H. Low, C. H. Ho, and S. Yaseghi, "Using estimated formant tracks for formant smoothing in text to speech synthesis," in *ASRU*, 2003, pp. 688–693.

[10] J. Wouters, *Analysis and Synthesis of Degree of Articulation*, Ph.D. thesis, Oregon Graduate Institute, Portland, OR, 2001.

[11] D. Broad and F. Clermont, "A methodology for modeling vowel formant contours in CVC context," *JASA*, vol. 81, no. 1, pp. 155–165, Jan. 1987.

[12] E. Klabbers, J. van Santen, and A. Kain, "The contribution of various sources of spectral mismatch to audible discontinuities in a diphone database," *IEEE Transactions on Audio, Speech, and Language Processing Journal*, vol. 15, no. 3, pp. 949–956, 2006.

[13] Agaath Sluijter, *Phonetic Correlates of Stress and Accent*, Ph.D. thesis, Holland Institute of Generative Linguistics, 1995.

[14] J. Wouters and M. Macon, "Control of spectral dynamics in concatenative speech synthesis," *IEEE Trans. Speech and Audio Proc.*, vol. 9, no. 1, pp. 30–38, Jan. 2001.

[15] Y.-S. Hsiao and D.G. Childers, "A new approach to formant estimation and modification based on pole interaction," in *Thirtieth asilomar conference on signals, systems and computers*, 1996, vol. 1, pp. 783–787.

[16] E. Turajlic, D. Rentzos, S. Vaseghi, and C.-H. Ho, "Evaluation of methods for parameteric formant transformation in voice conversion," in *ICASSP*, 2003, pp. 724–727.

[17] R. W. Morris and M. A. Clements, "Modification of formants in the line spectrum domain," *IEEE Signal Processing Letters*, vol. 9, pp. 19–21, Jan. 2002.

[18] Boris Doval, Christophe d'Allesandor, and Nathalie Henrich, "The voice source as a causal/anticausal linear filter," in *VOQUAL*, Aug. 2003.

[19] T. Toda, A. W. Black, and K. Tokuda, "Spectral Conversion Based on Maximum Likelihood Estimation Considering Global Variance of Converted Parameter," in *ICASSP*, Mar. 2005, vol. 1, pp. 9–12.

Feature Transformation Applied to the Detection of Discontinuities in Concatenated Speech

Barry Kirkpatrick, Darragh O'Brien and Ronán Scaife

Speech Group
Research Institute for Networks and Communications Engineering
Faculty of Engineering and Computing
Dublin City University
Dublin 9, Ireland

{bkirkpatrick, dobrien}@computing.dcu.ie, scaifer@eeng.dcu.ie

Abstract

The quality of concatenated speech depends on the degree of mismatch between successive units. Defining a perceptually salient join cost to represent the degree of mismatch has proven to be a difficult task. Such a join cost is critical in unit selection synthesis to ensure that the optimum sequence of speech units is selected from the units available in the speech inventory. In this study the problem of defining a join cost is extended to include a feature transformation stage. Two feature transformations are considered, principal component analysis and a neural network-based approach. Each transformation was investigated for its ability to improve the detection of discontinuities in concatenated speech for a given feature set. The results indicate that a feature transformation combining principal component analysis as a preprocessing stage to a neural network-based transformation can increase the rate of detection of discontinuities. The neural network was trained using perceptual data obtained from a subjective listening test indicating if a join is continuous or discontinuous. The highest scoring measure based on this strategy provided a correlation with perceptual results of 0.8859 compared with a value of 0.7576 over the baseline MFCC measure on the same test data set.

1. Introduction

Unit selection synthesis is currently considered state-of-the-art in text-to-speech synthesis. Synthetic speech is generated by concatenating units of speech which are selected from a large speech database. Cost functions are employed to select the optimum sequence of units. The quality of speech generated can be quite inconsistent; natural sounding speech is generated when the join between successive speech units is inaudible; much lower quality speech results when the transition between units sounds discontinuous. An audible discontinuity occurs when two units are not appropriately matched, specific criteria for a perceptually continuous join remain undefined to date. Join costs currently employed in unit selection typically consist of f0 and spectral measures usually represented by Mel-frequency cepstral coefficients (MFCC).

1.1. Background

An ideal join cost should accurately reflect human perception of discontinuity. A number of studies have attempted to determine which distance measures are most successful at predicting audible discontinuities in concatenated speech [1–6]. Many of these studies have presented conflicting results, with measures that ranked highly in one study performing poorly in another. It is difficult to make direct comparisons between studies as each used a different database and different criteria to rank each measure. A consistent element in each of the studies is that the degree of correlation with human perception is often quite weak, also many studies report improvement in results with the inclusion of basic perceptual modelling.

The aforementioned studies predominantly focused on a comparison of standard speech parametrisations as measures of spectral continuity typically based on representations found to be useful in automatic speech recognition and coding. Both Bellegarda [7] and Vepa and King [8] have tailored specific strategies for the problem of defining spectral join costs. Bellegarda developed an alternative transform approach based on a singular value decomposition of speech frames extracted about the points of concatenation in the speech inventory. Vepa and King developed a Kalman filter based strategy that measured the degree of mismatch between idealised trajectories predicted by the Kalman filter and the actual trajectories about the point of concatenation.

In this study feature transformations are investigated to enhance the ability of existing spectral measures to detect discontinuities, specifically principal component analysis (PCA) and neural networks. This extends the existing distance measure framework to a feature space based framework and enables the application of feature space transformations. The objective is to maximally exploit the discriminating information in the features extracted with the proposed transformations and as a result determine a spectral join cost that correlates better with human perception of discontinuity.

1.2. Motivation

In unit selection systems the spectral join cost is computed by extracting spectral features from speech frames adjacent to the unit boundaries and calculating the Euclidean distance between the features. In this computation the level of spectral mismatch between corresponding features is treated equally for all features. Perceptually it is unlikely that all features are equally significant. Mismatch below a certain threshold is likely to be perceptually irrelevant and should be discarded with no contribution to the overall distance measure. Certain spectral bands may be of more significance, for example mismatch coinciding with the location of a formant would be expected to be of more perceptual importance than mismatch in other regions of the spectrum. It has been reported that an abrupt increase in an acoustic component is more perceptually significant than a sudden drop in amplitude [9], this indicates that mismatch due to the introduction of a new component should be weighted more heavily than mismatch due to a drop in component energy. With the application of neural networks a mapping can be learned from data provided from subjective listening tests relating continuous and discontinuous joins with input feature vectors representing a join. The appropriate weighting of the features is data driven and does not require advanced knowledge of auditory processing.

The testing procedure to quantify the performance of the proposed techniques is outlined in section 2. Section 3 introduces the background associated with generalising the distance measure approach to a feature space representation and the application of feature transformations. The application of PCA and neural networks as feature transformations are also discussed in section 3. Section 4 contains the results from employing PCA and neural networks to transform features for the task of detecting discontinuities in concatenated speech. Section 5 contains discussion and conclusions.

2. Testing

To test each proposed technique it is necessary to correlate the perceptual response of a human listener with each candidate measure. This enables a comparison of the standard spectral distance for a given feature set with the proposed measure after the feature transformation has been applied.The evaluation of each measure was conducted using the database from [5] and the corresponding perceptual results. The perceptual stimuli consisted of 1800 monosyllabic words. Each of these words was generated by concatenating two half words with the same vowel nucleus. The inventory of units consisted of 300 words recorded from an adult male. The inventory of 300 words consisted of 50 sets of 6 words. Within each set the words share the same vowel nucleus and differ in the final or initial consonant. The perceptual test required the listeners to make a forced decision for each test word: continuous or discontinuous. Twelve listeners in total contributed perceptual results with coverage of three listeners per subtest. A majority scoring system was employed to indicate if a test word was continuous or discontinuous.

In order to test the performance of the neural network-based measures the database was split into training and testing subsets. The training set contained 50% of the database and the remaining 50% made up the testing set. The database was divided to have an equal number of discontinuities in both the training and testing sets. The database contained a total of 434 discontinuities. The database was split such that joins contained in each vowel type represented in the database are equally spread between training and testing subsets.

3. Feature transformations

Many studies have been conducted in the automatic speech recognition literature investigating the use of feature transformations to improve the discriminating qualities of the features for speech recognition [10,11]. In this study PCA [12] and neural networks [13] are applied to transform features representing the spectral join cost in concatenated speech. The objective is to investigate the ability of these techniques to enhance existing measures for objective detection of discontinuities.

3.1. Defining a join vector

In order to apply feature transformations that fully exploit the discriminating information within each feature, it is necessary to define a suitable vector to represent a join. Existing methods employ a distance to represent a join and feature vectors to represent individual units of speech. In this study the error vector is used to represent a join, hereby referred to as the join vector, which is computed by subtracting the left and right unit feature vectors, $\mathbf{x}_{left}$ and $\mathbf{x}_{right}$. Each feature in the join vector represents the degree of mismatch between the corresponding features in the left and right units.

$$\mathbf{x}_{join} = \mathbf{x}_{left} - \mathbf{x}_{right} \tag{1}$$

Different strategies to construct join vectors motivated by the standard, l_p norms and the symmetric Kullback-Leibler were investigated in [14]. The join vector resulting from the subtraction of the left and right features was found to be suitable. This generalises the standard distance measure approach for the l_p norms. Classification of the join vectors in the feature space without further processing for the join vectors constructed using equation (1) corresponds exactly with calculating the l_p distance between the original left and right feature vectors, for a given p, equation (2).

$$l_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{N} |x(i) - y(i)|^p \right)^{1/p} \tag{2}$$

The ideal join should correspond with the origin in the feature space and the quality of the join can be quantified as the distance of the join vector from the origin. Thus joins can be classified as continuous or discontinuous with respect to distance from the origin. When classification is based on a distance from the origin, the subsequent choice of norm for the feature space establishes the geometry of the classifier. The classifiers corresponding to the l_1, l_2, l_4 and l_∞ norms are illustrated in Fig. 1. This illustrates how the standard distance measure can be interpreted in the feature space.

With the join vector representation it is possible to apply a transformation, $\mathbf{A}$, on the join vector before computing the final measure of mismatch, equation (3).

$$\mathbf{X} = \mathbf{A}(\mathbf{x}_{join}) \tag{3}$$

With this approach standard techniques can be applied to increase the separability between join vectors representing continuous and discontinuous joins. The application of a linear feature transformation is equivalent to stretching or contracting the individual axes and rotating the classifiers in Fig. 1, with a possible reduction in dimensionality. The final measure of mismatch, D, can be computed from the transformed vector, $\mathbf{X}$.

$$\mathbf{D} = ||\mathbf{X}|| \tag{4}$$

Two techniques were investigated; PCA and a neural network-based approach. For the neural network-based approach PCA was used as a preprocessing stage for dimensionality reduction of the input data. Ideally a feature transformation will remove redundant information and weight perceptually important information resulting in improved discrimination between continuous and discontinuous joins.

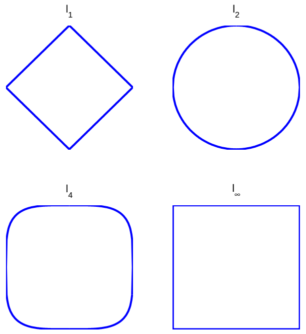


Fig. 1: Classifier shape in 2 dimensions employing l_1, l_2, l_4 and l_∞ norms.

3.2. Principal component analysis

Principal component analysis is an unsupervised learning technique and does not require splitting the database. In this study PCA is investigated for two roles; firstly for its ability as an unsupervised learning technique to improve the detection of discontinuities and secondly as a dimensionality reduction technique to remove redundant information preceding the application of neural networks. The removal of redundant information with PCA often leads to an improvement in performance in many pattern recognition tasks [12].

In the implementation of PCA the data is centred in the feature space about the origin by subtracting the mean vector computed over the complete database. The data is also normalised with respect to variance such that the standard deviations are equal to one. The normalised data is transformed using PCA, this produces transformed join vectors whose components are uncorrelated and ordered according to the magnitude of their variance.

3.3. Neural networks

For each of the feature sets considered and for each possible combination of feature sets a corresponding neural network is trained from the training set of the database. PCA is applied as a preprocessing step to reduce the dimensionality of the input vectors before training the networks. When the join vector is passed through the neural network a distance measure is output. To train the neural networks join vectors corresponding with discontinuities are assigned an output value of 1 and continuous joins are assigned an output value of 0.

A number of neural network architectures were investigated for the task of detecting discontinuities. General regression neural networks (GRNN) [15] were found to be the most suitable for the task. Feedforward neural networks were investigated but were found to be less consistent than GRNNs. GRNNs do not suffer from the problem of getting trapped in local minima, which can be a problem with iteratively trained neural networks.

4. Results

The results presented were computed by generating receiver operating characteristic (ROC) curves [16] that relate the perceptual results of human listeners with the proposed measures. Two probability density functions, $p(\tau|1)$ and $p(\tau|0)$, are estimated for each distance measure, τ, based on the perceptual results for continuous (0) and discontinuous (1) joins. The ROC curves were calculated from the probability density functions and provide information regarding the separability of $p(\tau|1)$ and $p(\tau|0)$, for each distance measure.The ROC curves are generated by plotting the hit rate, P_H, against the false alarm rate, P_{FA}.

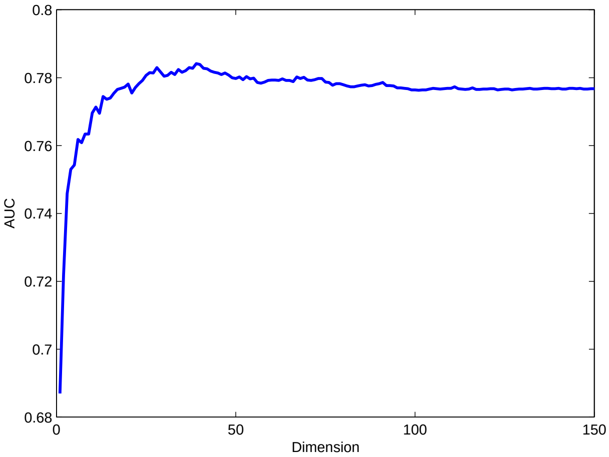


Fig. 2: Plot of AUC value as the output dimension from PCA is varied using Log PS.

$$P_H(\tau_0) = \int_{\tau_0}^{\infty} p(\tau|1) d\tau \quad (5)$$

$$P_{FA}(\tau_0) = \int_{\tau_0}^{\infty} p(\tau|0) d\tau \quad (6)$$

The performance metric employed was the area under the ROC curve (AUC). The AUC represents the separability of the sets of continuous and discontinuous joins for each measure tested. The AUC values are presented for before and after the application of the proposed transforms for each feature set tested.

4.1. Features

The features employed were the log power spectra (Log PS) computed from the fast Fourier transform (FFT), MFCCs and Line spectral frequencies (LSF). They were all extracted using a frame of one pitch period in length with a Hanning window. The MFCCs were computed from FFT spectra and the LSFs were computed from a 16th order LPC analysis on a Mel scale .

4.2. PCA

The results comparing the AUC values computed before and after the application of PCA are presented in Table 1. These results were computed across the entire dataset as the database did not require the separation into training and testing for the application of PCA.

Features	**x**	$PCA[\mathbf{x}]$
MFCC	0.75	0.7696
LSF	0.7381	0.6966
Log PS	0.7615	0.7841

Table 1: Comparison of results with and without the application of PCA for each feature set; the table entries indicate the AUC value.

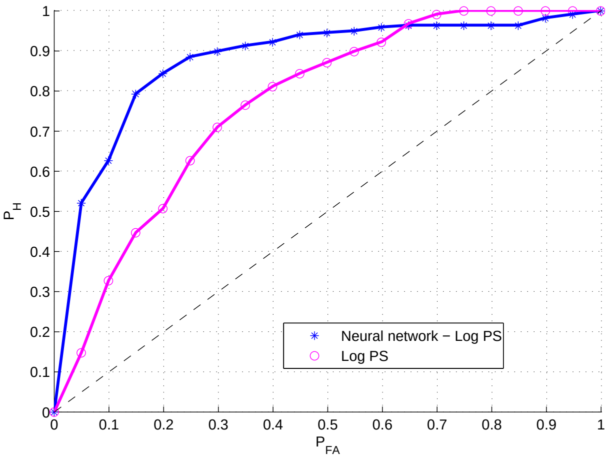


Fig. 3: Illustrating the ROC curves computed from the Log PS before and after applying the neural network.

Features	x	$ANN[\mathbf{x}]$
Log PS	0.7673	0.8744
MFCC	0.7565	0.8413
LSF	0.7468	0.7955

Table 2: Comparison of results before and after the application of the neural network for each feature set; the table entries indicate the AUC value.

For both MFCCs and Log PS the application of PCA was found to improve the rate of detection of discontinuities. For LSFs, PCA was found to result in a decrease in the AUC value. The dimension of the transformed vector was chosen to maximise the AUC value for each of the feature sets. Figure 2 illustrates the resulting AUC values as the dimension of the transformed vector is varied for the case of join vectors constructed from Log PS. The maximum AUC value in Figure 2 occurs for a dimension of 39. This indicates how effective PCA is at retaining the discriminating information in relatively few dimensions; the original dimension was 256. This justifies the use of PCA as a preprocessing stage prior to applying neural networks. For MFCCs the maximum AUC value was obtained at a dimension of 3; the original dimension was 19. For LSFs the maximum AUC value corresponded with the maximum possible dimension of 16, although the AUC value essentially plateaued at a dimension of 4 (AUC = 0.6953 for dimension 4).

4.3. Neural networks

The results for each of the feature sets before and after the application of the proposed neural network-based transformation are presented in Table 2. The neural network is trained on the training set and tested in a separate testing set. PCA is employed as a preprocessing stage to reduce the dimensionality of the input vectors. The results presented are for GRNN type networks.

Table 2 indicates that the neural network-based approach significantly enhances the performance, this is most notable for Log PS in which the AUC value increased from 0.7673 with the standard distance measure approach to a value of 0.8744 with the proposed approach. The ROC curves comparing each of these measures before and after they were passed through their respective neural networks are illustrated in Figures 3, 4 and 5.

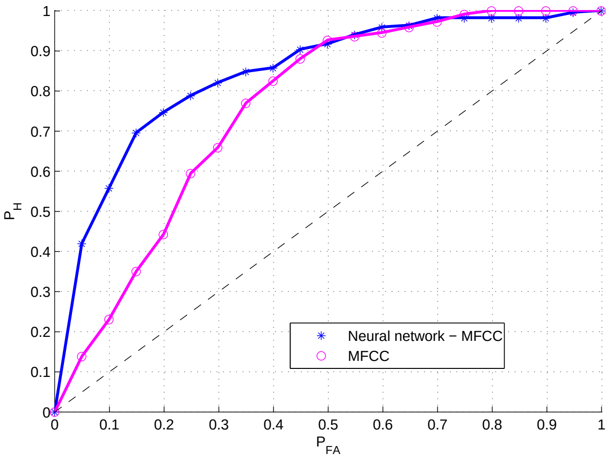


Fig. 4: Illustrating the ROC curves computed from MFCCs before and after applying the neural network.

4.4. Combined measures

To combine the measures each join vector is concatenated and subsequently PCA is applied, at this point the neural network is trained. The results computed from the standard distance measures with no transformation based on the concatenated join vectors and those computed from the neural network-based measures are presented in Table 3.

The neural network-based measure based on both MFCC and Log PS features is the best performing measure tested with an AUC value of 0.8859. For each of the combined measures the neural network-based measure outperforms the corresponding standard measure. The ROC curves comparing the performance of MFCCs combined with Log PS features before and after they were passed through the neural network are illustrated in Figure 6.

Features	x	$ANN[\mathbf{x}]$
MFCC + LSF	0.7468	0.8581
MFCC + Log PS	0.7673	0.8859
LSF + Log PS	0.7517	0.8753
LSF + MFCC + Log PS	0.7517	0.8829

Table 3: Comparison of results with and without the application of the ANN for possible combination of feature sets; the table entries indicate the AUC value.

5. Discussion and conclusions

This paper discusses a framework for applying feature transformations to spectral features for join cost optimisation in concatenative speech synthesis. PCA and a neural network-based strategy were investigated. The results indicate that PCA can be employed as an effective mechanism for dimensionality reduction without losing critical information in the detection of discontinuities. PCA does not always provide an increase in the performance as illustrated in the results for the LSF-based measure. The potential gain in performance is relatively small when it does occur. Perceptual data is required to optimally select the output dimension. The neural network-based measures were found to outperform the corresponding standard distance

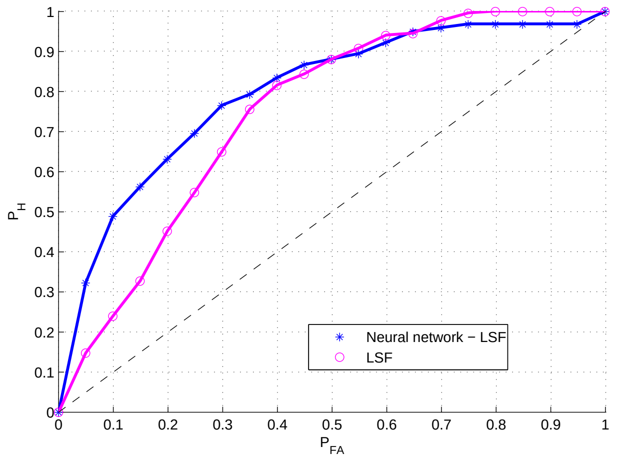


Fig. 5: Illustrating the ROC curves computed from LSFs before and after applying the neural network.

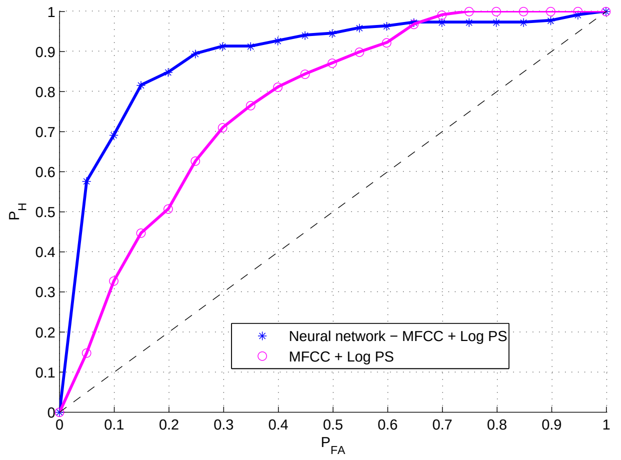


Fig. 6: Illustrating the ROC curves computed from the combined features from the Log PS and MFCCs before and after applying the neural network.

measure approach for each feature set tested and can be employed to enhance an existing feature set for its ability to detect discontinuities. The neural network-based strategy provided the best results of all measures tested and produced the highest detection rates on the test database to date. This suggests that the proposed feature transformation framework used in conjunction with neural networks is an effective strategy to learn the levels of mismatch that give rise to discontinuities for a given feature set. A critical issue with the proposed strategy is that training the neural network requires perceptual data which requires conducting perceptual experiments. This is a laborious and difficult task; most studies that involved listening experiments for the detection of discontinuities reported that the listeners found the task difficult.

6. Acknowledgements

This work is funded by Science Foundation Ireland, grant number 04/BRG/E0111.

7. References

[1] J. Wouters and M. Macon, “Perceptual evaluation of distance measures for concatenative speech synthesis,” in *Proc. ICSLP*, vol. 6, Sydney, Australia, 1998.

[2] E. Klabbers and R. Veldhuis, “Reducing audible spectral discontinuities,” *IEEE Trans. on Speech and Audio Processing*, vol. 9, pp. 39 – 51, 2001.

[3] Y. Stylianou and A. K. Syrdal, “Perceptual and objective detection of discontinuities in concatenative speech synthesis,” in *Proc. ICASSP*, Salt Lake City, USA, 2001.

[4] J. Vepa and S. King, “Subjective evaluation of join cost and smoothing methods for unit selection speech synthesis,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, pp. 1763 – 1771, 2006.

[5] B. Kirkpatrick, D. O’Brien, and R. Scaife, “Feature extraction for spectral continuity measures in concatenative speech synthesis,” in *Proc. ICSLP*, Pittsburgh, USA, 2006.

[6] E. Klabbers, J. P. H. van Santen, and A. Kain, “The contribution of various sources of spectral mismatch to audible discontinuities in a diphone database,” *IEEE Transactions on audio speech and language processing*, vol. 15, pp. 949 – 956, March 2007.

[7] J. Bellegarda, “A global, boundary-centric framework for unit selection text-to-speech synthesis,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, pp. 990 – 997, 2006.

[8] J. Vepa and S. King, “Kalman-filter based join cost for unit-selection speech synthesis,” in *Proc. EUROSPEECH*, Geneva, Switzerland, 2003.

[9] Q. Summerfield, A. Sidwell, and T. Nelson, “Auditory enhancement of changes in spectral amplitude,” *J. Acoust. Soc. Am.*, vol. 81, pp. 700 – 708, 1986.

[10] P. Somervuo, B. Chen, and Q. Zhu, “Feature transformations and combinations for improving ASR performance,” in *Proc. EUROSPEECH*, Geneva, Switzerland, 2003.

[11] P. Somervuo, “Experiments with linear and nonlinear feature transformtions in hmm based phone recognition,” in *Proc. ICASSP*, Hong Kong, 2003.

[12] I. Jolliffe, *Principal Component Analysis*. Springer-Verlag, 1986.

[13] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford, 1995.

[14] B. Kirkpatrick, D. O’Brien, and R. Scaife, “A comparison of spectral continuity measures as a join cost in concatenative speech synthesis,” in *Proc. of the IET Irish Signals and Systems Conference (ISSC)*, Dublin, Ireland, 2006.

[15] D. F. Specht, “A general regression neural network,” *IEEE Trans. on Neural Networks*, vol. 2, 1991.

[16] R. Duda and R. E. Hart, *Pattern Classification*, 2nd ed. John Wiley and Sons, 2001.

Towards Conversational Speech Synthesis;
Lessons Learned from the Expressive Speech Processing Project

Nick Campbell

NiCT/ATR-SLC
National Institute of Information and Comunications Technology
& ATR Spoken Language Communication Research Labs
Keihanna Science City, Kyoto 619-0288, Japan
nick@nict.go.jp, nick@atr.jp

Abstract

This paper discusses some ideas for the requirements and methods of conversational speech synthesis, based on experience gained from the collection and analysis of a very large corpus of conversational speech in a variety of real-life everyday contexts. It shows that because variation in voice quality plays a significant part in the transmission of interpersonal and affect-related social information, this feature should be given priority in future speech synthesis research. Several solutions to this problem are proposed.

Keywords: non-verbal speech, expression of affect, concatenative synthesis, conversational speech corpus, syntax of spoken language

1. Introduction

The JST/ATR Expressive Speech Processing project took place over a period of five years from July 2000 to March 2005 [1]. In that time, 1,500 hours of natural unprompted conversational speech was recorded in a variety of everyday situations using the voices of up to forty paid volunteers as they went about their normal daily activity. Recordings were made directly to DAT or MD using high-quality head-mounted close-talking microphones and all the speech was transcribed manually to form the JST/ATR ESP corpus [2]. A subset of the corpus was further manually labelled to annotate speaking-style and affect-related features.

Given such a large pool of speech samples, including more than 600 hours of speech from one adult female volunteer, a concatenative speech synthesiser was built and tested. The assumption being that given five-years of one person's daily conversations, the system should already contain and so be able to accurately generate most of the speech needed for the sixth or future years from that supply. This turned out not to be the case, but from this work it was discovered that a large amount of the speech was used for expressing interpersonal relationships and affective information [3], rather than propositional content, and considerable effort has since been put into producing a dictionary and a grammar of that mode of nonverbal speech.

2. The Function of Conversational Speech

If speech synthesis is to be developed for conversational applications, such as virtual agents [4], speech translation [5], or 'customer-care' types of two-way spoken interactions, then it will perhaps need to cover the full range of vocal activity encountered in human conversational speech. In other words, it will need to be able to express 'personal feelings' as well as to transmit linguistic information. This will require a degree of prosodic control for which we are currently not well prepared.

Many speech synthesis applications assume a 'broadcast' mode of speech, where the synthesiser speaks and a human listens, with little interaction between the two sides. The focus in broadcast speech is on correctly rendering an input text so that its prosody expresses the syntactic and semantic relations of the component words and their linguistic organisation [6]. Its function is to transmit linguistic information. Contrast this with a conversational mode [7], where the synthesiser also has to take on the role of a listener, providing feedback sounds to signal comprehension (i.e., adequate processing by the dialogue system of the recognised input speech stream), agreement, sympathy, interest, alarm, etc., and their opposites. In this latter 'paralinguistic' mode of speech, the verbal content is limited but its prosodic impact is great. In real interactive speech, laughs and other affect-bursts are common, and 'grunts' take the place of more formal semantics. Phatic communion [8, 9] is as common as (or even more so than) the transfer of linguistic information.

2.1. Non-Verbal Speech Sounds

From the analysis of the ESP corpus, it was learned that approximately half of the conversational-speech utterances are difficult to comprehend from their transcriptions alone. That is, a knowledge of their prosody and voice quality; i.e., of *how* they were spoken, is necessary before an interpretation of their meaning and the speaker's intention can be formed.

A dictionary listing the 100 most frequent utterances in this conversational speech corpus [10] contains words such as "yeah", "okay", "maybe", "gotcha", "uhuh" (i.e., their Japanese equivalents), as well as many laughs, intakes of breath, grunts (such as "ummm", "hmmm", "ooh", etc) and greetings. This list alone is sufficient to cover half of the speech data in terms of utterance frequency. Such non-verbal speech sounds are extremely common. Expanding the list to include similar but less frequent utterances gives at least 2,000 entries excluding laughs, which if we include verbatim (where haha is distinguished from hahaha and hahahaha) involves a further 2,000 types or more.

2.2. Synthesis of Non-Verbal Speech Sounds

Several methods have been tested for the synthesis of such sounds using speech data from the ESP corpus. Being concatenative, they entails little signal processing or prosody or voice-quality manipulation, and simply require the construction of an efficient index to retrieve suitable speech samples from the corpus for replay intact. Although no phone-level concatenation is required for these short complete and self-contained utterances, this method arguably still falls under the umbrella of 'speech synthesis' as it entails the generation of interactive speech utterances by use of a computer system.

The selection of a phatic speech utterance obviously cannot be done just by text alone, as the style and nuance of such speech sounds is much more variable (and informative) than their textual representation. Even something as literal as a greeting, e.g., "Good Morning!", becomes a delicate indicator of speaker-state and speaker-listener relationships through subtle differences in prosody and tone-of-voice, when its phatic role is considered.

Hand-in-hand with the task of selecting appropriately expressive waveform segments is the problem of input; since a computer keyboard (which is limited to the generation of plain text as input) may not be the most appropriate device. Assuming for example that many of our users might prefer to use a portable telephone keypad as their input device of choice, we tested an icon-based menu interface, 'NATR', whereby common conversational utterances could be chosen by toggling the selector button up, down, left, or right (using the thumb) and then pressing a function key to send/synthesise the target speech (see Figure 1). An extended version, 'Chakai', for use with notebook computers with space for occasional free input (shown in Figure 2) has been described elsewhere [11].

Common to both these input devices is a matrix of valency and activation for selection of an appropriate utterance, with icons depicting characteristic features of the utterance in a non-text-based manner. This is because a fundamental assumption of this form of unit selection for conversational speech segments is that the target speech sound is constrained by a set of discourse and interactional features that determine not only its resulting prosody and voice quality, but also the text of the utterance itself. The greeting above is only one form such an utterance might take; when spoken to a close friend it might instead be realised as "Hi!", or clipped down to "mornin' " if the speaker is not feeling too bright. The selection of a phatic utterance should therefore result in a complete and appropriate discourse event, rather than being thought of as determining the prosody and speaking style for a predetermined lexical sequence. This gives the selection procedure a greater freedom to produce what is most common in the corpus, provided that the labels can effectively constrain selection by representing the factors that generate such an event in the real world.

3. Characteristics of Non-Verbal Speech

In previous work [13] it was proposed that the structure of conversational speech can best be explained as an intermingled sequence of 'wrappers' and 'fillers' such that linguistic content is chunked into small segments that are 'wrapped' by the common and frequently repeated non-verbal speech segments so that both the propositional content and the intended interpretation of the linguistic sequence can be simultaneously conveyed through speech, allowing even a listener unfamiliar with the speech habits of the speaker to be able to interpret the subtle affective changes expressed through micro prosodic and voice-quality variations.

In this paper, we focus more on the lexical, syntactic, acoustic and prosodic characteristics of these 'wrappers' in an attempt to explain how they function and how they might be used to produce natural-sounding utterance sequences for conversational interaction between a person and an agent or agents using speech synthesis.

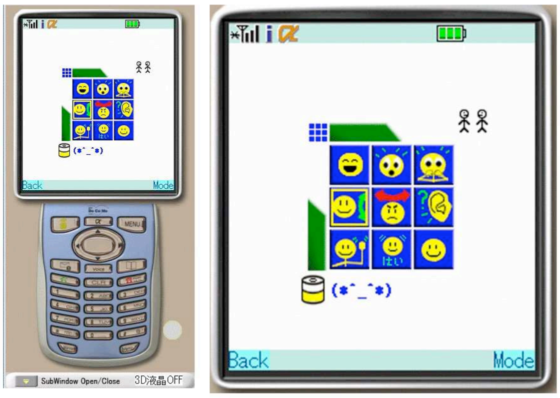


Figure 1: *Input device using portable telephone.*

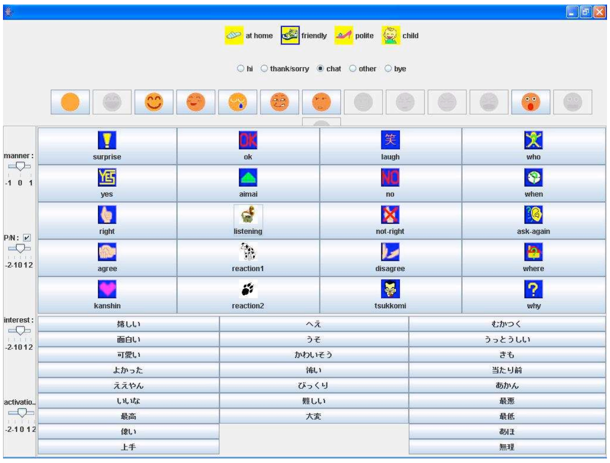


Figure 2: *Input device using notebook computer. Note the space in the centre of the bottom two rows for the input of free text.*

3.1. Wrappers and Fillers - Interaction Devices

Erm, this might seem obvious, but, err, we don't usually use 'wrappers' in text, do we? The previous sentence could better be expressed by seven words (those from 'we' to 'text' inclusive, i.e., what we are calling here the 'fillers'), but nine were added to make it more conversational in style. Contrary to the theory of least effort, it seems that people produce much more speech than 'necessary' (sic) to communicate their intentions. This has been discussed in linguistic science under the competence-performance framework [12], and even today many non-verbal speech sounds are considered to be 'noise'; removed from a recording, not transcribed, covered by a 'garbage-model' in speech recognition, or similarly downgraded and ignored. Errm, one does NOT start a sentence with 'errm'!

And yet these sounds perform a very useful function in discourse. Hesitation is a way of indicating politeness for example, and starting an utterance with 'errm' (or its equivalent) to indicate hesitation is therefore a form of politeness in speech. Stating the obvious, similarly, should not be necessary. How-

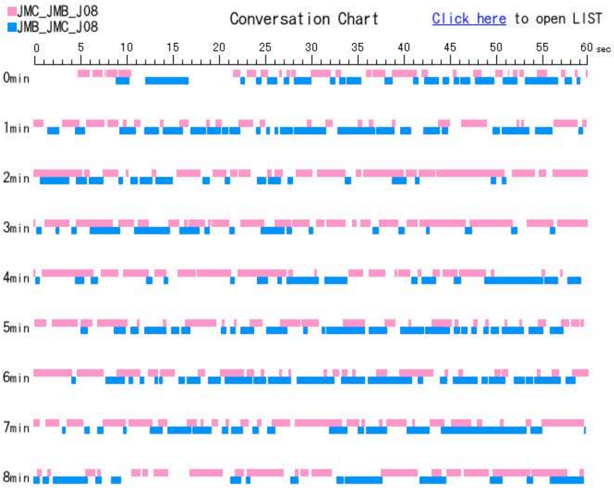


Figure 3: *Speech & silence plots for the first nine minutes of a conversation between two male speakers, JMC and JMB, showing fragmentation of the discourse and progressive but not absolute alternations of speaker dominance. Each line shows one minute of speech, with speaker JMC's speech activity plotted above and that of speaker JMB plotted below. White space indicates lack of speech activity*

Table 1: Results of a principal component analysis of the speech features. We see a decrease in the standard deviation (sd) of the rotated variables as the component number increases, and a decrease in the proportion of the variance (pov) that each component accounts for. By PC7 we note that 82.6% of the cumulative proportion of the acoustic variance (cp) in these data can be accounted for.

```
Importance of components:
       PC1   PC2   PC3   PC4   PC5   PC6   PC7
sd    1.86  1.62  1.35  1.18  1.05  1.00  0.95
pov   0.23  0.17  0.12  0.09  0.07  0.06  0.06
cp    0.23  0.40  0.53  0.62  0.69  0.76  0.82
       PC8   PC9  PC10  PC11  PC12  PC13  PC14
sd    0.87  0.74  0.64  0.59  0.51  0.39  0.31
pov   0.05  0.03  0.02  0.02  0.01  0.01  0.00
cp    0.87  0.91  0.94  0.96  0.98  0.99  1.00

Rotation:
        PC1   PC2   PC3   PC4   PC5   PC6   PC7
fmean -0.35  0.23  0.31 -0.13  0.06 -0.11  0.01
fmax  -0.33  0.15  0.36 -0.11  0.08 -0.14  0.02
fmin  -0.02  0.13 -0.10 -0.52 -0.52 -0.15 -0.11
fpct  -0.20  0.04  0.05 -0.10  0.38 -0.43 -0.57
fvcd   0.19  0.27  0.05  0.55  0.11  0.02 -0.19
pmean  0.03  0.54  0.09  0.11  0.00  0.26  0.01
pmax  -0.24  0.34  0.28 -0.07 -0.11  0.31  0.04
pmin   0.17  0.44 -0.21 -0.12 -0.04  0.09  0.12
ppct   0.05  0.03 -0.09 -0.34  0.67  0.02  0.49
h1h2   0.22 -0.06  0.43  0.15 -0.19 -0.41  0.27
h1a3   0.43 -0.01  0.35 -0.21  0.03  0.01 -0.04
h1     0.42  0.10  0.30 -0.08  0.02 -0.21  0.07
a3    -0.16  0.25 -0.22  0.33 -0.03 -0.46  0.26
dn    -0.11 -0.26  0.37  0.13  0.07  0.37 -0.10
```

ever, "this might seem obvious, but ..." hedges the utterance; it is not redundant but a part of the discourse where the speaker can express affective information, relating to the listener, and to his or her confidence and purpose in speaking. In much the same way "do we?" functions to bring the listener closer into the discourse and to personalise it. It is not a question but a phatic tag.

Furthermore, the very frequency of such tags as "do we?" (here we are including them in the more general term 'wrappers') allows precise variation in expressivity to carry considerable weight of information in the discourse, enabling the speaker to express the degree of belief which the statement is intended to carry. In other words, the linguistic content (i.e., the filling of the utterance) is wrapped in paralinguistic segments that serve to lighten it and to add speaker involvement. This form of speech is limited to conversational styles, and is not found in broadcast modes, where the voice is used solely to portray the content of the text rather than the feelings or attitudes of the speaker.

Such non-verbal (or fringely verbal) use of speech is also particularly common when listening. Active listening demands that the listener chip in frequently to confirm attention, understanding, agreement, etc., and if these phatic sounds are not produced as expected, then most people will simply stop talking. They 'dry-up', asking if the 'listener' is alright perhaps, and the discourse fails as an interactive two-way event.

Figure 3 shows a plot of such two-way activity during a telephone conversation between two people who do not know each other very well. It is probably clear at any given moment in the time sequence who is the dominant speaker, but there is considerable overlap as the listener verbally nods to the speech. Here the same "um" (which is by far the most common utterance in the corpus) can mean yes, no, maybe, just 'I'm listening", 'go on', etc., from differences in intonation, timing, loudness, and voice quality. These are the new challenges for the synthesis of conversational speech. It is not easy to specify the intended variant from text input alone.

3.2. Acoustic features of Wrappers and Fillers

This paper uses the term 'non-verbal' for these speech sounds, but rather than strictly limiting the term to laughs and grunts alone it should be interpreted in its wider meaning to include phrases used more as discourse-gesture than as linguistic content. The example above gave "this might seem obvious" and "do we?" as examples of speech segments that might look like linguistic content but which are actually used more for phatic rather than propositional information transfer. They wrap the linguistic content and give conversational speech its characteristic 'broken' or so-called 'ill-formed' structure illustrated in Figure 3.

Since these non-verbal wrappers function more to carry prosodic and voice-quality information, it is necessary to categorise them primarily by their acoustic features for unit selection in concatenative conversational speech synthesis. Whereas the prosody of a sentence for broadcast-mode speech synthesis can be largely determined from an analysis of its syntactic, semantic and lexical components and their interactions, the prosody of a phatic grunt for conversational speech synthesis has to be determined independently of (and arguably even before) its lexical composition.

To facilitate the use of acoustic features in unit selection, we used a short program written in Tcl/Tk-Snack [14] to extract the main acoustic and prosodic characteristics of each non-verbal utterance in the corpus to represent its speech waveform as a vector of 14 values (see details in [15]). These include five values (fmean, fmax, fmin, fpct, and fvcd) to represent the pitch contour (fundamental frequency of the speech waveform), four (pmean, pmax, pmin, and ppct) for signal amplitude (power), one for duration, and four to represent spectral characteristics (h1h2, h1a3, h1, a3) of the entire utterance.

The fourteen acoustic and prosodic features thus extracted were then subjected to a principal component analysis to reduce the complexity of the data and to determine the strength of any interactions between the factors. For this, the "prncomp" function in "R" [16] was employed *(pc=prcomp(feats, retx=T, center=T, scale.=T)* which yielded results as shown in Table 1.

3.3. Voice quality and Acoustics

While we see from Table 1 that the principal component analysis allows us to reduce our search space to a smaller number of dimensions, we also note that spectral features rank very highly in explaining the acoustic variation. The data shown in Table 1 were all from the single utterance "umm", the most common word in the corpus, so there is no inherent phonetic variation to be expected that might account for the spectral differences. Instead, the difference in voice quality or breathiness in the speech were used to differentiate between different interpretations of the utterance in the discourse.

"Umm" is used in Japanese, as in English, to mean 'yes', 'I'm listening', 'I understand', 'I agree', 'I don't agree', 'I don't understand', I'm surprised', and so on ..., with each intended meaning unambiguous to the listener but indistinguishable from the text alone. The table shows that more than 50% of this acoustic variance can be accounted for by the first three principal components alone, and that more than 80% can be explained by the first seven. This greatly facilitates search for an appropriate unit.

The table also shows that the first principal component is dominated by h1a3 (i.e., the difference between energy measured at the first harmonic and that measured at the third formant = 0.43), h1 (energy at the first harmonic = 0.42), fmean (mean fundamental frequency of the utterance = 0.35), and fmax (maximum fundamental frequency = 0.32). Power dominates the second principal component, and duration (or speaking rate) the third. Whereas there have been some interesting proposals for modification of spectral tilt in the speech signal (and hence breathiness and 'force' in the speech; see e.g., [17, 18]) the interactions between these four components of prosody is so great that the present author maintains their modification results in unacceptable degradation to the perception of naturalness in the resulting speech and loss of this important voice-quality dimension that is so important for signalling affect and social relationships.

4. KeyTalk

In order to explore the problem of synthesising with a very large number of utterances having a limited number of textual representations but considerable variety in their prosodic expression, and hence in their meaning, we tested a system using a midi-keyboard devise as input for unit selection (see Figure 4).

This system, 'KeyTalk', addresses the problem of grouping related utterances and also of selecting among them by use of a 'force' feature to represent prosodic strength of the utterance. Being coded in the midi language, it allows sequences to be recorded and replayed at a later time or modelled statistically for further synthesis development.

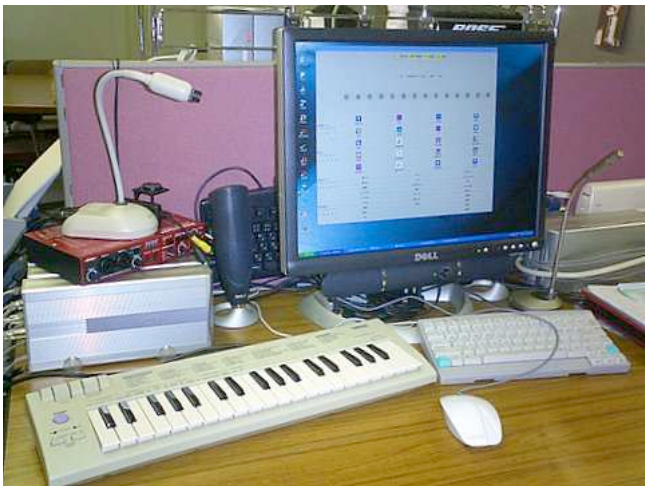


Figure 4: *The KeyTalk setup alongside the NATR conversational speech synthesis interface.*

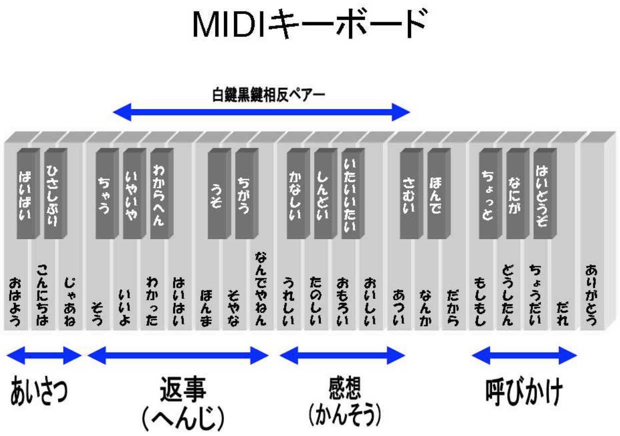


Figure 5: *KeyTalk sample mappings. Groups of keys provide input for related utterances. The keys are touch-sensitive. See text for an explanation.*

4.1. Grouping Related Utterances

Whereas the data for KeyTalk are complex, the software for the synthesiser itself is very simple. The small piano-like keyboard offers a compact view over the full range of several octaves. Each octave section is grouped into sets of seven and five keys, and alternating within each group are the black and the white keys. The keys are touch-sensitive so a strong keypress will produce a different output-value from a weak keypress, with up to 64 intermediate stages of touch sensitivity.

Each group of keys was mapped deterministically to a group of related and frequently-used conversational utterances, as illustrated in Figure 5 in Japanese. The first group of five keys on the left of the figure represent 'greetings', the next twelve map to 'replies', the next seven map to 'opinions', and those on the right to 'initiating' or 'calling' utterances.

By default, the white keys represent the more positive variants, and black keys their negative equivalents, reflecting the major/minor distinction on a musical keyboard. However, it is not always the case that such simple pairings exist; for example the greetings (white keys) map to morning, afternoon, and evening variants respectively, with the black keys for saying farewell.

Clearly, considerable further work would be required on the selection and grouping of the utterances if this were to be implemented as a commercial system for general use, but as a testbed for experimentation the present working prototype allows 'touch-and-feel' hands-on experience for the selection of individual utterances within a real-time interactive framework.

4.2. Touch-sensitive Selection

The mapping from key to utterance is only a token mapping with no guarantee that the exact word mapped to the key (drawn on the key in Figure 5 for illustration) will be the word that is ultimately spoken by the system. Several modifiers come into action to determine the precise prosody and phrasing of the final utterance. These are governed by global, local, and keypress settings.

Every midi keyboard is also equipped with two roller wheels, one (the 'pitch' wheel) with both positive and negative settings sprung to return to the centre position after each use, and the other, for scalar settings, with no spring, retaining its previous value until further changed. For use as a speech synthesiser, these rollers allow the user to modify the affective profile of each utterance to determine the segment for output. As explained in [11], a three-dimensional control space is posited for conversational speech, whereby the content and style of the utterance are determined from (a) the affective state(s) of the speaker, (b) the character the speaker wishes to display to the listener or conversation partner, and (c) the underlying pragmatic and discoursal intentions of the utterance [19]. This constraint-based unit selection is implemented by ranking candidates for each group of utterances in the database.

As all the speech in the corpus has been transcribed, it is a simple matter to select and group all utterances having an occurrence frequency above a predetermined threshold. These are then ranked according to values of the three principal components described above. The settings of the roller-wheels in combination with the force of the keypress determine which utterance segment from the many ranked candidates will be chosen for playback. The group of candidates from which this selection is made is determined by the key being pressed.

4.3. Evaluation

No formal evaluation has been performed on this prototype system, because each utterance synthesised is a complete and self-contained natural-speech segment. There is no concatenation, except at the phrase level, where utterances are separated naturally by pauses, no prosody modification, and no signal processing. By definition each utterance is natural. Judging how informative it is would be a research issue in itself, because there is as yet no formal grammar of non-verbal utterances against which a sequence of such sounds could be measured.

A test of its fun value was carried out in two public demonstrations, one at NAIST (the Nara Institute of Science and Technology) as part of the Open Campus demonstrations in 2006, and again at ATR in the same year as part of the Open House exhibition. In both these cases the keyboard attracted a large number of people and many, especially the younger ones stayed quite a time playing with it, laughing at the sounds that came out, and testing the variety of expression that differences in force of keypress produced.

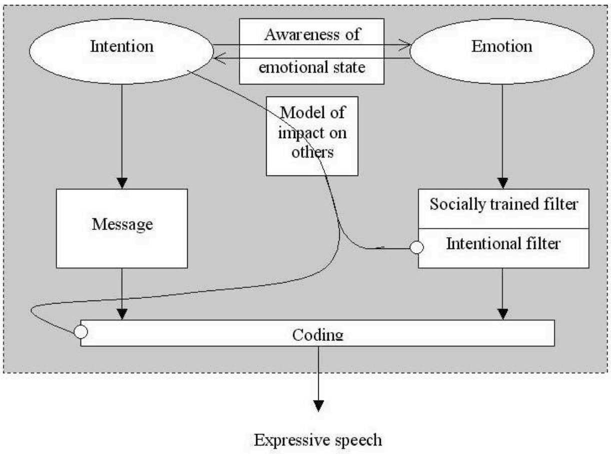


Figure 6: *A model of the constraints (rectangles) and drivers (ovals) underlying the expression of a conversational utterance.*

A different form of evaluation needs to be performed at the level of system design; to find an optimal mapping from keys and clusters of related keys to common utterances in a discourse, and on the mappings between their acoustic characteristics and the perceived intentions of the speaker, but this requires a formal grammar of spoken language that incorporates non-verbal utterances, and so remains as future work.

5. Discussion

For the system to be of use in an automatic speech synthesiser, a control model must be designed for the generation and integration of non-verbal utterances into the speech stream. One such has been proposed in [21] and is illustrated in Figure 6. Here, two elemental forces are considered as jointly having an influence at the most basic level of the desire to speak. These (marked by ovals in the figure) are hidden and not subject to conscious awareness but must be included in the control model as causative factors.

Below them in the figure are a series of filters (marked by rectangular boxes) representing the constraints that determine the coding of an utterance. This coding is at both the lexical and biomechanical level, resulting in the word sequence and its prosody, simultaneously.

The filters or constraints are of three kinds: (a) the message, i.e the intended pragmatic force of the utterance, what is to be conveyed by the speech, but not yet its precise wording, (b) the social impact of the utterance, on the listener, and in the discourse, and (c) the speaker's character and inhibitions, both trained and innate, as well as the facets of that character to be portrayed (revealed or hidden) in the speech through its content and style. The model assumes emotion and intention to be co-drivers of an utterance, but places most of the control at the level of the constraints.

6. Conclusion

There is growing need internationally for the synthesis of expressive speech, not just in speech translation environments, which are now well developed, but also in the growing area of virtual agents, such as Second Life, where animated beings function in a world of their own, interacting both with each other and with the human sponsors of their communities. The business needs for lifelike conversational speech synthesis are great, and very large amounts of real money are already being spent in the virtual communities by a growing number of people across the world.

This paper has described some recent attempts to model the characteristics of conversational speech for use in concatenative speech synthesis, using a very large database of recordings covering a variety of natural environments and interpersonal interactions. Rather than propose a single prototype system, which would be application-specific, it has described several factors that might be taken into consideration in the design of a generic conversational speech synthesis system on the basis of experience gathered from the analysis of the corpus of 1,500 hours of human spoken interactions.

A key theme of the paper is that interactive speech requires different uses of speech prosody from broadcast-mode synthesis, particularly for the expression of affect, interpersonal relationships, and discourse control. Furthermore, in a dialogue system employing conversational speech synthesis, modules will be required for 'active listening' wherein the synthesiser is required to make frequent non-verbal speech sounds to reassure the speaker, to maintain a steady flow of incoming speech, and to control the dialogue. This is an area of discourse which has been little studied, particularly within the engineering and speech technology communities.

The paper has described some of the acoustic features found to be important for the selection of non-verbal speech segments for conversational speech synthesis and has shown that a principal component analysis reduces these to a small manageable number that can be easily ranked and directly used for prosody-based selection of discourse units. The paper has further described some previous attempts at designing novel input devices suitable for use in a conversational environment, both by human users and by computer programs.

7. Acknowledgements

This work is supported by the National Institute of Information and Communications Technology (NiCT), and includes contributions from the Japan Science & Technology Corporation (JST), and the Ministry of Public Management, Home Affairs, Posts and Telecommunications, Japan (SCOPE). The author is especially grateful to the management of Spoken Language Communication Research Labs at ATR for their continuing encouragement and support.

8. References

[1] The Japan Science & Technology Agency *Core Research for Evolutional Science & Technology*, 2000-2005

[2] The JST/CREST Expressive Speech Processing Project homepage can be found at http://feast.atr.jp/

[3] Campbell, N., "Getting to the heart of the matter; speech as expression of affect rather than just text or language", pp 109-118, Language Resources & Evaluation Vol 39, No 1, Springer, 2005.

[4] Second Life: a 3-D virtual world entirely built and owned by its residents. Since opening to the public in 2003, it has grown explosively and at the time of writing is inhabited by a total of 5,788,106 people from around the globe. http://secondlife.com/

[5] Shimizu, T., Ashikari, Y., Sumita, E., Kashioka, H, Nakamura, S., "Development of client-server speech translation system on a multilingual speech communication platform", pp.213-215 in Proc IWSLT, 2006, Kyoto, Japan. 2006.

[6] Crystal, D., "Prosodic Systems and Intonation in English", Cambridge University Press, 1969.

[7] Allwood, J. "An activity based approach to pragmatics". Technical Report (GPTL) 75, Gothenburg Papers in Theoretical Linguistics, University of Goteborg, 1995.

[8] Malinowski, B. "The problem of meaning in primitive languages", pp. 146-152, Supplement to C. Ogden and I. Richards *The meaning of meaning*. London: Routledge and Kegan Paul. 1923

[9] Jakobson, R., "Linguistics and poetics", pp. 350-77 in Sebeok, T. A.(ed) *Style in language*. Cambridge, MA: MIT Press, 1960

[10] Campbell, N., "How speech encodes affect and discourse information", pp.103-114 in Esposito, A., Bratani ć, M., Keller, E., and Marinaro, M., *Fundamentals of Verbal and Nonverbal Communication and the Biometric Issue*, IOS Press, Amsterdam, 2007.

[11] Campbell, N., "Conversational Speech Synthesis and the Need for Some Laughter", in IEEE Transactions on Audio, Speech, and Language Processing, Vol 14, No.4, 1171-1179, July 2006.

[12] Chomsky, Noam. 1965. *Aspects of the Theory of Syntax.* Cambridge, Mass.: MIT Press.

[13] Campbell, N., "On the Use of NonVerbal Speech Sounds in Human Communication" in Proc ParaLing'07: Paralinguistic speech - between models and data, Saarbrucken, Germany, 2007.

[14] Sjölander, K., "The Snack Sound Toolkit: a Tcl/Tk library and toolkit for speech signal processing", http://www.speech.kth.se/snack/

[15] Campbell, N., and Nakagawa, A., "'Yes, yes, yes', a word with many meanings; the acoustics associated with intention variation", in Proc ACII '07 (Affective Computing and Intelligent Interaction) Lisbon, Portugal, 2007.

[16] R Development Core Team, "R: A Language and Environment for Statistical Computing", R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0, (http://www.R-project.org) 2006.

[17] d'Alessandro, C., & Doval, B. (2003). "Voice quality modification for emotional speech synthesis", pp. 1653-1656. Proc. Eurospeech 2003, Geneva, Switzerland

[18] Kawahara, H., de Cheveigné, A., Banno, H., Takahashi, T. and Irino, T., "Nearly defect-free F0 trajectory extraction for expressive speech modifications based on STRAIGHT". Proc. Interspeech2005, Lisboa, pp.537-540, 2005.

[19] Campbell, N., "On the Structure of Spoken Language" in Proc Speech Prosody, Dresden, Germany, 2006.

[20] Mokhtari, P. and Campbell, N., "Quasi-syllabic and quasi-articulatory-gestural units for concatenative speech synthesis", pp.2337-2340 in Proceedings of the 15th International Congress of Phonetic Sciences (ICPhS'03), Barcelona, Spain,. 2003.

[21] Campbell, N., "Expressive / Affective Speech Synthesis", in *Springer Handbook on Speech Processing and Speech Communication*, Benesty, J , Sondhi, M.M., and Huang, Y. (Eds), in Press, Springer, July 2007.

Communicative Speech Synthesis with XIMERA: a First Step

Shinsuke Sakai[1,2], *Jinfu Ni*[1,2], *Ranniery Maia*[1,2], *Keiichi Tokuda*[1,3], *Minoru Tsuzaki*[1,4]
Tomoki Toda[1,5], *Hisashi Kawai*[2,6], *Satoshi Nakamura*[1,2]

[1]National Institute of Information and Communications Technology, Japan
[2] ATR Spoken Language Comm. Labs, Japan
[3]Nagoya Institute of Technology, Japan
[4]Kyoto City University of Arts, Japan
[5]Nara Institute of Science and Technology, Japan
[6]KDDI Research and Development Labs, Japan

{shinsuke.sakai,jinfu.ni,ranniery.maia,satoshi.nakamura}@atr.jp
tokuda@ics.nitech.ac.jp,minoru.tsuzaki@kcua.ac.jp
tomoki@is.naist.jp,hisashi.kawai@kddilabs.jp

Abstract

This paper presents a corpus-based approach to communicative speech synthesis. We chose “good news” style and “bad news” style for our initial attempt to synthesize speech that has appropriate expressiveness desired in human-human or human-machine dialog. We utilized 10-hour “neutral” style speech corpus as well as smaller corpora with good news and bad news styles, each consisting of two to three hours of speech from the same speaker. We trained target HMM models with each style and synthesized speech with unit databases containing speech with the relevant style as well as neutral speech. From the listening tests, we found out that intended communicative styles were comprehended by listeners and that considerably high mean opinion score on naturalness was achieved with rather small, style-specific corpora.

1. Introduction

Corpus-based approaches to speech synthesis have been very popular in the past decade and concatenative synthesizers have been especially successful due to its high naturalness [1, 2, 3, 4]. After achieving highly natural-sounding synthetic speech, however, the research and user communities of speech synthesis have become more aware about the issues with using speech synthesizers that speaks in an articulate but uniform manner in all the situations in human-machine dialogs or machine-mediated human-human dialogs. Research efforts on expressive and emotional speech synthesis, therefore, have become more and more active these days [5, 6]. We are aiming at developing speech synthesis technology that is useful for human-machine dialogs such as those in speech-enabled automatic conversational services as well as machine-mediated human-human dialogs such as conversations through the speech-to-speech translation system [7]. To investigate the possibilities of achieving synthetic speech appropriate for the communicative purposes in those systems, we looked at different styles of spoken communication such as conveying *good news*, *bad news*, and *focus* (or emphasis) [5], and selected good news and bad news for the styles to handle in our first attempt at synthesizing communicative speech. Part of the reason that we did not choose the focus was that it seems the objective of making some part of the utterance more salient than the others is often achieved by some other linguistic means such as using a different syntactic structure or adding small function words, rather than prosodic means in Japanese, which was the first target language for our communicative speech synthesis efforts.

In this paper, we report our initial attempt at communicative speech synthesis in the framework of XIMERA, a concatenative speech synthesis system [4]. We developed two-hour additional speech corpora in good news and bad news styles and trained HMM target models using these corpora. We also used these corpora together with the 10-hour corpus of neutral speech to generate speech with communicative styles. We tested how much desired styles were achieved and how much of naturalness was maintained by subjective listening tests. In the rest of the paper, we introduce the XIMERA concatenative speech synthesis system, followed by a description of the present approach to communicative speech synthesis. We then report on the experiments followed by the conclusion.

2. XIMERA

The block diagram of Figure 1 shows the main procedures conducted by XIMERA. Like most concatenative TTS systems, XIMERA is composed of four major modules, namely text processing, prosodic parameter generation, segment selection, and waveform generation modules. The target languages of XIMERA are Japanese, Chinese and English. Although the framework of corpus-based synthesis is language independent, most modules, in reality, must be developed or tuned for a target language. The language dependent modules comprise text processing, acoustic models for prosodic parameter generation, speech corpora, and the cost function for segment selection. The search algorithm for segment selection is also related to the target language via the cost function.

2.1. Text processing

The text processing module consists of three sub-modules for morphological analysis, rough syntactic analysis, and pronunciation and accent generation. The morphological analysis is conducted based on a bigram language model and a morpheme dictionary consisting of 239,591 Japanese or 195,959 Chinese entries. The rough syntactic analysis determines (1) a depen-

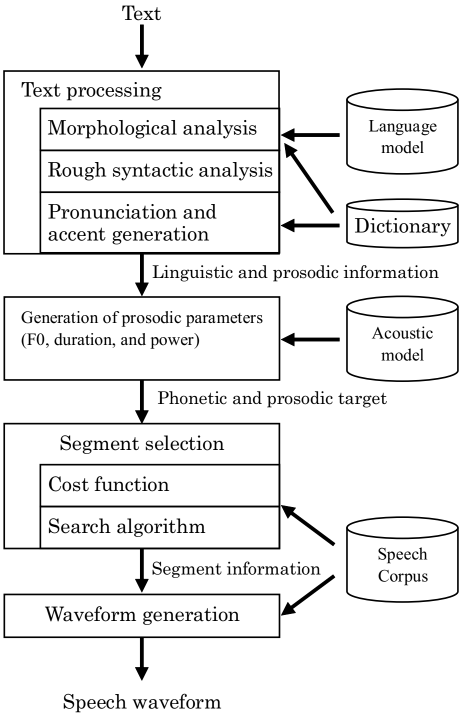


Figure 1: Main procedures performed by the XIMERA TTS system.

dency between adjacent words, which is mainly used for F0 generation, and (2) clause boundaries, which is mainly used for pause insertion. The pronunciation generation determines the readings of homographs and euphonic changes of unvoiced to voiced sounds. The accent generation determines the accent type of an accentual phrase based on accent types and the accent concatenation features of the constituent morphemes.

2.2. Generation of prosodic parameters

Prosodic parameters, namely $F0$, phone duration, and power, are generated according to the HMM-based speech synthesis technique [8, 9]. In other words, XIMERA includes an HMM-based synthesizer whose purpose is to produce the duration and power of the final concatenated waveform. Therefore, each HMM observation vector is composed of: (1) power; (2) mel-cepstral coefficients (without the 0-th coefficient); and $F0$. The generated parameters are also included in the concatenation cost for target selection.

2.3. Segment selection

2.3.1. Processing unit

The minimal processing unit is a half-phone [10, 11]. For Japanese synthesis, concatenation at a C-V boundary is inhibited by definition of the cost function. Moreover, a half phone unit in the resultant unit sequence should be at least either followed or preceded by a unit that was adjacent to it in the original speech corpus.

2.3.2. Cost function

The cost function of a sentence for segment selection is given by

$$C_g = \frac{1}{N}\sum_{i=1}^{N} C_l(u_i, t_i)^p, \qquad (1)$$

where N denotes the number of targets in the sentence, C_l denotes a local cost at the target t_i, and u_i and t_i respectively denote the i-th target and segment candidate. The power p was determined to be 2 as a result of perceptual experiments [12]. The local cost is given by

$$\begin{aligned} C_l(u_i, t_i) = w_{F0}C_{F0}(u_i, t_i) + w_{dur}C_{dur}(u_i, t_i) + \\ w_{cen}C_{cen}(u_i, t_i) + w_{F0c}C_{F0c}(u_i, t_i) + \\ w_{env}C_{env}(u_i, t_i) + w_{spg}C_{spg}(u_i, t_i), \qquad (2) \end{aligned}$$

where $C_{F0}(u_i, t_i)$, $C_{dur}(u_i, t_i)$, and $C_{cen}(u_i, t_i)$ respectively denote errors in $F0$, segment duration, and spectral centroid between the target and a segment candidate; representing therefore the target costs. On the other hand, $C_{F0c}(u_i, t_i)$, $C_{env}(u_i, t_i)$, and $C_{spg}(u_i, t_i)$ respectively denote discontinuities of $F0$, phonetic environment, and spectrum between adjacent segments; representing the concatenation costs. w_{F0}, w_{dur}, w_{cen}, w_{F0c}, w_{env}, and w_{spg} are corresponding weights for the local costs. Mappings from acoustic measures into the above local costs and weights were optimized based on perceptual experiments [13].

2.3.3. Search

The optimal sequence of waveform segments is searched for by using the Viterbi algorithm [14]. A problem due to large corpora is the heavy computation load required for evaluating candidate segments. To reduce the amount of computation, pre-selection based on target sub-costs is adopted.

2.4. Signal processing

XIMERA does not utilize prosodic modification of the final concatenated waveform. Toda et al. reported in [15] that the unnaturalness caused by prosodic modification algorithms, such as STRAIGHT [16], degrades the synthesized speech when the corpus size is greater than two hours. Therefore the waveform generation module is based on simple waveform concatenation. The concatenation point is searched for within a 5-ms range around the segment boundaries so that a short-term cross-correlation coefficient is maximized.

3. Communicative speech synthesis with XIMERA

We developed corpora of good news and bad news styles with the same speaker that we recorded 60 hours of neutral speech. Due to the limited time available for developing the prompt text, we reused part of the existing prompt text. The set of prompt sentences were equivalent to those used for a 2.6-hour subset of the existing neutral speech. Roughly 50% of the prompts were newspaper sentences, 25% were phonetically balanced sentence set, and the rest were travel conversation and novel sentences. The sentences in written form were modified to have conversational sentence ends.

The procedures of database collection, correction, labeling and phonetic segmentation, were conducted as described in [4], following the same directions employed for the construction of

Table 1: *Speech corpora used in this experiment.*

Speech corpus	ID	# utterances	# phones	Size
Neutral	N10	12,169	515,845	10 h
Neutral	N2	1,962	135,142	2.6 h
Good news	G2	1,962	139,551	3.0 h
Bad news	B2	1,962	138,558	3.2 h

Table 2: *The corpus sizes for training the target HMMs.*

Database style	Size (h)	# utterances	# phones	# feature labels
Neutral	2.3	1,807	118,701	117,638
Good news	2.8	1,852	128,468	126,962
Bad news	2.7	1,726	118,640	117,070

Table 3: *Description of the six databases for use in this test.*

ID	Size	Content
G2	3.0	3 h “good news” speech
G2+N10	13	3 h “good news”+10h neutral speech
N2	2.6	2.6 h neutral speech
N10	10	10 h neutral speech
B2	3.2	3.2 h “bad news” speech
B2+N10	13.2	3.2h “bad news”+10 h neutral speech

Table 4: *Combination of the three HMM-based targets with the six unit databases to form eight systems.*

Style	G2	G2+N10	N2	N10	B2	B2+N10
G	●	●		●		
N			●	●		
B				●	●	●

the original XIMERA database. The sizes of the corpora developed, with good-news and bad-news styles, were 3.0 and 3.2 hours, respectively.

4. Experiments and results

4.1. Goal

The effectiveness of an approach can be evaluated by some designed experiments. For this purpose, we built several TTS systems from the corpora listed in Table 1, under XIMERA framework, and conducted two perceptual experiments to investigate the ability of conveying communicative speech synthesis with a certain degree of naturalness. Therefore, Experiment I is intended to evaluate the naturalness of synthetic speech in each target speaking style, namely “good news”, “bad news” and neutral styles[1], whereas Experiment II is focused on rating the appropriateness of conveying “good news” and “bad news” by the synthetic speech.

4.2. Prosody generation modules and unit databases

In order to synthesize “good news”, “bad news” and “neutral” speech, we trained contextual HMMs for three style-specific prosody generation modules. Table 2 shows the amount of database used to train each style and the resulting number of feature labels.

The unit database were generated from the database sets shown in Table 1. Note that two versions of “neutral” database were generated, one with two hours and other with ten hours.

4.3. Different system versions used in the experiments

Through a few combination of the corpora shown in Table 1, the six databases listed in Table 3 were developed. The database N2 is a sub-set of N10. Further, by combining the HMM-based prosody generation modules of Table 2 with these databases, eight TTS systems are used in this experiment. Table 4 illustrates which acoustic model is combined with each individual database. In the following, each system is described.

1. **System G–G2**
   - Target: “good news”;
   - Unit database: G2.
2. **System G–G2+N10:**
   - Target: “good news”;
   - Unit database: G2+N10.
3. **System G–N10:**
   - Target: “good news”;
   - Unit database: N10.
4. **System N-N2:**
   - Target: “neutral”;
   - Unit database: N2.
5. **System N-N10:**
   - Target: “neutral”;
   - Unit database: N10.
6. **System B–B2:**
   - Target: “bad news”;
   - Unit database: B2.
7. **System B–B2+N10:**
   - Target: “bad news”;
   - Unit database: B2+N10.
8. **System B–N10:**
   - Target: “bad news”;
   - Unit database: N10.

The use of style-specific target HMMs combined with neutral database is intended to test how good performance can be achieved by use of a neutral speech corpus only.

4.4. The test sentences

The eight TTS systems above were used to supply synthetic speech for use in a listening test. We chose ten *ambiguous* Japanese sentences. These sentence can be literally interpreted as “good news”, “bad news”, or “neutral.” The use of *ambiguous* sentences is expected to be suited for testing synthetic speech in delivering an intended speaking style. Since there are eight versions of synthetic speech for each sentence, 80 distinct stimuli in total were yielded . The 80 stimuli are divided into two groups, 40 stimuli for each, using a different randomized order across groups. One group was used for Experiment I and the other for Experiment II.

[1] Hereafter referred to as G, B, and N, respectively.

4.5. Subjects

Twelve listeners participated this listening test, six male and six female speakers, all of whom are Japanese natives with normal hearing. These stimuli were presented to listeners with headphones in a silent office. The listeners were allowed to listen to a few samples before starting this test so as to get some idea of the quality of synthetic speech in the three styles. During this listening test, they could listen to each stimulus as many times as they liked, but could not go back and forth anyway.

4.6. Results and discussion

In Experiment I, the listeners were asked to rate the naturalness of synthetic speech on a 5-point scale from 1, the worst naturalness, to 5, very natural. In Experiment II, the same listeners were then instructed in the listening task of evaluating the appropriateness of synthetic speech in conveying “good news”, neutral news, and “bad news” on a 7-point scale from -3 (very good “bad news”), 0 (neutral), and 3 (very good “good news”).

Figure 2 shows Mean Opinion Scores (MOS) for each of the TTS systems enumerate above, and Figure 3 shows the MOS obtained in Experiment II for each system. Table 5 lists the number of stimuli in percentage evaluated as “bad news”, “neutral,” or “good news” on the 7-point scale for each system.

Several observations may be made from the experimental results. Firstly, synthetic speech in “good news” style has high naturalness, even we use the “neutral” unit database. The MOS obtained by System G-N10 in Figure 2 illustrates this fact.

Secondly, when both the targets and databases were built from style-relevant speech corpus, the resulting systems achieved better performance than the others. For instance, System G-G2 outperforms systems G-G2+N10 and G-N10, and System B-B2 outperforms systems B-B2+N10 and B-N10. The degradation in naturalness from systems G-G2 to G-G2+N10, and from B-B2 to B-B2+N10, perhaps might be partly caused by the unit selection algorithm, since the former was included in the latter. On the other hand, the MOS values obtained in Experiment II for rating the appropriateness of intended styles were quite similar in both systems G-G2 and G-G2+N10 as well as systems B-B2 and B-B2+N10, as shown in Figure 3.

Thirdly, when focusing on the naturalness of systems G-G2, N-N2, and B-B2, which are similar in speech corpus size, systems with “good news” and “bad news” styles achieved considerable better performance than the neutral system N-N2. This result might indicate that appropriate styles could possibly improve the naturalness of the synthesized speech. In other words, an effective way to improving naturalness in small corpus speech synthesis is to generate synthesized speech in varied styles.

Finally, while “good news” speech presented better naturalness than “bad news,” “bad news” speech could give clearer impression than “good news” speech, according to Figure 3.

Basically, the results of Experiment II showed that listeners could correctly identify synthesized speech in a particular style ranging from 98.4% for “bad news” to 66.7% for “good news”.

The results also imply that there is an overlap for distinguishing between “good news” and “neutral” styles. This can be supported by the numbers in Table 6, which shows how many units were selected from N10 in both systems B-B2+N10 and G-G2+N10. As shown in Table 6, 59% of units were selected from N10 when synthesizing the sentences in “good news” style, while only 3% of units were selected from N10 when synthesizing “bad news.” Further, a deeper examination showed that the vocal range in uttering “bad news” style (by the same speaker) is sharply narrowed and the mean is considerably lowered. Table 7 lists the voice ranges measured from B2, N2, and G2, and some examples are displayed in Figure 4.

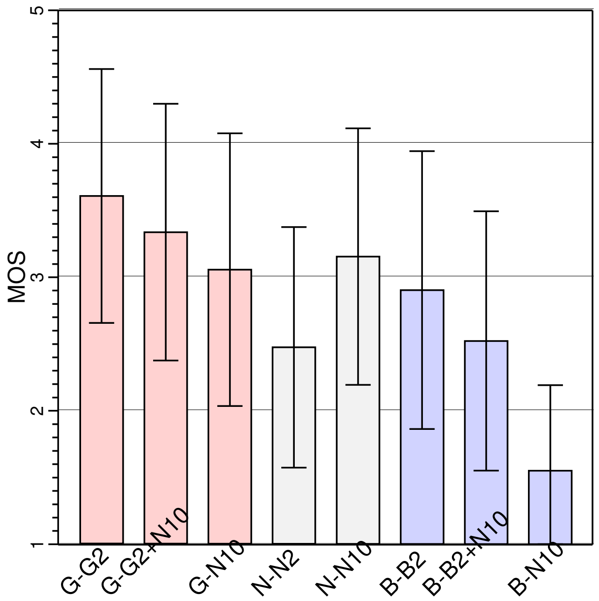


Figure 2: *Mean Opinion Scores and standard deviations for each of the synthesis systems in Experiment I described in the text.*

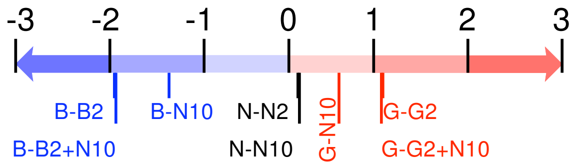


Figure 3: *Mean Opinion Scores for each of the synthesis systems in Experiment II described in the text.*

5. Conclusions

In this paper, we presented a corpus-based approach to communicative speech synthesis. We chose “good news” style and “bad news” style for our initial attempt to synthesize communicative speech and collected speech corpora with those styles. Target HMMs were trained with these style-specific corpora, whereas we also made use of neutral speech corpus for building style-specific unit databases in order to know how this existing resource can be utilized to generate speech with expressions relevant in the communication. From the listening tests, we found out that intended communicative styles were comprehended by listeners and that considerably high mean opinion score on naturalness was achieved with rather small, style-specific corpora. Currently we need to have separate model trees for each of the communicative styles. We plan to examine the possibilities of having a single model tree where styles are handled as one of the features for clustering HMM target models.

Table 5: *Percentage of stimuli evaluated as individual levels in Experiment II.*

System ID	−3	−2	−1	0	1	2	3
B–B2	26.7	41.7	30		1.6		
B–B2+N10	26.7	43.3	23.3	6.7			
B–N10	16.7	25	40	11.7	5	1.7	
N–N2			15	63.3	18.3	3.3	
N–N10			16.7	60	18.3	5	
G–G2			3.3	28.3	30	30	6.7
G–G2+N10			3.3	28.3	35	28.3	5
G–N10			0.86	35	46.7	10	

Table 6: *Percentage of units selected from the neutral speech corpus when synthesizing "bad news" and "good news" styles.*

Style	System	Units selected from subset N10
Bad news	B–B2+N10	3%
Good news	G–G2+N10	59%

6. Acknowledgements

The authors are grateful to J. Kunita for various contributions to the corpus development and implementation of the system described in this paper.

7. References

[1] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," pp. 373–376, 1996.

[2] M. Chu, H. Peng, Y. Zhao, Z. Niu, and E. Chang, "Microsoft Mulan – a bilingual TTS system," pp. I–264–I–267, 2003.

[3] E. Eide *et al.*, "Recent improvements to the IBM trainable speech synthesis system," pp. I–708–I–711, 2003.

[4] H. Kawai, T. Toda, J. Ni, M. Tsuzaki, and K. Tokuda, "XIMERA: a new TTS from ATR based on corpus-based technologies," in *Proc. of ISCA Speech Synthesis Workshop*, 2004.

[5] J. Pitrelli *et al.*, "The ibm expressive text-to-speech synthesis system for american english," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1099–1108, 2006.

[6] C. Wu *et al.*, "Voice conversion using duration-embedded bi-hmms for expressive speech synthesis," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1109–1116, 2006.

[7] S. Nakamura *et al.*, "The atr multi-lingual speech-to-speech translation system," *IEEE Trans. on Speech and Audio Processing*, vol. 14, no. 2, pp. 365–376, 2006.

[8] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. of ICASSP*, 2000.

[9] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. of EUROSPEECH*, 1999.

Table 7: *F0 ranges for the three styles in a speaker.*

Style	Mean	Standard deviation	F0 range
Bad news	161 Hz	33 Hz	(105 Hz, 242 Hz)
Neutral	249 Hz	56 Hz	(131 Hz, 365 Hz)
Good news	283 Hz	61 Hz	(167 Hz, 407 Hz)

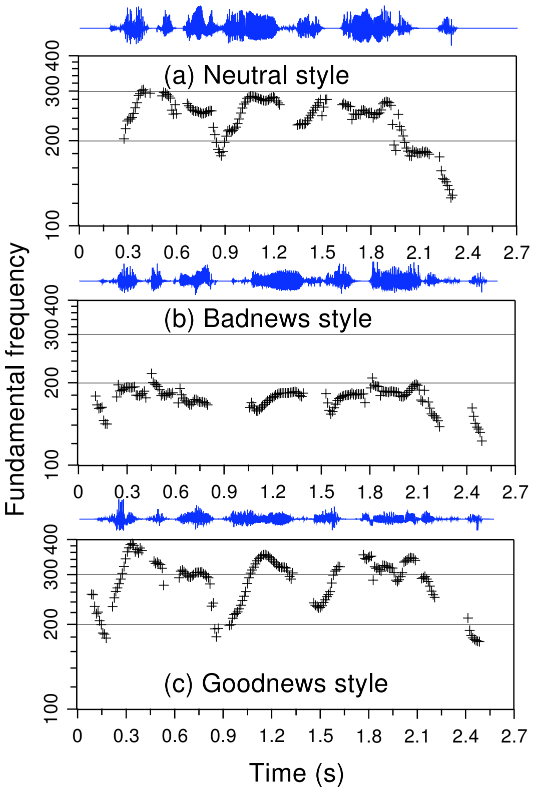


Figure 4: *Examples of F0 contours for a Japanese sentence produced by a native in (a) neutral styles, (b) "bad news" styles, and (c) "good news" styles.*

[10] M. Beutnagel, A. Conkie, and A. Syrdal, "Diphone synthesis using unit concatenation," in *Proc. of International Workshop on Speech Synthesis*, 1998.

[11] T. Toda, H. Kawai, M. Tsuzaki, and K. Shikano, "Unit selection algorithm for Japanese speech synthesis based on both phoneme unit and diphone unit," in *Proc. of ICASSP*, 2002.

[12] T. Toda, H. Kawai, M. Tsuzaki, and K. Shikano, "Segment selection considering local degradation of naturalness in concatenative speech synthesis," in *Proc. of ICASSP*, 2003.

[13] T. Toda, H. Kawai, and M. Tsuzaki, "Optimizing subcost functions for segment selection based on perceptual evaluations in concatenative speech synthesis," in *Proc. of ICASSP*, 2004.

[14] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. of ICASSP*, 1996.

[15] T. Toda, H. Kawai, and M. Tsuzaki, "Effectiveness of prosodic modification in concatenative text-to-speech syn-

thesis," in *Proc. of the Fall Meeting of the Acoust. Soc. of Japan*, 2003. In Japanese.

[16] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, Apr. 1999.

Automatic Exploration of Corpus-Specific Properties for Expressive Text-to-Speech: A Case Study in Emphasis.

Raul Fernandez and Bhuvana Ramabhadran

IBM TJ Watson Research Center
Yorktown Heights, NY 10598
{fernanra,bhuvana}@us.ibm.com

Abstract

In this paper we explore an approach to expressive text-to-speech synthesis in which pre-existing expression-specific corpora are complemented with automatically generated labels to augment the search space of units the engine can exploit to increase its expressiveness. We motivate this data-discovery approach as an alternative to an approach guided by data collection, in order to harness the full usefulness of the expressiveness already contained in a synthesis corpus. We illustrate the approach with a case study that uses *emphasis* as its intended expression, describe algorithms for the automatic discovery of such instances in the database and how to make use of them during synthesis, and, finally, evaluate the benefits of the proposal to demonstrate the feasibility of the approach.

1. Introduction

There has been recent interest in text-to-speech (TTS) research to address the need of speech synthesizers to not just sound natural and intelligible, but also to convey suitable expressions. Rather than being a decorative flourish, it can be argued that producing expressive synthetic speech is fundamental, not only to ensure that there is a match between the linguistic content of the text and the tone of voice in which it is delivered, but also to engage the user and maintain him motivated in the listening experience. This is particularly relevant as we move beyond simple short-prompts interactive scenarios (*e.g.*, a help desk application) toward more challenging, cognitively-taxing uses of text-to-speech technology (*e.g.*, a synthesized news podcast).

The IBM Expressive TTS System [1] is capable of generating speech in expressive styles suitable for conveying good news, conveying bad news, asking a question, or delivering emphasis. The system relies on augmenting its baseline speech corpus with smaller expression-specific corpora of speech, large enough to derive prosody models and to augment the search space with explicitly tagged expressive units. Although this approach works quite well, it is impeded by the fact that expanding the repertoire of expressions, or increasing the size of an existing corpus, is costly in terms of studio time and footprint size. As an alternative to indiscriminate data collection, we recently argued for an approach in which existing databases are exploited for the occurrence of (possibly more subtle) examples of expressions that are already contained in the database [2]. In this paper, we follow this philosophy and apply machine learning algorithms to our speech databases to automatically explore and learn new labels that can be used by the engine at run time to expand the range of its expressiveness. The purpose is not to discount additional data collection as a viable alternative, but rather to motivate exploring the overlap that there may already exist between the existing databases and a given category of interest, before proceeding with a data-collection approach. We will motivate and illustrate this approach with the case study of *emphasis*, an ”expression” for which we already have a corpus of suitable recordings which can be used as a basis for training learning schemes.

The organization of this paper is as follows. In Section 2 we present an overview of the expressive component of the IBM TTS System. We discuss the idea of mining attributes from the dataset in Section 3, present and evaluate algorithms for the automatic labeling of emphasis in speech, and discuss how to make use of this output at run time. In Section 4 we evaluate the proposal and discuss results and finally conclude in Section 5.

2. Expressive System Overview

In this section we review only the architectural components of the TTS engine that are responsible for addressing the generation of expressive (acoustic and prosodic) targets and the expressive unit selection at synthesis time. For a more complete overview of the IBM TTS system, the reader is directed to, *e.g.*, [1].

The baseline corpus used to build the core concatenative database (henceforth referred to as the *neutral* corpus) consists of approximately 10 hours of audio recorded from a professional speaker delivered in a lively, friendly style. In addition to the neutral corpus, the system makes use of smaller, expression-specific corpora containing approximately 1 hour of audio. Some of the expressions we have considered are *good news*, *apologies*, *confusion* and *emphasis*. During the concatenative database build process, the synthesis units in the database (which, in the case of the IBM TTS system correspond to subphonemic speech segments aligned with a single state of a typically 3-state HMM) are labeled with a discrete-valued attribute vector containing, *e.g.*, linguistic, expressive and other kinds of information about the units. Fig. 1 contains an example of a 3-dimensional vector that illustrates the kind of attribute information the system could make use of. Each *attribute type* has a default *attribute value* from its value set associated with it (shown underlined in Fig. 1). Synthesis units that are not explicitly labeled are assumed by the engine to bear the default value for the unlabeled attribute in question. The attribute vector definition (*i.e.*, the list of attributes, the value set each is defined on as well as its default value) is fully customizable to the application and can be specified through an external configuration file.

Separate attribute-specific prosody models are also built at this stage. The current implementation of the engine allows for

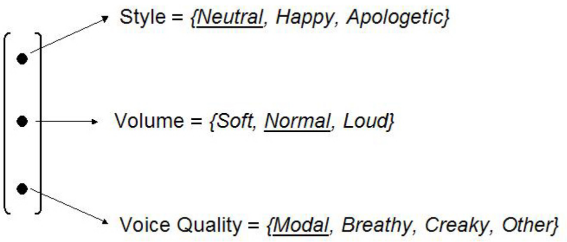

Style = {Neutral, Happy, Apologetic}
Volume = {Soft, Normal, Loud}
Voice Quality = {Modal, Breathy, Creaky, Other}

Figure 1: *Example of a 3-dimensional attribute vector, and the attribute value set for each component.*

the *style* attribute illustrated in Fig. 1 to be the attribute dimension that acts as a switch between different prosody models at run time to generate different prosody targets. The full dataset is therefore segregated into separate *style*-specific subsets, and the standard prosody models for pitch, duration (and, optionally, energy) are constructed for each subset [1] .

During synthesis, the input, assumed to be in the form of marked-up text, is processed by an XML parser. The resulting tags are used to assign attribute values to different words, and these values are propagated down to the subphonemic synthesis-unit level. The plain text is then processed by our standard rules-based front-end to produce phonetization and symbolic linguistic descriptions. Acoustic models, previously built while assembling the dataset, are used to generate a suitable list of context-dependent synthesis-unit candidates over which to evaluate the search, and the style-specific prosody models are invoked in order to generate prosody targets as a function of the corresponding style attribute value for each unit. Since prosody alone does not fully convey the desired style [3], we also include the smaller set of segments from each of the attributes in the search to allow the dynamic programming algorithm to evaluate trade-offs between matching different components of the target cost. That is, all segments from all attributes are considered in the search (as long as they fit the context-dependent constraints imposed by the acoustic models), and the attribute match is assessed through an additional component of the cost function, $C(\mathbf{t}, \mathbf{o})$, introduced to penalize using a speech segment labeled with attribute vector $\mathbf{o}$ when the target is labeled with attribute vector $\mathbf{t}$. Since all attributes are discrete-valued, this cost can be summarized by means of a square matrix. The elements of this matrix are usually tuned empirically.

3. Attribute Mining

The previous section highlighted an architecture that makes use of attribute annotations on the synthesis corpus to facilitate attribute-driven prosody target generation and dynamic programming search. We would now like to turn to the source of knowledge for these attribute labels. In one particular case, these attributes could be present in the data by experimental design and their specific occurrence be known *a priori*. Such is the case, for instance, of the approach we have followed in the past for collecting expression-specific corpora [1]: a professional reader is instructed to read text in a particular expressive style (*e.g.,* apologetic) in a recording studio and is closely supervised to make sure she delivers the intended expression. We have recently motivated going beyond this kind of *a priori* knowledge of the descriptors, and moving toward *discovering* them in the corpus to increase the range, and flexibility, of expressive concatenative text-to-speech [2]. A similar approach can be found in the work by Campbell and Marumoto [4], where prosodic and acoustic characteristics associated with different emotions are learned from emotion-specific corpora and then used to relabel segments in other databases. As a particular instance of this approach, consider the scenario where attributes can be expected to be in the dataset, and the occurrence of the different values for that attribute can be arrived at through a learning or rules-based mechanism. Imagine, as an example, labeling the speaking rate of every synthesis unit in the corpus as *slow, medium,* or *fast*. One could establish this discretization by some simple rules given knowledge of *e.g.,* text alignments, phone classes and speaker's average speaking rate. In the most general case of this approach, however, we may or may not have knowledge about whether the attribute is reflected in the dataset, the degree to which it is, and where it occurs. In this case, we wish to *mine* the corpus to discover these attribute values automatically.

In previous work [2], we focused primarily on attributes that could be derived from the text itself (and from the symbolic description thereof produced by the front-end analysis). While this is a reasonable first step, it has the limitation that we ultimately wish to establish properties of the *spoken* synthesis units; using the text string as proxy for analysis can only provide an approximation given the multiple prosodic realizations that can exist for a given syntactic structure [5]. In the work presented here, we are following the approach of mining attributes from the corpus by focusing on automatic discovery of properties of the spoken units. We are illustrating this with a case study in *emphasis*. The motivation for focusing on this type of attribute is manifold: First, emphasis is one of the expressive labels for which we already have an existing smaller corpus of in-studio recordings with professional speakers explicitly instructed to produce it. This corpus can therefore be exploited as learning material for data-driven algorithms; that is, automatically discovering this attribute in the larger synthesis corpus can be bootstrapped to a part of the corpus where we have very high confidence the attribute is present. Secondly, emphasis is a fairly pervasive attribute of spoken language, and, although different speakers can vary in the manner and degree of the realization, we expected that, at least for some of the speakers with a more ”lively” reading style, we would be able to find quite a few exemplars in the 10-hour baseline recordings. Finally, being able to properly produce emphasis is applicable to many text-to-speech scenarios where we would like to improve the expressiveness. This includes not just the canonical case of *contrast*, but also cases where you may want to highlight a rare word or increase the liveliness of a sentence by, *e.g.,* treating focus words differently, or speech-to-speech applications where user-intended focus or emphasis in the original language should be preserved and synthesized in the output target language.

3.1. Emphasis Classification

In this section we turn to the details of how to annotate the baseline corpus with emphasis labels. As mentioned above, since we have at our disposal smaller corpora containing emphasis annotations that we can use as training and development data, our approach will be to implement data-driven algorithms for automatically learning a mapping from a set of input predictor features to a binary emphasis label. What is understood in this work by emphasis is primarily a perceptual phenomenon. We are not adhering to any theoretical descriptions of how emphasis is accomplished. Rather, we are interested in modeling the characteristics of the speech obtained under the following conditions: A professional speaker is instructed to read a sentence in which some words are meant to be emphasized. When (usually) two judges present in the studio, plus the speaker herself, are satisfied with the outcome (*i.e.,* when the intended words, and only those, are perceived as emphatic), the recorded sentence is added to the emphasis corpus; otherwise, the speaker reads the sentence again. The script is carefully designed to ensure as much as possible that emphasis is requested for words where it would be natural to produce it. This avoids unnatural realizations that would be difficult for the speaker to produce, and which might hurt the quality at synthesis time.

Emphasis is treated here as a binary-valued word-level attribute (*i.e.,* a word is emphasized, or not), and the classification scheme implemented here is based on a set of features extracted at the word level. We realize that emphasis can be a continuous-valued attribute, or, since the architecture presented in the previous section relies on discrete attributes, that it would at least admit a multi-level discrete description rather than a binary one. However, for the purpose of the modeling done in this paper, and the instructions delivered to the speakers, it was treated as a binary variable.

The feature set is meant to capture some variations in pitch, energy and duration (the latter roughly modulated by broad phone classes) which are likely to be acoustic-prosodic correlates of emphasis. The full list of features is given below:

1. Average pitch in word, normalized by speaker's average pitch
2. Median pitch in word, normalized by speaker's average pitch
3. Standard deviation of pitch in word, normalized by speaker's pitch standard deviation
4. Pitch range over word, normalized by speaker's pitch standard deviation
5. Word duration, in seconds
6. Word duration in seconds, normalized by the number of phones in word
7. Word duration in seconds, normalized by the number of vowels in word
8. Ratio of vowel duration to overall duration in word
9. Previous value normalized by the vowels-to-phones ratio in word
10. Root-Mean-Squared energy value of word

Since one of the applications we envision for this kind of system is to be able to label corpora for speakers for whom we do not have any development data, we have avoided highly speaker-dependent features, such as absolute pitch-based features, from this list. However, for the case where the training and testing speaker were the same, we did consider these additional features, only to discover that they did not improve the performance. We have, therefore, omitted them from the final system and from the rest of the discussion.

We explored a variety of classification schemes on this task and found that K-Nearest-Neighbor and Support Vector Machines were the two top performers, in that respective order, over other classifiers like Decision Trees or Naive Bayes. Evaluations were done in all cases using 10-fold cross-validation. The fact that a simple K-Nearest Neighbor (with $K \approx 10$) consistently performs at the top is possibly due to the fact that, given the good amount of data we have, its performance is starting to approximate that of the Bayesian posterior for this feature set. Nonetheless, to explore the possibility of benefiting from classifier combination, we stacked the outputs of these 2 top performers into a combination scheme using a Naive Bayes classifier. The first stage of the training, therefore, maps the input feature space listed above into two (intermediate) estimates, $P_{KNN}(\omega|\mathbf{x})$ and $P_{SVM}(\omega|\mathbf{x})$ of the class posterior probability whereas the second (output combination) stage takes these estimates as input features and maps them to one final class posterior $P_{NB}\left(\omega\middle|P_{KNN}(\omega|\mathbf{x}), P_{SVM}(\omega|\mathbf{x})\right)$. A word with feature vector $\mathbf{x}$ is assigned to the class ω which satisfies the Bayes decision rule:

$$\hat{\omega} = \arg\max_{\omega} P(\omega) P_{NB}\left(P_{KNN}(\omega|\mathbf{x}), P_{SVM}(\omega|\mathbf{x})\middle|\omega\right) \tag{1}$$

where

$$\begin{aligned} P_{NB}\left(P_{KNN}(\omega|\mathbf{x}), P_{SVM}(\omega|\mathbf{x})\middle|\omega\right) \quad = \quad & \frac{P(\omega)}{Z} \times \\ & P_{KNN}(\mathbf{x}|\omega) \times \\ & P_{SVM}(\mathbf{x}|\omega) \end{aligned} \tag{2}$$

and Z is a normalizing constant to ensure the posterior sums up to 1. Analysis of the error distribution of the two intermediate classifiers reveals that there is considerable overlap between their outputs. This lack of complementarity, therefore, limits the usefulness of a classifier combination scheme and does not satisfy the independence assumption on which the success of the Naive Bayes classifier depends. Combining classifiers only offered a modest 2% absolute improvement. However, since the classification is done off-line, we have accepted the extra computational cost in exchange for the minor improvement. The results reported below are all based on the final output of the combining classifier.

We trained and tested two separate speaker-dependent systems, one for a male speaker and one for a female speaker, with 15,204 and 13,278 word tokens respectively. The empirical prior distribution for emphasized words for each set was 22%. Although this number may seem low for a corpus that was expressly designed to collect emphasis, it is challenging to maintain the naturalness and flow of the sentences during the data collection process, as explained above, when emphasized words appear much more frequently than this. Performance was assessed on the training set by means of 10-fold cross-validation. The confusion matrices showing the performance for the two speakers are shown in Tables 1 and 2. The systems achieve an overall recognition rate of 91.17% (male speaker) and 89.86% (female speaker).

		Labeled	
		Emph	Non-Emph
True	Emph	2710	602
	Non-Emph	741	11151

Table 1: Confusion matrix showing the emphasis classification results for the male speaker. Overall recognition rate is 91.17%. Prior class probabilities are $[0.22, 0.78]$ for emphasis and non-emphasis respectively.

A class-dependent analysis of the systems showing different performance measures, Recall (Rec), Precision (Prec), False Positive Rate (FP) and F-Measure (F-Meas), is also shown in

		Labeled	
		Emph	Non-Emph
True	Emph	2295	578
	Non-Emph	768	9637

Table 2: Confusion matrix showing the emphasis classification results for the female speaker. Overall recognition rate is 89.86%. Prior class probabilities are $[0.22, 0.78]$ for emphasis and non-emphasis respectively.

Table 3. For a class ω, these performance measures are defined as follows:

$$
\begin{aligned}
\text{Recall} &= \frac{\text{Num. correctly labeled } \omega}{\text{Total number of actual } \omega} \\
\text{Precision} &= \frac{\text{Num. correctly labeled } \omega}{\text{Total number of predicted } \omega} \\
\text{False Positive Rate} &= \frac{\text{Num. incorrectly labeled } \omega}{\text{Total number of not } \omega} \\
\text{F-Measure} &= \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}
\end{aligned}
$$

We want, in general, to obtain both a high recall and a high precision measure (a combined fact that a high F-Measure ought to reflect) while minimizing the number of false positives. As we can see from Table 3, the best results for the Emphasis class are obtained for the male speaker with an F-Measure of 0.8 and a False Positive Rate of 0.06. The results for the female speaker are only marginally different. These numbers suggest that the feature set and classification scheme described here are doing a reasonable job at modeling the emphasis class, while still allowing some room for improvement in future iterations of this work. Further work could explore, for instance, how spectrally-derived features, such as energy in different spectral bands [6], contribute to the realization and perception of emphasis and can aid in its automatic classification.

Speaker	Class	Rec	Prec	FP	F-Meas.
Male	Emph	0.82	0.78	0.06	0.80
	Non-Emph	0.94	0.95	0.18	0.94
Female	Emph	0.80	0.75	0.07	0.77
	Non-Emph	0.93	0.94	0.20	0.94

Table 3: Performance figures (recall, precision, false-positives and F-measure) derived from the confusion matrices in Tables 1 and 2.

3.2. Building an Expressive System with Automatic Labels

We can apply the systems proposed in the previous section to the task of discovering examples of emphasis that may occur throughout the rest of the unlabeled baseline database. When we do this, we discover that approximately 8% to 10% of the words in this corpus receive the emphasis label. An empirical subjective analysis of the output of this labeling suggests that the results are better for the case of the male speaker, who speaks in a style that shows more demarcated alternation between emphasized and unemphasized words. The words that are automatically labeled as being emphasized are then given an attribute value that can be used at run time by the framework described in Section 2 to bias the search toward choosing segments with emphasis. Our approach is to keep a distinction between those units that belong to the small emphasis corpus from which the classifiers were built, and the units that are automatically labeled. The former are closely scrutinized in the studio during the recording session, and therefore carry a high confidence on the label. The confidence on the latter set is clearly limited and constrained by the performance of the automatic classifiers, which are always prone to have a margin of error. Since the main objective of this work is to increase the inventory of units with a particular attribute from which the search can choose at run time, the prosody models are not rebuilt in light of the additionally labeled data now available. Rather, we retain the models originally developed with the original emphasis data only. However, it remains to be explored whether this additional (noisier) data can offer improved prosody targets.

		Target			
		neutral	collEmph	labEmph	
Segment	neutral	0.0	0.3	0.2	· · ·
	collEmph	0.5	0.0	0.0	· · ·
	labEmph	0.5	0.1	0.0	· · ·
	· · ·	· · ·	· · ·	· · ·	0.0

Table 4: Attribute cost matrix to combine automatically- and hand-labeled expressions. Automatically-labeled segments are weighted differently than hand-labeled ones to reflect inaccuracies in labeling scheme.

Fig. 2 summarizes the procedure used to build an expressive system with emphasis: after the steps described in the previous section to build the emphasis classifiers, the baseline corpus is analyzed and augmented with emphasis annotations, and both the annotated baseline and emphasis corpora are combined to produce one final system. Provided we keep the labeled emphasis as a distinct attribute value, the reliability of the annotation scheme can be addressed at run time by choosing weights for the attribute cost matrix that reflect this uncertainty. This is illustrated by the sample cost matrix shown in Table 4. Here *collEmph* is used to describe the emphasis annotations attached to the emphasis-specific corpus collected in the studio whereas *labEmph* describes the annotations produced by the automatic labeling scheme. This is in theory a square matrix although, in practice, the *labEmph* label is unlikely to be requested as an explicit target: A user would mark up the text with a tag that translates to a *collEmph* request directly, not to a request for a target with labeled emphasis. *labEmph* acts as an additional annotation that is tied in some sense to *collEmph* by the system developer behind the user interface layer. However, the engine architecture allows us to directly make this kind of request if, for instance, we wish to test how well these labels alone produce the percept of emphasis. In this example, the 0.1 value in the matrix reflects the fact that, whenever a *collEmph* target is sent to the search, we penalize retrieving a *labEmph* unit slightly more than retrieving a *collEmph* unit (by definition a perfect match, and therefore 0 cost). If we wish to make the two labels equivalent, we can do so by making their two respective row entries identical.

4. Evaluation and Discussion

In order to test the usefulness of the proposed automatic mining approach, we designed a listening test where subjects were presented with pairs comprising one neutral sentence and one sentence containing emphatic words, and asked to make choices about the emphasis-carrying sentence. Three type of sentence

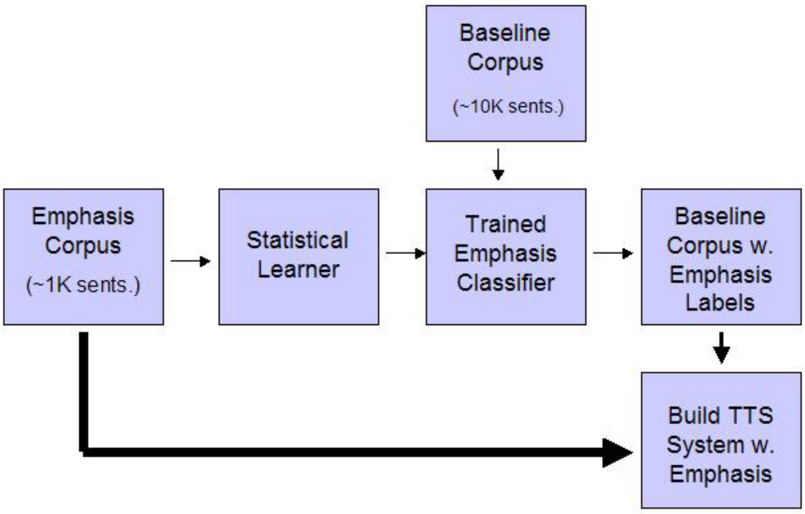

Baseline Corpus
(~10K sents.)
Emphasis Corpus
(~1K sents.)
Statistical Learner
Trained Emphasis Classifier
Baseline Corpus w. Emphasis Labels
Build TTS System w. Emphasis

Figure 2: *Building an expressive TTS system with collected and mined expression. The relative width of the two lines feeding into the final system is meant to illustrate the fact that we can assign different weights to these different data subsets based on our confidence of the labels.*

stimuli were used to make up the pairs:

- Type A: a sentence where no word was marked for emphasis was synthesized using the unannotated baseline corpus.
- Type B: a sentence where one or more words were marked for emphasis was synthesized using the unannotated baseline corpus, plus the collected emphasis corpus.
- Type C: a sentence where one or more words were marked for emphasis was synthesized using the baseline corpus automatically annotated with emphasis labels, plus the collected emphasis corpus.

The texts of 12 distinct sentences, each containing one or more words marked-up for emphasis, were synthesized as described above to produce 3 sets of stimuli (when synthesizing Type A sentences, the marked-up emphasis was ignored). Both the Type B and Type C versions of each sentence marked-up the same word or words for emphasis. We tried to mark-up words where emphasis might fall naturally but avoided contrastive-emphasis constructions since in those cases the text alone is often a predictor of where the emphasis should be realized. After generating these basic stimuli, 12 pairs were produced for 2 testing conditions, as follows:

- Condition 1: a pair consisting of 1 sentence of Type A (neutral) and 1 sentence of Type B (emphatic), both with the same text
- Condition 2: a pair consisting of 1 sentence of Type A (neutral) and 1 sentence of Type C (emphatic), both with the same text

A total of 24 pairs, 12 from each condition, were combined to produce one final set of listening samples. All run-time parameters were set to be the same for all conditions. A playlist was created by randomly interleaving the pairs from each condition, and by randomizing the order within each pair. Additionally, a second playlist was assembled by reversing the order within each pair from the first list. Thirty-one listeners took part in the test; 16 listened to the samples in the original order and 15 in the reverse order.

When synthesizing emphasis, we usually resort to making use of very brief pauses (usually in the order of 5 to 10 msecs.) around the emphasized words. This alone often suffices to create the impression of emphasis although the acoustic and prosodic realization of the units that follow are often at odds with this impression if no further emphasis units are used. Since the focus of this work has been on this last component of the emphasis realization (*i.e.*, using suitably labeled units to produce emphasis), we have left out the pauses around the emphatic words since we felt this effect might confound or overwhelm the effect we are trying to study. The implication is also that the stimuli become much harder to evaluate in this case since the listener might benefit, or expect, a salient break index around emphasized words [7].

During the test, listeners were given the chance to listen to each pair, repeatedly if they wished, and were told that each sentence in the pair *may* contain one or more words bearing emphasis. Their task was to select which sentence of the pair they thought best conveyed emphasis. The overall results from all 31 subjects are summarized, according to condition, in Table 5.

Condition	Neutral	Emphatic
1	229 (61.6%)	143 (38.4%)
2	181 (48.7%)	191 (51.3%)

Table 5: Results of listening test for conditions 1 and 2. Each cell contains the number of times (and percentage) that a particular type of sentence (neutral or emphatic) was preferred within each condition

This is a difficult listening identification task for some of the reasons we have already highlighted. Additionally, the listener's attention is not directed toward specific words in the pair so that he can contrast those words. Moreover, the word(s) that may be candidates for perceived emphasis in the first sentence may not be the same word(s) that the listener is considering as candidates in the second sentence (in which case he may have to resolve based on, *e.g.*, the relative degree of emphasis). In spite of this, it is surprising that, when evaluating Condition 1, subjects indicated 61.6% of the time that the neutral baseline sentence was the emphasis-bearing stimulus and only chose the emphatic sentence 38.4% of the time. Our hypothesis for why this is so is the following: in the emphatic samples of Condition 1 (sentences of Type B), we are highly biasing the search toward choosing emphasis units from a smaller inventory of units (the small studio recording), and trying to aggressively recruit units from this limited inventory for synthesis creates artifacts (*e.g.* clicks or warbles) that might interfere with the perception of emphasis.

However, when we compare across conditions, we see that there is a large improvement from 38.4% to 51.3% (statistically significant at the $p < 0.001$ level) [1] in the identification of intended emphasis when automatically labeled units are allowed to play an explicit role in the synthesis of emphatic words (sentences of Type C). Since our ultimate goal is to improve how accurately emphasis is conveyed, in practice we would adopt the hybrid approach described, where we use a combination of break-index and unit selection. This would allow us to bias less aggressively toward choosing the ”right” units while exploiting the perceptual salience of carefully placed pauses. However, the experiment we have carried out here demonstrates the advantage of exploring the synthesis corpus, by means of automatic expression-recognition algorithms, to extract examples of expressive units that can be found scattered throughout a large database, and which can be harnessed at synthesis time to increase the expressiveness of TTS.

5. Conclusions

In this paper we have tried to make a case for applying machine learning and datamining techniques to concatenative-unit speech synthesis corpora in order to enhance the expressiveness of TTS. Although recording a large database for every desired expressive style can be very effective, it can also be costly in terms of recording time, voice talent fees, and system footprint size. The premise of this work lies in recognizing that there is often noticeable expressive variability to be found within large databases which can be exploited as an alternative to enlarging existing expression-specific corpora. The approach is of course limited by the extent to which the expression can be at all found in a baseline, mostly neutral, database: some speakers may exhibit less expressive variability, especially when they may have been coached to speak with a consistent style for the purpose of speech synthesis. Or the limitation may be one of degree: a given expression may be found, but in a much mitigated form.

To apply these ideas we focused on the identification and realization of perceptual emphasis, something which we expected to find with some likelihood in a large database of approximately 10 hours. We described a system that can be automatically trained to identify with reasonably high performance the occurrence of emphasized words throughout the database, and then demonstrated that augmenting the corpus with these automatically discovered labels significantly enhances the perception of intended emphasis.

6. Acknowledgements

The authors wish to thank Andy Aaron for useful discussions and feedback leading to the design of the perceptual tests, Andy and Larry Sansone for their help in running these tests, and Michael Picheny for useful discussions and motivation for this research. The authors also wish acknowledge and thank the support of the TC-STAR project (Technology and Corpora for Speech to Speech Translation; <http://www.tc-star.org/ssw6/>.) [2], a long-term effort financed by the European Commission within the Sixth Framework Program to advance research in speech-to-speech translation technologies.

7. References

[1] J. F. Pitrelli, R. Bakis, E. M. Eide, R. Fernandez, W. Hamza, and M. A. Picheny, “The IBM expressive Text-to-Speech synthesis system for American English,” *IEEE Trans. Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1099–1108, July 2006.

[2] E. Eide and R. Fernandez, “Database mining for flexible concatenative Text-to-Speech,” in *Proc. ICASSP*, vol. 4, Honolulu, Hawai'i, April 2007, pp. 697–700.

[3] M. Bulut, W. Narayanan, and A. Syrdal, “Expressive speech synthesis using a concatenative synthesizer,” in *Proc. ICSLP*, Denver, CO, U.S.A., 2002.

[4] N. Campbell and T. Marumoto, “Automatic labelling of voice-quality in speech databases for synthesis,” in *Proc. ICSLP*, vol. 4, Beijing, China, October 2000, pp. 468–471.

[5] E. O. Selkirk, *Phonology and Syntax: The Relation Between Sound and Structure*. Cambridge: MIT Press, 1984.

[6] F. Tamburini, “Automatic prosodic prominence detection in speech using acoustic features: An unsupervised system,” in *Proc. Eurospeech*, vol. 1, Geneva, Switzerland, September 2003, pp. 129–132.

[7] J. Pitrelli, “Expressive Speech Synthesis using American English ToBI: Questions and contrastive emphasis,” in *Proc. IEEE ASRU: Automatic Speech Recognition and Understanding Workshop*, St. Thomas, U.S. Virgin Islands, December 1-4 2003.

[8] R. Fernandez, “A computational model for the automatic recognition of affect in speech,” Ph.D. dissertation, (http://affect.media.mit.edu/pdfs/04.fernandez-phd.pdf) Media Arts and Sciences. Massachussets Institute of Technology, 2004.

[1] Statistical significance is assessed in a Bayesian fasion by treating the “identification rate” as a random variable $x \in [0, 1]$ with a parametric distribution $p(x|k, n) \propto x^k(1-x)^{n-k}$, where k is the number of times a stimulus is identified in a population of size n. Significance is then evaluated as the p value which satisfies the inequality $p(x_1 < x_2|k_1, n_1, k_2, n_2)$ for the two events examined. See Appendix D in [8] for mathematical details.

[2] Project No. FP6-506738

Modeling and Perceiving of (Un)Certainty in Articulatory Speech Synthesis

Charlotte Wollermann[1], *Eva Lasarcyk*[2]

[1]Institute of Communication Sciences, University of Bonn, Germany
[2]Institute of Phonetics, Saarland University, Germany
cwo@ifk.uni-bonn.de, evaly@coli.uni-sb.de

Abstract

This paper deals with the role of paralinguistic expression in articulatory speech synthesis. We describe two experiments which investigate the perception of certain vs. uncertain utterances produced by articulatory speech synthesis, using the system developed in [1].

Experiment 1 tests to what extent subjects are able to identify certainty and uncertainty as intended paralinguistic expressions in the acoustical signal by the varying acoustic cues intonation and delay. Further on, we investigate if (un)certainty influences the intelligibility of the synthetic utterances. Results show that the utterances are identified as intended with respect to (un)certainty. Regarding intelligibility, hardly any influence is measurable.

Experiment 2 looks more in detail into the perception of uncertainty by using several levels. Therefore, not only intonation and delay are varied as acoustical cues but also fillers. Results show that our intended different levels of uncertainty indeed evoked different degrees of perceived uncertainty.

1. Introduction

The role of emotion and attitude in human-machine interaction has gained extensive importance in the last few years. One interesting question in this context is to what extent machines are able to recognize emotions in spoken dialogs (e.g. [2], [3], [4]). A typical scenario would be the interaction between a user and a spoken dialog system. Here, emotion detection of the user can be used in order to modify the dialog (cf. [5]). For instance in the case of user frustration or annoyance, the system could react adequately.

On the speech synthesis side, the modeling of emotion and attitude has gained more and more importance as one aims to generate synthetic speech which is as natural and human-like as possible. Emotional TTS systems have been developed by [6] and [7] among others. Most of the emotional TTS systems are based on the prototypical emotions *happiness*, *sadness*, *anger*, *fear*, *surprise* and *disgust* according to [8]. Beyond that, the interface EmoSpeak as part of the TTS System MARY ([7], [9]) uses *evaluation*, *activation* and *power* as basic dimensions for representing emotional states[1]. Thus it is possible to express "... gradual emotional states in a more flexible way than has previously been possible." ([7]: Preface).

Current emotional TTS systems are based on different techniques. One technique, which has become more and more popular, is *Unit Selection* ([11], [12]). With this technique, a reduced set of units is processed from a large speech corpus for concatenative synthesis (cf. [13]: 279). The advantage of this technique can be described as follows: "The synthesis often perceived as being most natural is unit selection, or large database synthesis, or speech re-sequencing synthesis." ([7]: 91). But the big drawback is that the technique shows a deficit when there are no appropriate units in the synthesizer (cf. [23]: 4).

The approach chosen for this study is articulatory speech synthesis. The 3D-articulatory synthesizer used here [1] allows for a great degree of freedom and precise adjustment of single parameters at the same time. It is not limited by any set of prerecorded utterances. Thus it might be suitable for emotional synthesis, which, by nature, can be very rich in variations. Nevertheless, the modeling of emotion and attitude in articulatory speech synthesis has been barely investigated. There are, however, some recent projects investigating the synthesis of laughter [15] or voice quality variation [16] with articulatory speech synthesis, which let us catch a glimpse of the continuum of possible manipulations with this kind of synthesis.

1.1. Production and perception of (un)certainty in natural speech

In order to simulate emotional ways of speaking or to convey paralinguistic expressions in synthetic speech, it is firstly necessary to know how emotional or paralinguistic cues are produced and perceived in *natural* speech. In the following, we will give an overview of selected studies that deal with the role of (un)certainty in human-human dialog.

The work of [17] serves as source of inspiration for many studies on this field (e.g. [18], [19]). The authors investigated memory processes in question answering situations. Question-answering in their framework is regarded as a social process, which is characterized by information exchange and also self-presentation (cf. [19]). For testing the hypothesis that uncertainty of a speaker is marked differently than certainty, they use Hart's [20] so-called *Feeling of Knowing (FOK)* paradigm. With this method, it is possible to elicit meta-memory judgments. Their experimental investigation brought to light that uncertainty is not only signaled by using linguistic hedges like "I guess", but also by prosodic features like rising intonation and delay (cf. [19]).

In order to investigate how people perceive the *FOK* of another speaker, [18] defined the *Feeling of Another's Knowing (FOAK)*. Their study showed that the *FOAK* "... was affected by the intonation of answers, the form of answers ...

[1] The idea that emotional experience can be presented by using gradual dimensions goes back to [10]. An overview of the different models and names for the dimensions is given in [7].

the latency to response, and the presence of fillers." ([18]: 396). The term *filler* is defined as "interjections such as 'um', 'uh', 'hmm'" ([18]: 383).

As the studies mentioned focused on the role of (un)certainty in the *acoustical* signal, the question remained open about which role (un)certainty plays in the *audiovisual* modality. With respect to production, Swerts, Krahmer and colleagues ([21], [19]) found that several characteristics are used for the production of uncertainty: *delay*, *pause* and *fillers* for the audio modality; *smiles*, *"funny faces"* etc. for the visual one. Tests on the perception side showed that subjects were able to distinguish certain from uncertain utterances for all three conditions (audio-only, visual-only, and audiovisual); the identification was even easier in the bimodal condition than in the unimodal conditions (cf. [19]).

1.2. Characteristics and goal of the current study

The current study deals with the modeling and perception of different degrees of certainty in articulatory speech synthesis. The stimuli are characterized by a high/low degree of certainty (experiment 1) as well as by several more fine-grained degrees of uncertainty (experiment 2).

For generating our stimuli, we use the articulatory speech synthesis system developed in [1]. It offers a high degree of speech quality combined with a very high degree of high level control over all articulatory parameters (cf. Sec. 2.2).

The goal of experiment 1 is to investigate if subjects are able to distinguish intended *certain* utterances from intended *uncertain* ones under the audio condition. Another purpose is to survey if certainty influences the intelligibility of the synthetic utterances.

The purpose of experiment 2 is to look into the perception of (un)certainty in articulatory synthesis more in detail by using several degrees of uncertainty. Thus, it should be determined which acoustical cues exactly are relevant for perceiving an utterance as *uncertain*.

2. Modeling (un)certainty in articulatory speech synthesis

2.1. Acoustical criteria for modeling (un)certainty

According to [21], uncertainty will be distinguished from certainty acoustically along the dimensions *delay* (presence or absence), *intonation* (high or low) and, later on (in experiment 2), also *fillers* (presence or absence), as shown in Table 1.

Delay values are 1000 ms in an unmarked question-response case (for the *certain* stimuli) (cf. [17], who report an average silence of 0.97 s), and 2200 ms for a "delayed" answer in uncertain stimuli without fillers. The *uncertain* stimuli that do contain fillers (we chose the sound "hmmm") have a delay structure of 1500 ms before the filler and another 1000 ms after the filler, before the actual two-word answer starts.

Variation of the F0 contour takes place at the end of a stimulus, basically on the last word. It is characterized either by a rising F0 contour (*high intonation*, according to [21]) or a falling F0 contour (*low intonation*).

Table 1: Acoustical criteria for (un)certainty according to [21].

Acoustic cue	Certain	Uncertain
Delay	-	+
High intonation	-	+
Low intonation	+	-
Filler	-	+

2.2. The articulatory speech synthesis system

As described above, the test stimuli are generated with the speech synthesizer in [1]. This synthesizer uses a three-dimensional model of the vocal tract (see Fig. 1). Based on its geometry, an aerodynamic-acoustic simulation generates the speech output. The shape of this geometrical model is controlled by a *gestural score* on which the pertinent parameters are varied according to the intended articulation.

The movements of the supraglottal articulators (such as lips, tongue, jaw, velum) can be subsumed under vocalic, consonantal, and velar gestures. Their basic interaction is held constant to convey the phonemic (*linguistic*) content of the utterance, i.e. the words. On top of that, *paralinguistic* features are changed according to the degree of certainty aimed at. In this way, the F0 movements are specified on an F0 tier, matching the desired intonational patterns.

Fillers can be inserted when needed by simply adding the corresponding gestures in the score.

The variations in response delay are accounted for during preparation of the *complete* stimulus as a question-answer pair (see next section below).

In addition to the audio part, the system is characterized by a three-dimensional visualization of the articulatory gestures (see Fig. 1). The lips can be seen together with the front teeth and parts of the tongue. Other facial parts such as eyes or eyebrows are not displayed.

As our goal for the current study is to have a *first* look into the perception of (un)certainty in articulatory speech synthesis, we will focus in our experiments on the pure audio condition. Future work is also going to consider the visual part.

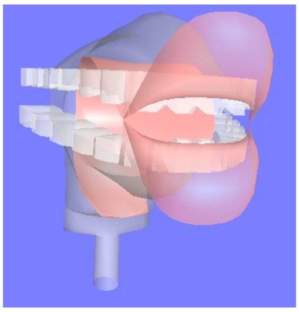


Figure 1: 3D model of the vocal tract of the articulatory speech synthesizer [1].

Table 2: Features of the dialogs presented in experiment 1 (with the starred IDs, 6 in total), and experiment 2 (ID 1 - 4; 7 - 10). Levels of certainty: C: *certain*, U1: *uncertain 1*, U2: *uncertain 2*, U3: *uncertain 3*. For further explanations cf. Sec. 2.

ID	Caller's question	System's answer	Level of certainty	Intonation high	Delay	Filler
1*	“Wie wird das Wetter nächste Woche in X?“	“Ziemlich kühl“	C	-	-	-
2			U1	+	-	-
3*			U2	+	+	-
4			U3	+	+	+
5*		“Relativ heiss“	C	-	-	-
6*			U(2)	+	+	-
7*		“Eher kalt“	C	-	-	-
8			U1	+	-	-
9*			U2	+	+	-
10			U3	+	+	+

2.3. Scenario

For embedding the stimuli in a context, we chose the interaction between a caller and a telephone weather expert system. The caller asks the question: “Wie wird das Wetter nächste Woche in X?” (*How is the weather going to be next week in X?*) and the program gives an answer. Since this study is meant as an initial investigation, the answers are very short (two-word sentences) and there are only three different wordings: “ziemlich kühl” (*pretty chilly)*, “relativ heiss” (*relatively hot*) and “eher kalt” (*rather cold*).

For experiment 1, each wording is generated in two versions: in a *certain* and an *uncertain* way of speaking. All in all there are six dialogs. They are shown with a starred ID in Tab. 2.

For experiment 2, we leave out the wording “relativ heiss” due to the low intelligibility[2] measured in experiment 1 (cf. Sec. 3.1.3). The other two wordings (“ziemlich kühl”, “eher kalt”) are generated in four versions: One *certain* way of speaking, and three *uncertain* ones. They are intended to capture different degrees or different acoustic aspects of uncertainty. The final intonation is always high but there are differences regarding delay times and fillers. The presumably weakest version concerning the level of uncertainty (*uncertain 1*) has no more marked features (only high intonation), a middle version (*uncertain 2*) possesses the delay structure mentioned in Sec. 2.1 and high intonation, and the strongest version (*uncertain 3*) incorporates all acoustic signals of uncertainty concentrated on in this paper (i.e. intonation, delay, and filler). Altogether, there are eight relevant dialogs (see Tab. 2, IDs 1 - 4 and 7 - 10).

Two additional wordings are generated: “Trocken” (*dry*) and “heiss” (*hot*). These wordings are embedded in different contexts and the resulting dialogs serve as filler items[3]. In addition, “trocken” embedded in another dialog is used as an example for making subjects familiar with the stimuli. In these cases, the level of certainty of the wordings is intended to be neutral. We define *neutrality* as an unmarked version between *certain* and *uncertain*: The acoustic features show regular delay, regularly low intonation (not as deep as in the certain version), and contain no fillers.

3. Perception studies

3.1. Experiment 1

3.1.1. Goal

The goal of experiment 1 is to determine if subjects are able to recognize certainty and uncertainty as intended paralinguistic expressions in articulatory speech synthesis under audio condition. Another purpose is to investigate if certainty affects the intelligibility of articulatory synthetic utterances.

3.1.2. Method

Subjects were 38 students of the Universities of Bonn and Saarbrücken with an average age of about 25.5 years. 18 participants were female, 20 male, all of them German native speakers. They were tested in group experiments or individually: The audio stimuli were presented to them in two different random orders[4] over a loudspeaker. When the example stimulus was presented to the subjects, they had the chance to ask questions. After the procedure started, subjects were neither supposed to ask any questions nor was any feedback given. For each dialog between the caller and the weather expert system, the subjects were asked to score the answer of the system regarding its certainty and also its intelligibility on a 5-point Likert-Scale with 1 meaning *uncertain/unintelligible* and 5 meaning *certain/intelligible*, respectively.

The results were statistically analyzed using the Wilcoxon Signed Rank Test. This test was chosen since our dependent data were measured on an ordinal scale. The ratings of the stimuli were compared in pairs to test if there were significant differences in rating the intended *uncertain* and *certain* utterances. The null hypothesis (H_0) was as follows: There is no dependency between the rating of the utterances as *certain*/*intelligible* and their intended certainty and uncertainty respectively. The alternative hypothesis (H_1) was: The rating of the utterances as *certain*/*intelligible* depends on their intended certainty and uncertainty respectively. The level of significance was 5 %.

3.1.3. Results

Results for the perception of *certain* and *uncertain* utterances regarding their certainty are visualized in Fig. 2 and Tab. 3. The intended *certain* versions of “ziemlich kühl”, “relativ heiss”, and “eher kalt” were all rated with a median of 4, whereas the median for the *uncertain* versions received a lower median of 3. The comparison of both data series showed a highly significant difference for each wording

[2] Technical problems with the initial consonant in “relativ” seemed to be the reason for this.

[3] *Filler items* in this case are items which are not intended to be relevant to this experiment. Thus, the ratings for the example dialog and also for the filler items will not be considered in the analyses.

[4] In order to minimize the influence of the sequence of the stimuli when calculating the overall results.

("ziemlich kühl": V = 342.5; p < 0.001, "relativ heiss": V = 308.5; p < 0.001, "eher kalt": V = 351; p < 0.001).

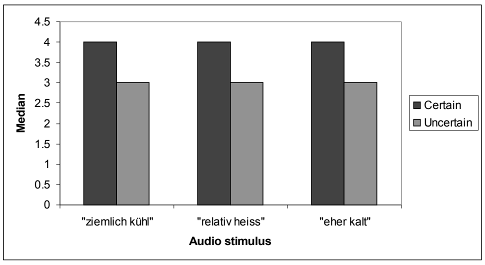


Figure 2: Medians for perceiving certainty of the three wordings in the *certain* and the *uncertain* way of speaking (experiment 1).

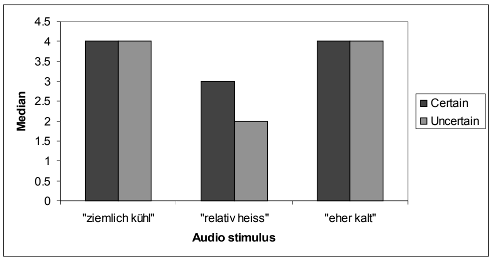


Figure 3: Medians for perceiving intelligibility of the three wordings in the *certain* and the *uncertain* way of speaking (experiment 1).

Table 3: V and *p* values of the pairwise comparison of *certain* vs. uncertain utterances; significant differences are marked in bold face.

Wording	Certainty	Intelligibility
"Ziemlich kühl"	**342.5; < 0.001**	56.5; 0.63
"Relativ heiss"	**308.5; < 0.001**	**123; 0.04**
"Eher kalt"	**351; < 0.001**	79.5; 0.14

The results for the intelligibility of (un)certain utterances are shown in Fig. 3 and Tab. 3. Subjects ranked both the *certain* and the *uncertain* versions of "ziemlich kühl" and "eher kalt" with a median of 4. In both wordings, there was no significant difference between these two data series ("ziemlich kühl": V = 56.5; p = 0.63, "eher kalt": V = 79.5; p = 0.14).

Further on, the median value for "relativ heiss" in a *certain* way of speaking was 3, whereas the *uncertain* version had a lower median of 2. The statistical analysis resulted in a significant difference between the judgments (V = 123; p = 0.04).

When regarding the absolute ranking values for intelligibility, it became obvious that the wording "relativ heiss" was less intelligible than the other two wordings. Statistical testing showed that the *certain* version of "relativ heiss" was rated significantly less intelligible than the *certain* versions of the other wordings ("relativ heiss" vs. "ziemlich kühl": V = 398.5; p < 0.001, "relativ heiss" vs. "eher kalt": V = 514; p < 0.001). The intelligibility of the *uncertain* version of "relativ heiss" was also rated significantly lower than that of other two wordings ("relativ heiss" vs. "ziemlich kühl": V = 465; p < 0.001, "relativ heiss" vs. "eher kalt": V = 547; p < 0.001).

In summary, as the results of the perception of certainty indicate, subjects could clearly distinguish between the intended *certain* and *uncertain* utterances in all wordings.

Furthermore, intelligibility was only very weakly influenced by the intended certainty and uncertainty, respectively.

3.1.4. Discussion

The relatively low ranking of the intelligibility of "relativ heiss" might come from the fact that some of the phones used in this utterance seemed to be hard to understand (presumably the /r/ of "relativ"), because they presented some technical problems during the speech generation process. Therefore, only the wordings "ziemlich kühl" and "eher kalt" will be considered in experiment 2.

Since experiment 1 is meant to be an initial investigation and therefore focuses on the perception of intonation and delay, the question remains open which role *fillers* play in perceiving articulatory speech synthesis. This leads to the setup of a second experiment in which the stimuli cover *more* acoustic aspects of uncertainty by defining different degrees of uncertainty to get more detailed results.

3.2. Experiment 2

3.2.1.Goal

The goal of experiment 2 is to determine if there is a ranking regarding the impact of the different cues signaling uncertainty. Thus, subjects are tested to find out to what extent different combinations of acoustic cues affect the perception of uncertainty.

3.2.2. Method

The same method was applied as in experiment 1. Subjects were 34 seminar students[5] (23 females, 11 males, average age of 23 years), tested within three group experiments, each one having a different random order of stimuli. After listening to one example dialog, subjects were presented 10 test dialogs[6]. For each answer of the expert system, they were asked to evaluate the certainty on a 5-point Likert scale with 1 meaning *uncertain* and 5 meaning *certain*.

The results were again analyzed using the Wilcoxon Signed Rank test. Like in experiment 1, the ratings of the stimuli were compared pairwise to test if there were significant differences in rating the intended certain utterances and uncertain ones. However, now there were three different levels of uncertainty. The null hypothesis (H_0) was as follows: There is no dependency between the rating of the utterances as *certain* and *uncertain,* respectively, and their particular level of certainty. The alternative hypothesis (H_1) was: The rating of the utterances as *certain* and *uncertain,* respectively,

[5] Subjects were different from those of experiment 1.

[6] Including the two filler items described in Sec. 2.3.

depends on their intended certainty and level of uncertainty, respectively. The level of significance was 5 %.

3.2.3. Results

The results for the perception of “ziemlich kühl” and also of “eher kalt” are displayed in Fig. 4 and Tab. 5. The certain version of “ziemlich kühl” was rated with a median of 4 as most certain compared to the uncertain versions. The comparison of the data series between *certain* and *uncertain 1* (median = 3) was statistically significant (V = 162; p < 0.01). In a similar way, *uncertain 2*, compared to *certain*, achieved a median value of 3 in a statistically highly significant way (V = 251; p < 0.001). *Uncertain 3* was rated lowest with a median of 2. The difference between *certain* and *uncertain 3* was highly significant (V = 519; p < 0.001). The graph also shows that the ratings for “ziemlich kühl” were very similiar with 3.5 and 3 for *uncertain 1* and *uncertain 2*. The statistical analysis showed no significant difference (V = 64.5; p = 0.11). In contrast, the difference between *uncertain 1* (with a median of 3) and *uncertain 3* was highly significant (V = 504.5; p < 0.001), as well as the one between *uncertain 2* and *uncertain 3* (V = 369.5; p < 0.001).

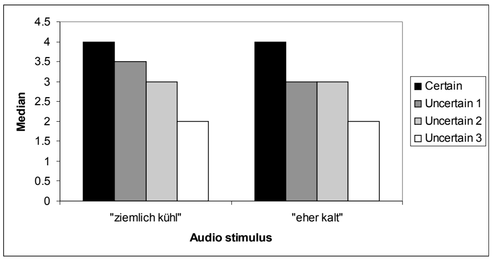


Figure 4: Medians for perceiving certainty of two wordings in one *certain* and three *uncertain* ways of speaking (experiment 2).

Table 5: Results of the pairwise comparisons of *certain* (C) vs. different types of *uncertain* utterances (U1,2,3) in experiment 2. Significant differences are marked in bold face.

Wording	Levels of certainty compared	*V* value; *p* value
“Ziemlich kühl“	**C vs. U1**	**162; < 0.01**
	C vs. U2	**251; < 0.001**
	C vs. U3	**519; < 0.001**
	U1 vs. U2	64.5; 0.11
	U1 vs. U3	**504.5; < 0.001**
	U2 vs. U3	**369.5; < 0.001**
“Eher kalt“	**C vs. U1**	**268; < 0.001**
	C vs. U2	**210; < 0.001**
	C vs. U3	**528; < 0.001**
	U1 vs. U2	70.5; 0.32
	U1 vs. U3	**397.5; < 0.001**
	U2 vs. U3	**319; < 0.001**

“Eher kalt” in a *certain* version of speaking was judged with a median of 4, whereas the median of *uncertain 1* was lower with a median of 3. The difference between the two data series was highly significant (V = 268; p < 0.001). The lower ranking of *uncertain 2* (median = 3) in comparison with the one for *certain* was also statistically significant (V = 210, p < 0.001). Furthermore, *uncertain 3* obtained a much lower median with 2 than *certain*. The statistical analysis resulted in a highly significant difference (V = 528; p < 0.001).

Additionally, the analysis showed that the rankings for *uncertain 1* differed not significantly from those for *uncertain 2* (V = 70.5; p = 0.32): the median value was 3 each time. In contrast to that, the judgments for *uncertain 1* and those for *uncertain 3* yielded a highly significant difference (V = 397.5; p < 0.001). Along these lines, the rankings for *uncertain 2* also differed in a highly significant way from those for *uncertain 3* (V = 319; p < 0.001).

In summing up, the results indicate that each of the intended *uncertain* versions (of all levels) were clearly perceived as being more uncertain than the *certain* versions for both wordings.

Within the set of uncertain stimuli for each wording, *uncertain 3* was judged significantly less certain than the other two levels of certainty.

However, in both of the wordings, there was no significant difference in evaluating the degree of certainty of *uncertain 1* vs. *uncertain 2*.

3.2.4. Discussion

First of all, the results of experiment 2 generally confirm the ones of experiment 1, in that intended certain utterances can be clearly distinguished from uncertain ones. While in experiment 1 there was only one level of uncertainty, conveyed by high intonation and delay, our more fine-grained analysis in experiment 2 showed more detailed results. Even if uncertainty is signaled only by high intonation, this is sufficient to be perceived as *uncertain.* The role of delay and fillers exclusively cannot be inferred from our data due to the design of our set of stimuli. It can only be said that, firstly, delay as an additional acoustic cue to high intonation does not yield a higher degree of perceived uncertainty. Secondly, our data suggest that the combination of fillers, delay, and high intonation have the strongest effect on the perception of uncertainty. However, from our data it is not clear how far this strongest effect is *purely* due to fillers.

4. Conclusions

The experiments presented in this paper present a first step towards the modeling of certainty and different degrees of uncertainty with the means of articulatory speech synthesis. Previous studies identified acoustical cues such as intonation, delay, and fillers in human-human dialog that convey uncertainty. Our study focused on the role of these cues in human-machine interaction.

The experiments brought to light that intonation by itself does contribute to the perception of uncertainty in articulatory speech synthesis in our test data. This is also true for the combination of all three cues. Further experiments are necessary, though, to determine how far the perception of uncertainty is purely influenced by fillers and delay, respectively. In contrast to previous studies, our data might suggest that delay by itself does not contribute to a stronger perception of uncertainty. One should take into account, though, that listeners are presumably less sensitive to delays in our context since they expect from a machine that the response time is not as quick as from a human being. Future work could also consider the set of problems which are linked with the judgment of a machine's meta-cognitive state.

It can be well argued that the choice of the wordings (e.g. “ziemlich” as an adverb denoting vagueness) could also convey different levels of certainty in themselves. When further investigating *para*linguistic features conveying uncertainty, it seems useful to choose lexically more neutral wordings.

It would be interesting to run a cross-technique evaluation, since so far our cues only covered the purely acoustic domain – a domain that other kinds of synthesis could also cover.

As there is much evidence in the literature that prosody is not only conveyed by the acoustical channel but also by the visual one (e.g. [22]), we are planning to test unimodal and bimodal stimuli for several levels of certainty, finally making use of the three-dimensional vocal tract provided by the articulatory synthesizer.

5. Acknowledgements

We would like to thank Bernhard Schröder and Jürgen Trouvain for helpful comments. We are also grateful to Wolfgang Hess, Petra Wagner, Bernhard Fisseni, Stefan Breuer, Caja Thimm, Ulrich Schade, and Jürgen Trouvain for helping us with the experimental setup. Our thanks go, last but not least, to Peter Birkholz for giving us the chance to use the articulatory speech synthesis system, and to our speaker Tobias Ebel.

6. References

[1] Birkholz, P. (2005). *3-D Artikulatorische Sprachsynthese*. Berlin: Logos Verlag.

[2] Ang, J., Dhillon, R., Krupski, A., Shribers, E., Stolcke, A. (2002). “Prosody-based automatic detection of annoyance and frustration in human-computer dialog”. In *Proceedings of ICSLP*, vol. 3, 2037-2040.

[3] Lee, M. and Narayanan, S. (2005). “Towards detecting emotions in spoken dialogs”. In *IEEE Transactions on Speech and Audio Processing, 13* (2), 293-303.

[4] Litman, D. and Forbes-Riley, K. (2004). “Predicting Students Emotions in Computer-Human Tutoring dialogs”. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL),* 351-358, Barcelona, Spain.

[5] Ai, H. (2006). “Position paper”. In *Online Proceedings of the Second Annual Young Researchers Roundtable on Spoken Dialogue Systems*. Pittsburg, PA. http://people.csail.mit.edu/alexgru/yrrsds/proceedings/yrrsds_proceedings06.pdf

[6] Burkhardt, F. (2001). *Simulation emotionaler Sprechweise mit Sprachsyntheseverfahren*. PhD Thesis, University of Berlin. Shaker Verlag.

[7] Schröder, M. (2004). *Speech and Emotion Research: An overview of research frameworks and dimensional approach to emotional speech synthesis*. PhD thesis, PHONUS 7, Research Report of the Institute of Phonetics, Saarland University.

[8] Ekman, P. (1972). “Universals and cultural differences in facial expressions of emotion”. In Cole, J. (ed.), *Nebraska Symposium on Motivation 1971*, vol. 19, 207-283. Lincoln, NE: University of Nebraska Press.

[9] Schröder, M. and Trouvain, J. (2003). “The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching”. In *International Journal of Speech Technology, 6,* 365-377.

[10] Wundt, W. (1896). *Grundriss der Psychologie*. Leipzig: Verlag von Wilhelm Engelmann.

[11] Sagisaka, Y. (1988). “Speech synthesis by rule using an optimal selection of non-uniform synthesis units”. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 679- 682.

[12] Black, A., Campbell, N. (1995). “Optimising Selection of Units from Speech Database for Concatenative Synthesis”. In *Proceedings of Eurospeech*, *Vol 1*, Madrid, 581-584.

[13] Campbell, N., Black, A. (1996). “Prosody and the Selection of Source Units for Concatenative Synthesis”. In Santen, J. van, Sproat, R., Olive, J., Hirschberg, J. (eds), *Progress in speech synthesis,* 279-282, Springer Verlag.

[14] Beskow, J. (2003). *Talking Heads – Models and Applications for Multimodal Speech Synthesis*. Doctoral Dissertation, KTH, Stockholm, Sweden.

[15] Lasarcyk, E. and Trouvain, J. (to appear). “Imitating conversational laughter with an articulatory speech synthesizer.” To appear in *Proceedings of the Interdisciplinary Workshop on the Phonetics of Laughter*, Saarbrücken, August 2007.

[16] Lasarcyk, E. (to appear). “Investigating Larynx Height With An Articulatory Speech Synthesizer”. To appear in *Proceedings of the 16th ICPhS*, Saarbrücken, August 2007.

[17] Smith, V. and Clark, H. (1993). “On the course of answering questions”. In: Journal of Memory and Language, 32, 25-38.

[18] Brennan, S. E. and Williams, M. (1995). “The feeling of another's knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers”. In *Journal of Memory and Language* , 34, 383-398.

[19] Swerts, M. and Krahmer, E. (2005). ”Audiovisual prosody and feeling of knowing”. In *Journal of Memory and Language*, *53:1*, 81-94.

[20] Hart, J.T. (1965). “Memory and the feeling-of-knowing experience”. In *Journal of Educational Psychology*, 56, 208–216.

[21] Swerts, M., Krahmer, E., Barkhuysen, P. & van de Laar, L. (2003). “Audiovisual cues to uncertainty”. In: *Proceedings of ISCA workshop on error handling in spoken dialog systems,* Chateau-d'Oex, Switzerland, August/September 2003.

[22] Massaro, D.W. (1998). “Perceiving Talking Faces: *From speech perception to a behavioral principle*. Cambridge, Massachusetts: MIT Press.

[23] Beskow, J. (2003). *Talking Heads – Models and Applications for Multimodal Speech Synthesis*. Doctoral Dissertation, KTH, Stockholm, Sweden.

Perceptual Annotation of Expressive Speech

Lijuan Wang[1]*, Min Chu*[1]*, Yaya Peng*[2]*, Yong Zhao*[1]*, Frank Soong*[1]

[1]Microsoft Research Asia, Beijing, China
[2]Department of Linguistics & Modern Languages, The Chinese University of Hong Kong, China
[1]{lijuanw;minchu;yzhao;frankkps}@microsoft.com

Abstract

A six-dimensioned label set for annotating expressiveness of speech samples is proposed. Unlike conventional emotional annotation labels that require annotators to make rather difficult judgments on speakers' emotional (high-level) status, the new annotation set of six low-level labels, i.e., "pitch", "vocal effort", "voice age", "loudness", "speaking rate", and "speaking manner" can be more easily labeled by non-experts. 800 expressive utterances were annotated by four annotators with the proposed labels. The labeling also shows a good consistency (71%) among the annotators. The proposed six labels capture the different styles (expressiveness) well in the audio-book. The difference between styles, measured by the intensity of styles along the six labels, is highly correlated (0.85) with the perceptual distance obtained from a subjective AB test. A compact classification and regression tree (CART) is built to automatically group sentences of similar expressiveness into several "pure" speaking styles. The interpretation of each speaking style can be explicitly understood from the CART structure.

1. Introduction

Synthesis of speech with rich expression has become an active research topic recently, driven by the technology advances and market demands. In contrast with the neutral speech, expressive speech normally carries richer para-lingual and extra-lingual information that gives listeners clues on the emotional status, attitude, or intention of a speaker [1] [2]. To synthesize expressive speech, a better understanding of how people convey and perceive expressions in speech signals becomes necessary, which requires annotating and analyzing speech expressiveness. However, so far, no universal methodology for annotating expressive speech has been widely accepted. In this paper, we studied how to annotate the speaking styles in expressive speech. In addition, we showed how to classify the underlying expressive styles based on the small amount of annotated speech data, which may help to annotate the expressive styles of the whole corpus automatically.

The first step in studying speech expressiveness is collecting a speech corpus that embeds speech of the target emotions or expression. The typical ways for such data collections include the followings [1]:

1. Hiring professional actors to act with specified expressions;
2. Inducing a subject to speak in a pre-designed style or emotion by providing appropriate stimulus;
3. Selecting emotional speech segments from a large conversational speech database.

In the first two ways, the expressions collected tend to be exaggerated. In the third way, the expression quality cannot be guaranteed and some expected styles may not exist in the speech database.

In this paper, we explored the expressive speech in an audio-book of a fiction narrated by a professional voice talent. Not only is the speech recorded with high quality, but also the expressions throughout the audio-book sound natural. Another unique feature of the audio-book is that all the characters are mimicked by the same speaker. To distinguish the multiple characters from each other, the speaker did his best to control his voice to establish various styles. It hence has a good coverage of the speaker's expressive space and serves as a good source for studying.

Once the expressive speech corpus is obtained, the second step is to annotate the various speaking styles in the corpus. However, it is difficult to annotate the speaking style directly, because 1) the number of the different speaking styles is unknown; 2) no definition for the speaking styles in the corpus is available. We decide to annotate the speech from multiple perspectives, and then the definition of its speaking style is derived as a result. So far, various annotation sets have been proposed to annotate emotion/expressiveness. Generally, they can be classified into two groups: categorical and dimensional. In dimensional group [3], three "emotion dimensions", i.e. activation, valuation and power, have been suggested to describe the gradient nature of expressive speech. In categorical group, primary emotions such as the "big six" emotions have been widely used [3]. Currently, there is no universal annotation set for labeling expressive speech. Previous studies show that specific annotation is needed to handle the special features of the target corpus. Hence, we designed specific annotation set for the corpus of audio books, so that 1) the difference in the expressive styles can be accurately discriminated; 2) high consistency among different annotators can be achieved; 3) the annotation method is simple, reliable, and robust for non-experts.

In our audio-book corpus, we found that the conventional annotation sets cannot fully discriminate the different characters. Therefore, we proposed 6 perceptually discernible labels -- "pitch", "vocal effort", "voice age", "loudness", "speaking rate", and "speaking manner". Unlike the conventional annotation sets that require annotators to summarize high-level emotions from the perceived acoustic clues, our set only requires the annotators to distinguish the levels of perceptually discernible acoustic information. We believe such an annotation set eases labeling and provides a more comprehensive description of the para-lingual information of an utterance. With the new set, about 800 utterances sampled from the audio-book are annotated subjectively. The reliability and effectiveness of the annotation results will be discussed. The annotated 800 sentences are used to train a Classification and Regression Tree (CART) to group the expressive speech into several "pure" expressive styles.

The rest of the paper is organized as follows. Section 2 introduces the expressive speech corpus and the annotation preparation. Section 3 describes the proposed annotation set and the subjective annotation methodology. Section 4 analyzes the annotation results in terms of inter-annotator agreement and discriminating ability of the six labels. Section 5 shows how to use the annotated data to classify the underlying speaking styles. Section 6 gives the conclusion and the discussion of future research.

2. Data preparation

2.1. Expressive speech corpus

The corpus used in this study is from a fiction audio-book narrated by a professional voice talent. Not only is the obtained speech of high-quality, but also the expressions in the full context of the audio-book sound natural. Containing multiple distinctive characters mimicked by one speaker is a unique feature of the audio-books in speech expressiveness. Since speech is the only means to present the whole story, the voice talent tries his best to perform different characters or the same character in different conditions by changing his sound. During the recording, the speaker matches the characters to his acquaintances, like neighbors and relatives, so that the speaking styles of the characters are distinctive. The audio-book thus has a good coverage of the voice talent's expressive space.

The speech data are first segmented into utterances automatically by forced aligning them with the corresponding scripts. Then, the utterances of different characters are separated from the narration according to the location of the quotation marks in the text script. At last, the character ID is assigned to each utterance and is manually checked according to the text context. The whole corpus contains about 18,000 utterances of 50 characters including the narrator.

2.2. Previous study on the corpus

In our previous work [13], we measured the acoustic distance between the top 10 characters in the audio-book. We employ state-of-the-art speaker identification technologies to study the character speaking styles by assuming each character as an individual speaker. The identification procedure is as follows: A speaker independent Gaussian Mixture Model (GMM) [10], or Universal Background Model (UBM), is first trained from a corpus of multiple speakers. Then, each individual speaker's model is obtained by adapting the UBM with the utterances of the specific speaker in the Maximum A Priori (MAP) sense. The UBM is trained from all (18,000) utterances and each character model is adapted with 250 utterances of the character. During the adaptation process, only means are adjusted. MFCC's and fundamental frequencies and their delta coefficients are used as the acoustic features. Only voiced frames are used for model adaptation.

Once the character models are adapted, they are evaluated with a test set consisting of 50 utterances from each of the 10 characters. For each utterance in the test set, the acoustic likelihood measured against 10 character models and the UBM is calculated. The character whose model yields the highest likelihood is identified as the character. The character identification rate is 81.7%. Such a result indicates that, on one hand, there are significant differences between most characters, i.e. the characters in the corpus have distinctive speaking style; on the other hand, some characters are still confused with others. Therefore, it is necessary to have a more precise description of the speaking style or the expressiveness of an utterance.

2.3. Selection of data to be annotated

Annotating all the 18,000 utterances is time-consuming and costly. Therefore, in this study, we carefully selected 800 utterances, which is only 4% of the total utterances, to sample the speaker's expressive space. The 800 utterances are from 32 characters (denoted as C01~C32), with 5~40 utterances for each.

3. Annotating expressive speech

3.1. Design of annotation set

An ideal annotation set for labeling the corpus of audio books should be able to 1) discriminate various speaking styles in this corpus; 2) achieve high consistency among different annotators; 3) facilitate simple, reliable, and robust labeling by non-experts. Table 1 show some typical voice descriptions made by listening to some utterances. The listeners feel difficult to align the conventional emotion such as "happy", "sad" or "angry" to most speech utterance in the audio-book corpus, because those emotions are usually exaggerated to describe the subtlety of the expressiveness in natural speech. On the other hand, the listeners find that there is more information embedded in speech signals which cannot be well summarized by a single emotion label. Hence, we propose annotating the corpus from multiple perspectives, such as annotating the speech rate, the softness of speech, etc., which are easily grasped by the listeners without much subjective abstraction. Apparently, "voice age", "loudness", "speaking rate", and other perceptually discernible elements are good candidates in the annotation set.

Here, we borrow the framework of [2], where Campbell proposed a categorical annotation set of finer granule that annotates speaker state (e.g., confidence, emotion, mood, interest of speaker), speech style (purpose, sincerity, manner, mood, bias), and voice characteristics (energy, softness, brightness). Within this framework, we further take the inconsistency of human labeling, during both perception experiments and corpus annotation, into account. The result is a simplified, robust, and reliable annotation set, which only keeps those labels that can be easily, accurately, and consistently annotated by even non-experts.

Table 1: Description of voice portraits

Utterance example	Description
1	The voice comes from a middle-aged man. He speaks calmly in a narrative mode.
2	The voice comes from a young woman, who usually speaks quickly.
3	The voice comes from an old man, who speaks slowly and sometime stridently.

As shown in table 2, the proposed annotation set includes: “pitch”, “vocal effort”, “voice age”, “loudness”, “speaking rate”, and “speaking manner”. Each label has three perceptual levels. The detailed description of each annotation label is shown in table 3. The description uses simple and common words, so that non-experts can understand them clearly.

Table 2: Perceptual annotation set for expressive speech

	pitch	vocal effort	voice age	loudness	speaking rate	speaking manner
A	low	strong	teenager	loud	fast	conversational
B	normal	normal	adult	normal	normal	addressing
C	high	weak	elder	low	slow	narrative

Table 3: Specification for each element in annotation set

Annotation Set	Specification
pitch	Whether the voice’s pitch sounds high, or normal, or low?
vocal effort	Whether the voice sounds strong, or normal, or weak?
voice age	How old the voice sounds?
loudness	How loud the voice sounds?
speaking rate	How fast the speech sounds?
speaking manner	Whether the speaker is talking in front of a group of people, chatting or discussing with one or several persons, or speaking in a narrative mode?

3.2. Annotation procedure

Four experienced annotators, who listen to the utterances several times and indicate the perceptual level for each of the six features, label the utterances using the proposed annotation set. None of them had ever listened to this audio-book before. Therefore, they don’t have any pre-knowledge about the utterances.

Before the formal labeling, pre-experiment training was carried out in order to make the annotators familiar with the environment and to stabilize their decision standard. They were asked to study the annotation guideline and understand the connotations for every label and its options. During the training session, they were encouraged to compare their results with others, discuss different choices, and adjust the decision standard. Comparing the statistical results before and after a round of training, their agreement improved and tended to saturate after 3 rounds of training sessions, 3~4 hours each time.

After the training session, the formal labeling started. To avoid user-fatigue, 800 utterances were split into 16 sub-sessions, 50 utterances each. On average, it took an annotator 1.5 hours to finish one sub-session. The annotators completed the whole task in two weeks. After having the annotations, we performed the reliability and effectiveness study. The results are reported in the next section.

4. Annotation results analysis

This section shows the analysis of annotation results in terms of inter-annotator agreement and discriminating ability of the proposed annotation set.

4.1. Inter-annotator agreement

To check the consistency of the annotations from the four annotators, inter-annotator agreement is measured for each of the six annotation features (labels). We calculated the distribution of the ‘*four agreed*’ (high), ‘*three agreed*’ (medium) and ‘*two or less agreed*’ (low) samples on the 800 utterances along the 6 annotation features. As shown in Table 4, for each feature, more than two thirds of the utterances (64%~83%) achieve high or medium agreement. To get a more consistent annotated speech samples, we select a sub-set for the following studies, which only contains samples of high or medium agreement.

Table 4: Inter-annotator agreement of each annotation feature

Annotation feature	Agreement Distribution		
	High	Medium	Low
pitch	25%	39%	36%
vocal effort	21%	44%	35%
voice age	22%	47%	31%
loudness	24%	46%	30%
speaking rate	30%	45%	25%
speaking manner	53%	30%	17%
Avg.	29%	42%	29%

4.2. Intensity distribution on each annotation feature

To investigate the distribution of the annotated samples, we calculate the number of samples of different intensity on the six features. As table 5 shows, each level on the 6 annotation features has enough samples, at least 9% of the annotated utterances. Therefore, the data used in experiment is good sampling of the whole corpus.

Table 5: Intensity distribution on each annotation feature

Annotation feature	Intensity Distribution		
	A	B	C
pitch	58%	23%	19%
vocal effort	45%	42%	13%
voice age	9%	52%	39%
loudness	38%	48%	14%
speaking rate	15%	58%	27%
speaking manner	79%	12%	9%

4.3. Discriminating ability of each annotation feature

We assume one character has one expressive style. To find out how well the proposed six annotation features can discriminate different characters/styles, we calculate the *Mutual Information* (MI) [10] between the character group {Characters | Ci, i = 1, 2, …, 32} and the three-level values for each label {Levels | A, B, C} upon the l^{th} label, as shown in Eq. 1. Mutual information measures how knowing an utterance’s character ID reduces the uncertainty about its corresponding level on the l^{th} annotation feature. On the other hand, it also measures how much the knowledge of an utterance's annotation level on the l^{th} feature reduces the uncertainty about which character speaks the utterance. The MI for each annotation feature is given in Table 6.

$$MI_l = \sum_{y \in Characters} \sum_{x \in Levels} p(x,y) \log \frac{p(x,y)}{p(x) \cdot p(y)} \qquad (1)$$

As Table 6 shows, "pitch", "vocal effort", "voice age", and "loudness" are four most important annotation features in discriminating characters. Compared with these four features, "speaking rate" and "speaking manner" are less useful. This is because the "pitch", "vocal effort", "voice age", and "loudness" are more closely related to the physical voice characteristics of a character. While the "speaking rate" and "speaking manner" (of even the same character) may change due to the variation of the conversation environment and the conversation object.

Table 6: Mutual information on each annotation feature

Annotation feature	Mutual Information
pitch	0.41
vocal effort	0.31
voice age	0.42
loudness	0.33
speaking rate	0.24
speaking manner	0.22
Avg.	0.32

4.4. Discriminating ability of the annotation set

Annotating a corpus using the proposed 6 annotation features virtually discretizes a speaking style along the 6 dimensions. In other words, a style corresponds to a point in the six-dimensioned annotation feature space. If the annotation set is reasonably built, the distance between two styles in the six-dimensioned annotation feature space should indicate the perceptual difference between the two styles. Ideally, the larger the distance, the bigger the perceptual difference between two styles, and vice versa.

4.4.1. Perceptual difference between styles (characters)

By assuming one character has one speaking style, we designed an AB comparison test to measure the perceptual difference between characters, which was carried out by 20 subjects. In this test, utterances from the same character or different characters were paired and presented to the subjects. Subjects were asked to judge whether the two utterances were said by the same speaker or not (subjects didn't know that these utterances were in fact said by the same voice talent). 650 pairs of utterances were prepared for the experiment. Among them, 200 sentence pairs were inner-character comparison, 20 intra-character pairs for each character; 450 pairs are inter-character comparison, 10 pairs for between any two characters.

All utterance pairs were sorted randomly and separated into two sessions. Subjects were asked to finish the two sessions with a not-less-than 30-minute break in between. The utterance pairs were played to the subjects by a scoring tool on a standard PC and subjects listened to them through headphones. The sequence of stimuli played to each subject was randomly generated. Subjects were allowed to listen to each pair as many times as needed before making a final decision of "same speaker" or "different speakers". After the choice is made, next utterance pair will be presented. On average, it took a subject 3 hours to finish the experiment. Before the formal testing, a short training session was carried out. 20 Chinese graduate students, fluent in English speaking and with normal hearing, participated in the experiment. No one had ever listened to this audio-book before. Also they didn't know how many 'speakers' were involved in the test.

The style distance between two characters P and Q is defined by Eq. 2, i.e. the number of utterance pairs between P and Q that were judged as "different" over the total number of pairs between P and Q. Small perceptual distance means character P and Q are perceptually similar and vice versa.

$$Perceptual\ Difference(P,Q) = \frac{number\ of\ pairs\ between\ P\ and\ Q\ judged\ as\ "different"}{number\ of\ total\ pairs\ between\ P\ and\ Q} \qquad (2)$$

4.4.2. Style (character) divergence in the six-dimensioned annotation feature space

Each annotated exemplars is a point in the six-dimensional annotation feature space. If these features can capture the speech style of each character, points from the same character should cluster together. Thus, the clustering of them in the six-dimensioned space should represent the character's speaking style. Each character in the six-dimensioned annotation feature space is then modeled by 6 Gaussian distributions, one distribution for each dimension. Therefore, the character distance in the six-dimensioned annotation feature space can be measured by Kullback-Leibler divergence between the character models.

Symmetric Kullback-Leibler divergence (KLD) between GMMs [11] is used in this study. Given a set of N character models, denoted as $\{\Lambda_n, 1 \le n \le N\}$, the symmetric KL divergence is defined as the sum of relative entropy between model Λ_i and model Λ_j plus the relative entropy between model Λ_j and model Λ_i as shown in Eq. 3:

$$KLD_{(\Lambda_i,\Lambda_j)} = E_{\Lambda_i(X)}[\log \frac{\Lambda_i(X)}{\Lambda_j(X)}] + E_{\Lambda_j(X)}[\log \frac{\Lambda_j(X)}{\Lambda_i(X)}] \qquad (3)$$

where $\Lambda_i(X)$ and $\Lambda_j(X)$ are the occurrences likelihoods of observation X, given Λ_i and Λ_j respectively. According to [11], normalized KLD (NKLD) in Eq. 4 fits human perception better. Therefore, the normalized KLD is used as the acoustic distance between two voice characters. By calculating the normalized KLD between each pair of characters, an N-by-N (N is the number of the characters) symmetric acoustic distance matrix with zeros in the diagonal cells is obtained.

$$NKLD_{(\Lambda_i,\Lambda_j)} = \log(KLD_{(\Lambda_i,\Lambda_j)} + 1) \qquad (4)$$

Therefore, we define the distance between two characters P and Q in the annotation feature space as the sum of the 6 normalized Kullback-Leibler divergences as shown in Eq. 5:

$$Character\ Divergence(P,Q) = \sum_{annotation\ features} NKLD_l(P,Q) \qquad (5)$$

4.4.3. Correlation between character divergence and perception difference

Since both the character divergence and the perceptual difference between any two of the top ten characters are obtained, the correlation between them can be calculated. As Figure 1 shows, the correlation coefficient is 0.85, which indicates strong correlation. Therefore, the proposed six features are the key building elements of expressive styles. It is by manipulating these six features the voice talent creates various characters in the corpus. Based on this result, we conclude that the proposed annotation set is capable of capturing the characteristics of various expressive styles and discriminating them.

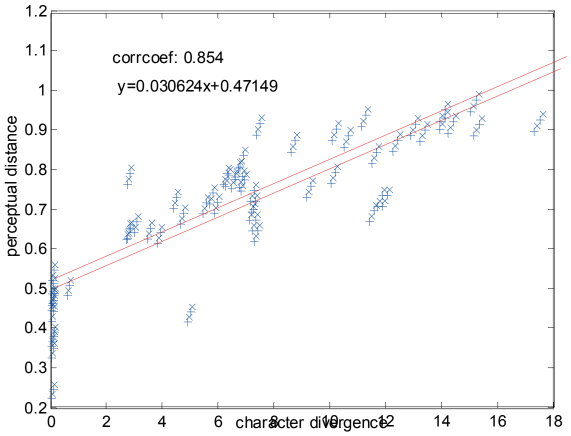


Figure 1: Correlation and regression between the character divergence and the perceptual difference *(*The regression line is: y=0.030624x+0.47149*.)*

5. Speaking style classification

Section 4 shows that the expressive utterances can be reliably and consistently annotated by non-experts. Under the assumption that different characters have different speaking styles, the proposed six labels are verified to well capture and discriminate the different styles (characters) in the audio-book corpus. However, the relationship between "characters" and the underlying "speaking styles" is not a one-to-one mapping. The speaking styles of some characters cannot be stably discriminated from others. Therefore, classifying the speaking styles in term of character IDs can still make mistakes. In this section, we will discuss how to cluster similar styles into statistically robust speaking styles in an unsupervised way. Based on the annotated 800 expressive speech utterances, a classification and regression tree (CART) [10] is created to automatically group sentences of similar expressiveness into several "pure" speaking styles. In the following subsections, we will describe in details how to construct the question set, measure each split, and build the tree.

5.1. Speaking style classification by CART

Each annotated utterance is a training sample represented in a six-dimensional categorical feature vector. To construct a CART from these training samples, a question set regarding the measure variables is needed, e.g., "Is the speaking manner = *narrative speech*?", "Is the speaking manner = *public speech*?", "Is the voice age = *old*?", etc., or a combination of these questions: "Is the speaking manner = *public speech*?" AND "Is the voice age = *old*?".

Once the question set is determined, CART uses a greedy algorithm to generate the decision tree. All training samples are placed at the root. The best question is then chosen from the question set to split the root into two. An "impurity" measurement of how well a question can split (partition) the data samples and reduce the global impurity is defined to pick the "best question". The algorithm then recursively splits the most promising node until a stopping criterion is reached. (e.g. a minimum number of samples in a leaf node, or a threshold of the global "impurity" reduction)

We define the impurity for any tree node t as follows:

$$H_t(Y) = \sum_{i \neq j} Perceptual\ Different(s_i, s_j) \qquad (6)$$

where Y is a random variable that decides the division of data sample X, s_i and s_j are two different samples in node t, with the perceptual difference between *Si* and *Sj* defined as Eq. 2. The impurity reduction of a question q to split node t into two children l and r is defined as Eq. 7. Now finding the best question becomes evaluating impurity reduction $\Delta H_t(q)$ for each potential question (split) and picking the question of the greatest impurity reduction as in Eq. 8.

$$\Delta H_t(q) = H_t(Y) - (H_l(Y) + H_r(Y)) \qquad (7)$$

$$q^* = \arg\max_q \left(\Delta H_t(q)\right) \qquad (8)$$

The global impurity of a tree T is defined as the sum of impurities for all the leaf nodes:

$$H(T) = \sum_{t \text{ is leaf node}} H_t(Y) \qquad (9)$$

Apparently, the global impurity decreases as the tree grows. By setting the minimum training samples on leaf nodes at 10, a CART tree of 4 layers, 9 leaf nodes is constructed, i.e., 9 "pure" speaking styles are obtained.

5.2. Speaking style interpretation

The interpretation of each speaking style can be explicitly understood from the CART structure. We listed them out in the following table. For example, the 1st speaking style can be summarized as heavy (A) in pitch, strong (A) in vocal effort, old (C) on voice age, loud (A), fast (C) in speaking rate, and in a conversational speaking manner (A).

Once the CART is built, the speaking style that best matches any annotated sample can be found by trickling down the CART tree. Statistically robust model for each of the nine speaking styles, like UBM-GMM, can be trained from the samples that belong to it. The speaking style models may help annotate the expressive styles of the whole corpus automatically. However, the modeling of the speaking style, as well as the speaking style identification for an un-annotated speech utterance, is beyond the scope of this paper.

Table 7: Speaking style interpretation

speaking style	pitch	vocal effort	voice age	loudness	speaking rate	speaking manner
1	A	A	C	A	C	A
2	B	B	A	C	A	A
3	A	A	C	C	C	B
4	A	A	C	B	C	A
5	B	B	B	B	B	A
6	B	B	B	B	A	C
7	C	A	C	A	C	A
8	A	A	B	B	B	A
9	A	B	C	C	C	A

6. Conclusions

In this paper, we carried out systematic experiments in an attempt to find a good annotation set for capturing the expressiveness in a speech signals. Different from other high-level emotional annotation set, the proposed labels focus on perceptually discernible clues that can be reliably distinguished by both human and machine. After applying the annotation to expressive speech samples in a multi-character audio-book, we found that the six features we used are effective in distinguishing different characters in the audio-book. This implies that the voice talent of the audio-book mimics characters mainly by manipulating the “pitch”, “vocal effort”, “voice age”, “loudness”, “speaking rate” and “speaking manner”. Using the 800 annotated expressive speech utterances, a compact classification and regression tree (CART) is created to automatically group the expressive speech utterances into several “pure” speaking styles. The interpretation of each speaking style can be explicitly understood from the CART structure. In our future work, we will look for the acoustic correlates of these features and find ways to label them automatically.

7. References

[1] M. Tatham and K. Morton, “Expressive in Speech: Analysis and Synthesis,” Oxford university press, 2004.

[2] N. Campbell, “Databases of Expressive Speech,” Proc. of ISCA ITRW on Speech and Emotion, pp. 34-38, 2000.

[3] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, et al. “Emotion recognition in human-computer interaction,” IEEE Signal Processing Magazine, 2001.

[4] C. O. Alm, R. Sproat, “Perceptions of Emotion in Expressive Storytelling,” Proc. of INTERSPEECH 2005, 2005.

[5] D. Erickson, “Expressive speech: Production, perception and application to speech synthesis,” Journal of Acoustic Sci. & Tech., 26, 4(2005).

[6] V. Auberge, N. Audibert, A. Rilliard, “Auto-annotation: an alternative method to label expressive corpora,” Workshop on Emotional Corpora (LREC 2006), 2006.

[7] M. Shami, W. Verhelst, “Automatic Classification of Emotions in Speech Using Multi-corpora Approaches,” Proc. of SPS DARTS 2006, 2006.

[8] H. Traunmuller, “Perception of speaker sex, age, and vocal effort,” http://www.ling.su.se/staff/hartmut/F97.pdf, 1997.

[9] J. Odell, D. Ollason, P. Woodland, S. Young and J. Jansen, The HTK Book for HTK V3.0, Cambridge University Press, Cambridge, UK, 2001.

[10] X. D. Huang, A. Acero, H. Hon, Spoken Language Processing, Prentice Hall, New Jersey, 2001.

[11] S. Kullback and R. Leibler, “On information and sufficiency,”Annals of Mathematical Statistics, vol. 22, pp. 79–86, 1951.

[12] Y. Zhao, D. Peng, L. Wang, M. Chu, Y. Chen, P. Yu, J. Guo, “Constructing Stylistic Synthesis Databases from Audio Books,” Proc. of INTERSPEECH 2006, 2006.

[13] L. Wang, Y. Zhao, M. Chu, F. Soong, Z. Cao, “Exploring the Expressive Space of a Voice Talent Using an Audiobook,” Proc. of Speech Prosody 2006, Dresden, Germany.

[14] C. Pereria, “Dimentions of Emotional Meaning in Speech,” Proc. of ISCA ITRW on Speech and Emotion 2000, Newcastle, UK.

Joint Analysis of Speech Frames for Synthesis Based on Lossy Tube Models

Karl Schnell and Arild Lacroix

Institute of Applied Physics, Goethe-University Frankfurt
Max-von-Laue-Str. 1, D-60438 Frankfurt am Main, Germany
schnell@uni-frankfurt.de

Abstract

This paper discusses a model-based synthesis approach focused on the estimation of model parameters. For the treated approach, tube models are used for analysis and synthesis of speech units. In comparison to the standard lossless tube model, an extended tube model is used which includes the frequency dependent vocal tract losses. The parameters of the tube models are estimated by minimizing the spectral error between the tube model and a speech segment. For the analysis of speech units, the time evolution of the parameters is taken into account. For that purpose, the speech segments are analyzed jointly which ensures smooth parameter trajectories. The investigations show that, especially for extended tube models, the joint analysis of frames improves the quality of the synthesized speech signals. Additionally, the differences of the results obtained by the standard and the extended tube model are discussed.

1. Introduction

Speech generation is nowadays often performed by concatenation of speech units. The speech units to be concatenated can be represented by the speech signals themselves [1] or by model-based descriptions. The model-based description has the advantage of flexibility and possible data reduction with the disadvantage of decreasing more or less the speech quality in comparison to natural speech signals. For model-based synthesis or re-synthesis, the speech signals are usually generated by a model describing the vocal tract and/or nasal tract ranging from the standard LPC-model to articulatory models, e.g. [2-4]. For synthesis, the common task of these models is to shape the spectral envelope of the synthesized speech. It is known that, for linear prediction, the harmonic structure of the spectrum of voiced speech influences the estimation of formant frequencies and bandwidths, especially for high pitch. Underestimating of bandwidth by linear prediction decreases the synthesis quality, which can be compensated by a subsequent bandwidth expansion or by specialized analysis methods [5-6]. In contrast, in [7] an analysis is proposed considering multiple measurements. Besides spectral approximation, an important feature of the models is the type of model parameters. The interpretation of the parameters varies from parameters describing the spectral envelope such as MFCCs, LSF, or formants to parameters describing the geometry of the vocal tract. Tube models describe the geometry of the vocal tract by tube areas. Articulatory models are mostly based on tube models with articulatory parameters such as the center or tip position of the tongue. Different articulatory vocal tract models exist mapping articulatory parameters to vocal tract areas or to mid-sagittal cross-sections, which restricts the scope to feasible vocal tract configurations. One problem is to control the articulatory parameters for synthesis. For a data-driven approach the parameters are estimated from speech signals, which is not an easy task [8, 9]. The fact that the articulatory vocal tract models are more or less imperfect affects the estimation. A more practical obstacle is that model adaptation to an individual speaker needs measurements of the speaker's anatomy [9], and another more general problem of estimation is the non-uniqueness of acoustic-to-articulatory mappings, which has several reasons. One reason for ambiguity can lie in the type of spectral features. For example, if only the first formants are taken into account, not the whole spectral information is used for the estimation. To tackle the problem of non-uniqueness, look-up tables, obtained from acoustic and articulatory measurements, combined with dynamic constraints can be used [10]. The benefit of articulatory parameters is their meaningful interpretation; however, their drawbacks for data-driven synthesis are the difficulties of the parameter estimation and the restriction of the area function, which can be unfavorable for a precise spectral approximation. In this contribution, a lossy tube model is used whose parameters are estimated from the magnitude spectrum of a speech segment. In comparison to the standard lattice filter, the lossy tube model implies the frequency dependent losses of the vocal tract. The losses which are introduced influence spectral estimation, especially on the formant bandwidths, and, additionally, on the vocal tract areas. The areas of the model are unconstrained enabling detailed spectral modeling. In comparison to the investigations in [11-12], the main focus of this contribution is the discussion of a joint parameter estimation of speech segments implying dynamic constraints and the realization of the acoustic synthesis.

2. Extended Tube Model

Tube models can be described in the time domain or frequency domain. The advantages of the frequency-domain description are the direct realization of frequency dependent vocal tract losses and variable tube lengths; however, one drawback is that for the realization of the acoustic synthesis a conversion from the frequency domain to the time domain has to be performed, which is usually realized via the calculation of the impulse response [13]. In comparison to frequency-domain models, time-domain tube models enable a direct realization of the acoustic synthesis. Here, time-domain tube models are treated for synthesis.

The simplest tube model is the LPC-model in lattice structure, which describes a lossless tube model. The standard lossless tube model consists of tube elements realized by lossless delays z^{-1} and adaptors describing the area discontinuities; the tube termination at the lips is realized by a reflection coefficient ±1. In this contribution, a lossy tube model is used, which considers losses within the vocal tract and at the lips by radiation. The frequency dependent loss effects by vibrating walls, viscous friction, and heat conduction within the vocal tract are modeled by lossy tube elements, which includes a lossy delay $\vartheta(z)$ instead of the lossless delay z^{-1}. ϑ is realized by a pole-zero system

$$\vartheta(z) = \frac{0.9875 \cdot z^{-1} - 0.9047 \cdot z^{-2}}{1 - 0.9182 \cdot z^{-1} + 0.0041 \cdot z^{-2}} \quad . \qquad (1)$$

The coefficients of ϑ are obtained by an optimization with respect to the mentioned loss effects [11]; here, for a sampling rate of 16 kHz. The lossy delays are placed alternately in the upper and lower path of the signal flow graph of the lossy tube model depicted in Fig. 1. The reflection coefficients $r(i)$ can be transformed into the areas by $a(i+1) = a(i) \cdot (1 - r(i)) / (1 + r(i))$. For synthesis, power waves are chosen as wave quantities determining the adaptors in Fig. 1. The advantage of power waves is that alterations of the coefficients don't change the wave energy. The termination $\alpha \cdot L(z)$ at the lips is realized frequency dependent by the pole-zero lip-impedance model from Laine [14] with the lip opening area as parameter; α is an additional real damping factor. The termination of the tube model at the other end is reflection free since a fixed termination at the glottis has the disadvantage that the vocal tract length has to be estimated. The tube model consists of $L = 24$ tube elements whose area configuration is described by the vector $\boldsymbol{r} = (r(1), r(2), \dots r(L))^{\mathrm{T}}$ of reflection coefficients. Since for a sampling rate of 16 kHz the vocal tract length is smaller than 24 tubes, the first reflection coefficients can model the constriction of the glottis.

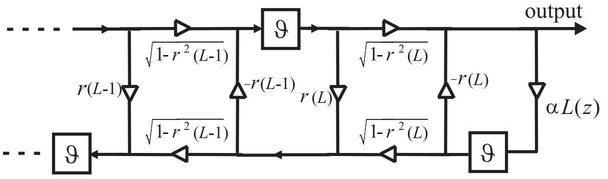


Figure 1: Flow graph of the lossy tube model for synthesis.

2.1. Parameter estimation of the lossy tube model

For the analysis of speech units, the model parameters are determined from the corresponding speech signals of the units. For that purpose, the units are segmented into overlapping frames s_k, which are multiplied by a Hanning window. For each frame, the parameters to be estimated are the reflection coefficients of the lossy tube model, whereas the parameters of the lossy delays and the lip termination are constant for the analysis due to ambiguity; the lip opening is chosen to 2.5 cm². To eliminate the influence of excitation and radiation on the spectral envelope, the speech segments s_k are filtered by a repeated adaptive pre-emphasis which is realized by inverse filtering with linear prediction of first order. The resulting pre-emphasis filter *P* consists of two real zeros

$$P(z) = \prod_{i=1}^{2} \left(1 - p(i) \cdot z^{-1}\right), \qquad (2)$$

which can balance better the spectral decrease of voiced speech than only one zero. Each segment *k* has its individual estimated pre-emphasis coefficients. After pre-emphasis, the reflection coefficients of the lossy tube model are estimated from the pre-emphasized speech segments s'_k by minimizing the error

$$e_k(S'_k, H_k) = \frac{1}{\pi} \int_0^{\pi} \left| \frac{S'_k(\omega)}{H_k(\mathrm{e}^{\mathrm{j}\omega})} \right|^2 \mathrm{d}\omega \, , \qquad (3)$$

which describes a spectral distance between the magnitude response $|H_k|$ of the tube model and the spectrum $|S'_k|$ corresponding to frame *k*. The transfer function of the tube model in Eq. (3) is calculated with adaptors which are equal to those used for the standard lattice filter. This is necessary for the error definition since the adaptors for power waves introduce a factor of the transfer function which is unfavorable for the estimation. The error definition (3) represents an inverse filtering approach in the frequency domain. Since the segments s'_k are finite signals, the integral in Eq. (3) can be represented by a sum with discrete frequencies. The error *e* is minimized by a gradient-based optimization algorithm. The gradient is approximated by error differences of small variations of individual reflection coefficients. Since the transfer function $H_k(\boldsymbol{r}_k, \mathrm{e}^{\mathrm{j}\omega})$ is a function of the parameter vector $\boldsymbol{r}_k$ of the *k*-th frame, the spectral error $e_k(\boldsymbol{r}_k)$ is a function of the reflection coefficients, too. The approximation of gradient is defined by

$$\nabla e_k = (\nabla_1 e_k, \nabla_2 e_k, \dots)^{\mathrm{T}} \qquad \text{with}$$

$$\nabla_i e_k = e_k(\dots r(i-1), r(i) + \varepsilon, \dots)^{\mathrm{T}} - e_k(\dots r(i-1), r(i), \dots)^{\mathrm{T}};$$

ε is a small constant about 10^{-8}. One iteration of the gradient algorithm is defined by a step in the direction of the negative normalized gradient with the adaptive step size δ:

$$\boldsymbol{r}_k^{j+1} = \boldsymbol{r}_k^{j} - \delta \cdot \nabla e_k / \max(|\nabla e_k|) \, ; \qquad (4)$$

the superscript with *j* indicates the iteration number and the function max() yields the maximum value. The step size $\delta = \sum_{l=1}^{7} c_l \cdot d_l$ is a parameterized function with the variables $c_l \in \mathbb{N}$ which are determined by

$$\arg\min_{c_l} \; e_k(\boldsymbol{r}_k^{j} - \delta \cdot \nabla e_k / \max(|\nabla e_k|)) \qquad (5)$$

with the constraints $\delta = \sum_{l=1}^{7} c_l \cdot d_l$ and $|r_k(i)| \le 0.99$. For the minimization of Eq. (5), the step size δ is, firstly, increased repeatedly by d_1 until the error is equal or greater in comparison to the previous error value or if $|r_k(i)| > 0.99$ is valid. Then, the next smaller d_l is used for increasing δ to minimize the error. The iteration is finished if the smallest d_l is reached. Here, the values of d_l are defined by $d_1 = 0.05$ and $d_{l+1} = d_l / 5$ for $l = 2 \dots 7$.

2.1.1. *Joint analysis of frames*

To ensure a smooth trajectory of parameter vectors, the frames are analyzed jointly. The joint analysis is realized by an exchange of interim results between adjacent frames during optimization. For that purpose, the parameter vectors of an individual iteration are averaged with the vectors of adjacent frames. For example, if the *j*-th iteration yields the vectors $\boldsymbol{r}_k^{j}$, then these vectors are updated by a mean of vectors including those of the neighboring frames. This averaging can be performed in different parameter descriptions. For that purpose, the vectors are transformed into the desired description $\boldsymbol{\psi}_k^j$, then the averaging is performed, and, finally, the averaged parameter vectors $\tilde{\boldsymbol{\psi}}_k^j$ are transformed back into reflection coefficients:

$$\begin{aligned} &\boldsymbol{r}_k^j \to \boldsymbol{\psi}_k^j \\ &\tilde{\boldsymbol{\psi}}_k^j = a_0 \cdot \boldsymbol{\psi}_k^j + \sum\nolimits_{i=1}^{W} a_i \cdot (\boldsymbol{\psi}_{k-i}^j + \boldsymbol{\psi}_{k+i}^j) \qquad (6) \\ &\tilde{\boldsymbol{\psi}}_k^j \to \tilde{\boldsymbol{r}}_k^j . \end{aligned}$$

The updated vectors $\tilde{\boldsymbol{r}}_k^j$ are used for the starting vectors $\boldsymbol{r}_k^{j+1}$ of the next iteration. The use of the averaging (6) imposes dynamic constraints and helps prevent divergence between neighboring frames during parameter optimization. The averaging (6) is performed in prescribed iterations with the numbers $j \in J$; J is a set of iteration numbers. The averaging can be performed every n-th iteration denoted by $J_n = \{n, 2n, 3n, ...\}$. The iterations without averaging allow that the vectors of the frames can evolve apart from each other a little bit. An irregular arrangement of the numbers in J can be suitable allowing a more unconstrained parameter evolution for the first iterations. For that purpose, the set $J_{ir} = \{40, 50, 52, 54, ..., 68, 70\}$ is used.

2.1.2. *Independent analyses of frames*

In comparison to the joint analysis, the frames can be analyzed by minimizing the errors $e_k(\boldsymbol{r})$ for each frame independently. Since there are no iterations with averaging, this independent analysis is denoted by the null set $J_0 = \{\}$.

2.1.3. *Analysis of speech units*

Both the joint and the independent analysis terminate after a prescribed number of iterations. In the following sections, analysis results are shown with a total iteration number of 70. The averaging by Eq. (6) is performed in the description of logarithmic areas. The values for the averaging in Eq. (6) are $W = 1$, $a_0 = 3/7$, and $a_1 = 2/7$ describing a weighted mean of adjoining frames, which emphasizes the middle frame.

3. Analysis of Diphones

The estimation procedure in the preceding section is used for speech analysis and synthesis. The sampling rate of the speech signals is 16 kHz. For re-synthesis, words are analyzed, whereas diphones are analyzed for synthesis. The diphones are from the diphone database de1 [15] for German from a female speaker. The analysis of the diphones yields the corresponding parameter vectors representing the diphones. To demonstrate the effect of the losses by the lossy tube model, also analysis and synthesis is performed with the lossless tube model represented by the standard lattice filter with power waves. In this case, the parameter estimation is performed as described in the preceding section, however, using the transfer function of the lossless model for the error definition of Eq. (3). The lossy model can be converted into the lossless tube model by the substitutions $\vartheta(z) := z^{-1}$ and $\alpha \cdot L(z) := -1$. If the estimation is performed with the lossless tube model by the independent analysis without averaging, the estimation results are comparable to those of the common linear prediction approach. For the analysis, the diphones are segmented into overlapping frames with the length of 625 samples and an overlap of 125. Figures 2-5 show the estimated logarithmic areas and the corresponding magnitude responses $|H_k|$ of each analyzed frame s_k' of diphones. Fig. 2 shows the results from the independent analysis of the frames without averaging using J_0. It can be seen that the estimated areas and magnitude responses fluctuate from frame to frame, especially in the case of the lossy tube model. These discontinuities of the model parameters decrease usually the quality of the synthesized speech and can be reduced by averaging during the optimization, which can be seen from Figs. 3 and 4. The iteration set J_2 is used for the results of Fig. 3, whereas the iteration set J_{ir} is used for the results of Fig. 4. The differences between the results using J_2 or J_{ir} are relatively small for the lossy tube model and almost negligible for the lossless tube model. The effect of J_{ir} is a slightly stronger emphasis on the temporal details in comparison to J_2. Here, a compromise should be made between smoothness and detailed approximation. It should be noted that temporal details can be caused by different effects: on the one hand, resonance movements by articulation which should be preserved and, on the other hand, by fluctuations of the excitation and by block-wise processing which should be ignored. Besides different uses of the averaging, the results obtained by the lossy and the lossless tube model show generally some differences in the estimated areas and magnitude responses. The areas estimated by the lossy tube model are more prominent in comparison to those of the lossless model; additionally, the shapes of the area configurations differ between the lossy and the lossless case. For example, for the fricative /v/ of the diphone /a-v/ in Fig. 5, the corresponding areas near the lips show more an open mouth for the lossless model, whereas the areas of the lossy model show more a closed mouth, which is more realistic. In general for the majority of the voiced sounds of the diphone database, the estimated logarithmic areas show reasonable vocal tract cavities. For example, from the figures 2-5 it can be seen that the estimated areas of the vowel /a/ shape a large front cavity ranging to the lips. For the sound /j/, the back cavity can be recognized. Due to the fact that the transfer function is more sensitive to the relationship of areas than to the absolute areas themselves, the logarithmic areas can be estimated more reasonably than the absolute areas. The assessment of the areas can be performed by regarding vocal tract areas from literature obtained from x-ray or NMR; however, this comparison can be used only for a rough assessment since these vocal tract shapes are from other subjects and the vocal tract configurations differ, in general, by coarticulation and by individual representations of the phonemes. An important pre-processing step for obtaining reasonable area configurations is an appropriate pre-emphasis [16]. Here, the repeated adaptive pre-emphasis seems to be suitable for that task.

In addition to the differences in terms of area functions, the magnitude responses estimated by the lossy tube model imply resonances of expanded bandwidths in comparison to the lossless model, which tends to underestimating of bandwidth. One reason for that may lie in the more realistic modeling of vocal tract acoustics by the lossy model.

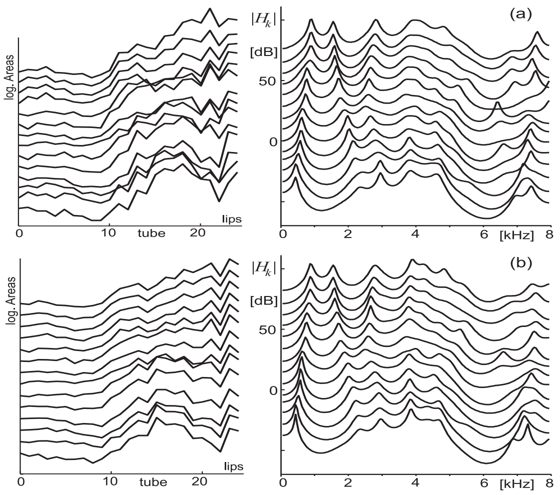


Figure 2: Estimated log. areas and magnitude responses of diphone /j-a/ by optimization without averaging using J_0, (a) for lossy tube model and (b) for lossless tube model.

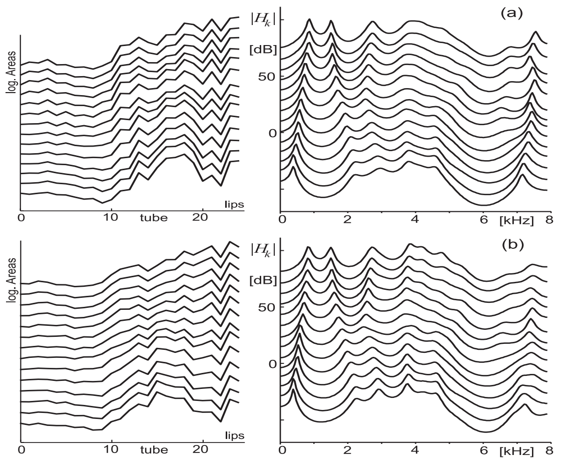


Figure 3: Estimated log. areas and magnitude responses of diphone /j-a/ by optimization with averaging using J_2, (a) for lossy tube model and (b) for lossless tube model.

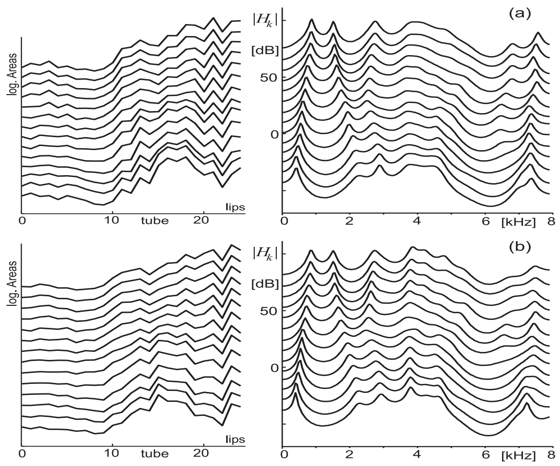


Figure 4: Estimated log. areas and magnitude responses of diphone /j-a/ by optimization with averaging using J_{ir}, (a) for lossy tube model and (b) for lossless tube model.

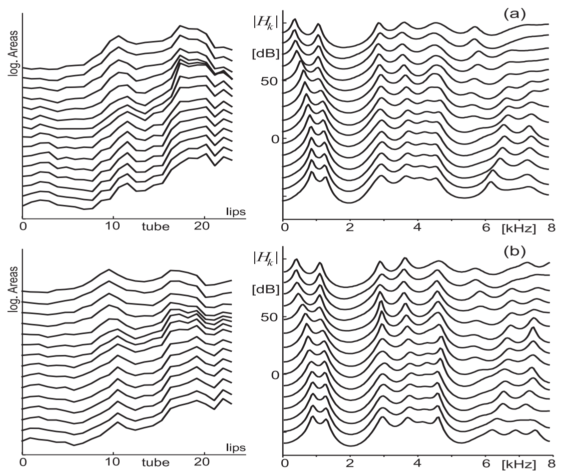


Figure 5: Estimated log. areas and magnitude responses of diphone /a-v/ by optimization with averaging using J_{ir}, (a) for lossy tube model and (b) for lossless tube model.

4. Synthesis of speech

The estimated areas and pre-emphasis coefficients obtained from the speech units are used for synthesis. For that purpose, the tube model is controlled by the parameter vectors successively. A de-emphasis filtering controlled by the pre-emphasis coefficients precedes the filtering of the tube model. To adapt the speech units to the required phoneme durations, parameter vectors can be doubled or omitted. The quality of the acoustic synthesis depends, aside from estimation of parameter vectors, also on the excitation of the tube system and on the concatenation of the model-based diphones. The diphone concatenation is performed by a parameter transition between the boundary vectors of the diphones to be concatenated, and is also treated in [12]. The excitation of the tube model is different for voiced and unvoiced sounds. For unvoiced fricatives, the excitation is relatively unproblematic and can be realized by noise. It is well known that the realization of the voiced excitation is more problematic due to its complicated structure and its impact on the speech quality; the voiced excitation has harmonic and non-harmonic components and its noisy components are non-stationary within a speech period. The use of an impulse train is the easiest way to implement a voiced excitation, however, with the disadvantage of introducing buzziness into synthesis [17]. To yield a more naturally sounding excitation, in [18] analyzed speech segments are used repeatedly for the voiced excitation. Related to that approach, here, a pitch-modified residual of an individual utterance of the schwa-sound is used for all voiced sounds, which avoids unnaturally sounding effects like the buzziness, for the most part. The pitch modification algorithm is based on a decomposition and parameterization of the residual signal in a low-pass filtered description, which is sketched in Fig. 6. The low-pass filter causes a smooth waveform related to the glottal flow. Each period of the low-passed residual *g* is decomposed into a small region *y* including the glottal closure instance and the remaining part *x*. The segments of the glottal closure instances are taken over unchanged, whereas the adaptation to the new period length is performed in the remaining parts. The remaining parts *x* are approximated by a polynomial which models the smooth contour of the waveform. Additionally, the approximation error is considered, which contains also noisy components. The modification of the lengths of the remaining parts is performed differently for the polynomial model and for the error of approximation, namely, by interpolation for the polynomial model and by a specific OLA-technique (overlap and add) for the error of approximation. After the modification of length, the parts are composed and the resulting signal is filtered by a high-pass filter. The pitch modification algorithm is explained in detail in [19]. For the realization of the excitation during synthesis, a sufficient number of adjacent periods between 15 and 30 of the schwa-sound is used one after the other. If the last period is reached, a period of the beginning is used randomly between the first and fifth period; in this way the repetition is irregular. Since the excitation is independent of the analyzed speech units, the acoustic synthesis needs only the estimated model parameters, which is favorable for data reduction. Fig. 7 shows spectrograms of synthesized speech signals of the German word “jawohl” [javo:l] by concatenation of the parameter vectors obtained from the diphones; the lossy tube model is used for analysis and synthesis. Figure 7(a) results from the synthesis with the excitation using the pitch-modified residual of the schwa-sound, whereas Fig. 7(b) results from the synthesis with impulse train excitation. The impulse train causes a harmonic structure in the whole frequency range, whereas the residual-based excitation reduces the harmonicity in the higher frequency range comparable to natural speech. A perceptive evaluation shows that the synthesized speech signals with the residual-based excitation sound less peaky with significantly reduced buzziness in comparison to synthesis with impulse train excitation. In general, the residual-based excitation yields a more natural timbre. It should be reiterated that the residual-based excitation is independent from the analyzed speech units. Figure 8 shows a segment of the synthesized speech signal. It can be seen that the synthesized speech signals with the residual-based excitation produces waveforms like natural speech, whereas the impulse train causes unnatural peaks in the synthesized speech waveforms.

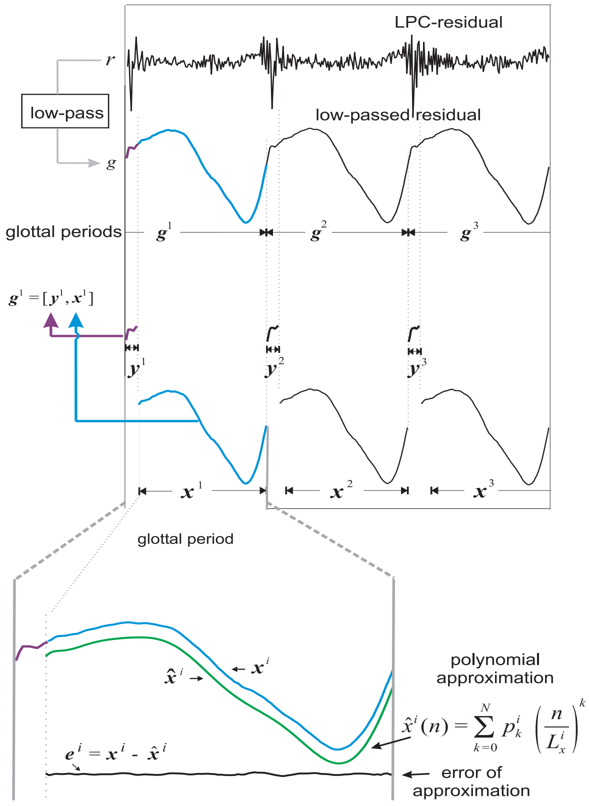


Figure 6: Sketch of the decomposition of the low-passed residual signal for pitch modification.

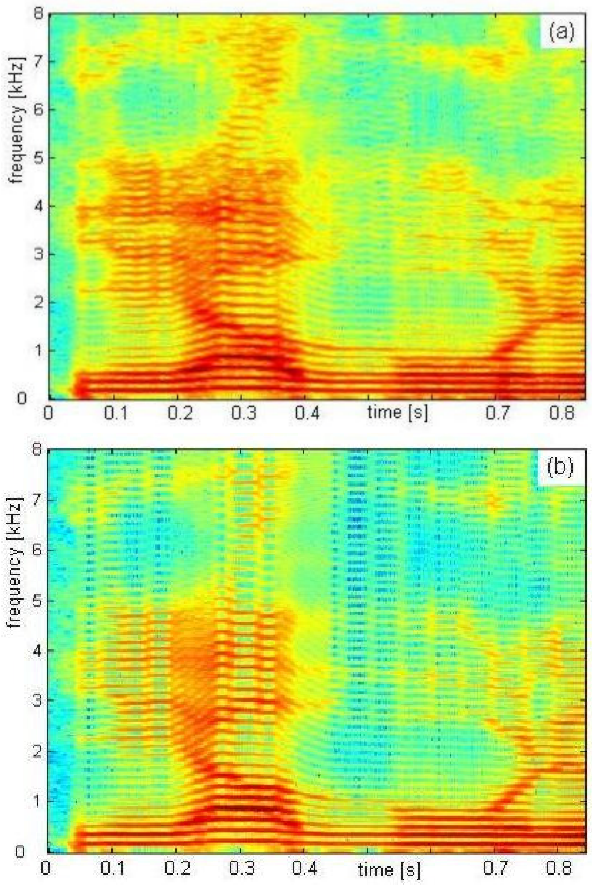


Figure 7: Spectrograms of synthesized word [javo:l] by model-based diphones with the lossy tube model: (a) with residual-based excitation obtained from the schwa-sound and (b) with impulse train excitation.

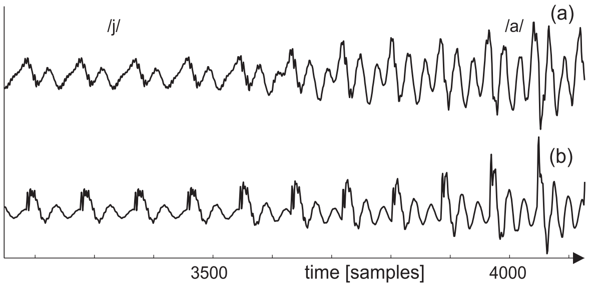


Figure 8: Segment of synthesized speech of [javo:l] by model-based diphones with the lossy tube model: (a) with residual-based excitation obtained from the schwa-sound and (b) with impulse train excitation.

4.1. Re-synthesis of words

To assess the influence of the lossy tube model without concatenation effects, whole words are analyzed and synthesized, too. The analyses reveal that the main spectral difference between the lossy and lossless model is that the estimated bandwidths are often narrower in the case of the lossless model. In Fig. 9, the resulting magnitude responses of the estimated areas are depicted for the German word “Julia” [jUlja] uttered by a male speaker; the averaging during the optimization is chosen with J_{ir}. Underestimating of bandwidth is usually caused by overemphasizing the harmonics of voiced speech. One effect of too small bandwidths concerning synthesis or re-synthesis with pitch modification is that artifacts can occur, for example, if the shifted harmonics don’t match the resonances with small bandwidths. Hence, the bandwidths should be not too small. In comparison to bandwidth expansion methods [5-6], the investigations show that the use of the lossy tube model implies an avoidance of bandwidth underestimating inherently.

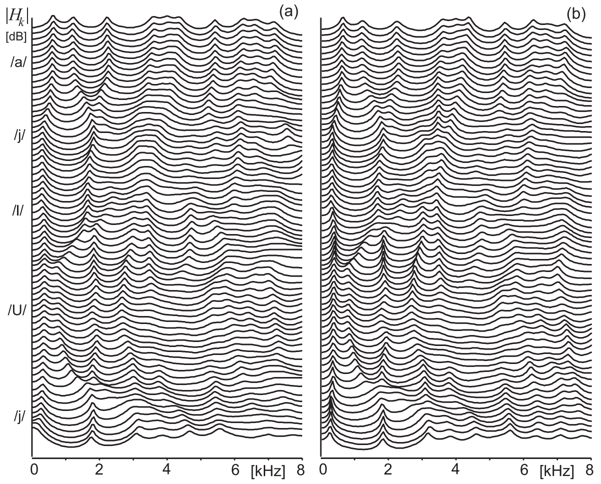


Figure 9: Estimated magnitude responses of word [jUlia] by optimization with averaging: (a) lossy tube model and (b) lossless tube model.

5. Conclusions

The parameter estimation and the realization of an acoustic synthesis for a model-based approach based on tube models is discussed. The results show that independent estimation without averaging causes, especially for the lossy tube model, fluctuations from frame to frame decreasing the synthesis quality. To yield a continuous trajectory of parameter vectors, the proposed joint analysis of frames is favorable. The main differences of the estimation results relating to the type of tube model is that the estimated areas of the lossy model are more prominent and the bandwidths of the corresponding resonances are expanded in comparison to the lossless model. The voiced excitation can be can be realized by a repeated use of a pitch-modified residual segment.

6. References

[1] Hunt, A.J. and Black, A.W., "Unit Selection in a Concatenative Speech Synthesis System using a Large Speech Database", *Proc. ICASSP, Atlanta 1996*, pp. 373-376.

[2] Childers, D.G. and Wu, K., "Quality of Speech Produced by Analysis-Synthesis ", *Speech Communication, Vol. 9, No. 2, 1990*, pp. 97-117.

[3] Goodyear, C.C. and Wei, D., "Articulatory Copy Synthesis Using a Nine-Parameter Vocal Tract Model", *Proc. ICASSP, Atlanta 1996*, pp. 385-388.

[4] Sondhi, M.M. and Sinder, D.J., "Articulatory Modeling: A Role in Concatenative Text To Speech Synthesis" in *Text To Speech Synthesis: New Paradigms and Advances,* edited by Narayanan, S. and Alwan, A., Prentice Hall PTR, New Jersey, 2004, pp. 63-87.

[5] Tohkura, Y., Itakura, F., Hashimoto, S., "Spectral Smoothing Technique in PARCOR Speech Analysis-Synthesis" *IEEE Trans. ASSP, 26 (5), 1978*, pp. 587-596.

[6] Ekman, L.A., Kleijn, W.B., Murthi M.N., "Spectral Envelope Estimation and Regularization*", Proc. ICASSP, Toulouse 2006*, pp. 245-248.

[7] Shiga, A. and King, S., "Accurate Spectral Envelope Estimation for Articulation-to-speech Synthesis", *Proc. 5th ISCA Speech Synthesis Workshop, 2004*, pp. 19-24.

[8] Schroeter, J. and Sondhi, M.M., "Techniques for Estimating Vocal-Tract Shapes from the Speech Signal", *IEEE Trans. Speech and Audio Proc., 2(1), 1994*, pp. 133-150.

[9] Sorokin, V.N., Leonov, A.S., Makarov, I.S, Tsyplikhin, A.I., "Speech Inversion and Re-synthesis", *Proc. INTERSPEECH, Lisbon 2005*, pp. 3209-3212

[10] Gupta, S.K. and Schroeter, J., "Pitch-synchronous frame-by-frame and segment-based articulatory analysis by synthesis", *J.A.S.A., Vol. 94, 1993*, pp. 2517-2530.

[11] Schnell, K. and Lacroix, A., "Analysis of Lossy Vocal Tract Models for Speech Production", *Proc. INTERSPEECH, Geneva 2003*, pp. 2369-2372.

[12] Schnell, K. and Lacroix, A., "Model Based Analysis of a Diphone Database for Improved Unit Concatenation", *Proc. INTERSPEECH, Lisbon 2005*, pp. 2605-2608.

[13] Sondhi, M. and Schroeter, J.: "A Hybrid Time-Frequency Domain Articulatory Speech Synthesizer", *IEEE Trans. ASSP, 35(7), 1987*, pp. 955-967.

[14] Laine, U.K., "Modeling of lip radiation impedance in the z-domain", *Proc. ICASSP, Paris 1982*, pp. 1992-1995.

[15] Englert, F., "Acquisition of a Diphone Database for German", in *Forum Phoneticum 63, Speech Processing,* edited by Wodarz, H.-W., Hector-Verlag Frankfurt am Main, 1997, pp. 23–32.

[16] Wakita, H., "Estimation of Vocal-Tract Shapes from Acoustical Analysis of the Speech Wave: The State of the Art", *IEEE Trans. ASSP, 27(3), 1979*, pp. 281–285.

[17] Sambur, M.R., Rosenberg, A.E., Rabiner, L.R., McGonegal, C.A., "On reducing the buzz in LPC synthesis", *J.A.S.A., Vol. 63, 1978*, pp. 918-924.

[18] Matsui, K., Pearson, S.D., Hata, K., Kamai, T., "Improving Naturalness in Text-to-speech Synthesis using Natural Glottal Source", *Proc. ICASSP, Toronto 1991*, pp. 769-772.

[19] Schnell, K., "Pitch Modification of Speech Residual Based on Parameterized Glottal Flow with Consideration of Approximation Error", *Proc. ICASSP, Toulouse 2006*, pp. 737-740.

Are Rule-based Syllabification Methods Adequate for Languages with Low Syllabic Complexity? The Case of Italian

Connie R. Adsett and Yannick Marchand

Institute for Biodiagnostics (Atlantic), National Research Council Canada,
1796 Summer Street, Suite 3900,
Halifax, Nova Scotia, Canada B3H 3A7

Faculty of Computer Science, Dalhousie University,
Halifax, Nova Scotia, Canada B3H 1W5
{connie.adsett, yannick.marchand}@nrc-cnrc.gc.ca

Abstract

Syllabification information is a valuable component in speech synthesis systems. Linguistic rule-based methods have been assumed to be the best technique for determining the syllabification of unknown words. This has recently been shown to be incorrect for the English language where data-driven algorithms have been shown to outperform rule-based methods. It may be possible, however, that data-driven methods are only better for languages with complex syllable structures. In this paper, three rule-based automatic syllabification systems are compared and two data-driven (Syllabification by Analogy and the Look-Up Procedure) on a language with lower syllabic complexity - Italian. Using a leave-one-out procedure on 44,720 words, the best data-driven algorithm (Syllabification by Analogy) achieved 97.70% word accuracy while the best rule-based method correctly syllabified 89.77% words. These results show that data-driven methods can also outperform rule-based methods on Italian syllabification, indicating that these may be the best approaches to the syllabification component of speech synthesis systems.

1. Introduction

Automatic syllabification is the process of determining the proper placement of syllable boundaries in a given word. Syllables have been used as key features in text-to-speech (TTS) systems for diverse languages. For instance, knowledge of syllable boundaries in written words is an essential component of a speech synthesis system for generating regional accents in English [1]. In the Hindi language, it has been demonstrated that using syllables as the units for speech synthesis gives better performance than smaller sized units [2]. Syllable structure information is also used in Czech [3], [4], European Portuguese [5], German [6], [7], Italian [8], [9] and Romanian [10] TTS systems.

Rule-based approaches have traditionally been the preferred method of determining the syllabification of unknown words (for example, see [2], [5], [7], [8] and [10]). Once defined by linguists, rules are straightforward to implement and apply. However, this approach is often time-consuming and requires expert knowledge. More importantly, several studies have raised the question of whether rule-based methods are actually the best approach to these tasks given the high performance of data-driven methods [11]–[14]. In particular, it has been demonstrated that, for syllabification of English words, data-driven methods perform significantly better than rule-based methods [14]. The success of data-driven methods on this language may be due to the fact that English is a language with a complex and irregular syllable structure [15], [16], which is challenging (and perhaps impossible) to fully capture with traditional linguistic rules.

Finnish, Italian and Spanish are considered to exhibit simple syllablic structure [16]–[20]. In this study, we performed an evaluation of syllabification methods on a language with simpler syllabic structure. The Italian language was selected to compare rule-based and data-driven methods for automatic syllabification in the same manner as these approaches were compared for English. This language was chosen because of the availability of both a large lexicon of syllabified Italian words and several rule-based algorithms for automatic syllabification in Italian.

2. Italian lexicon

The Italian lexicon used in this paper is part of the Italian Festival TTS project [21]. It was created by ITC-irst (Instituto Trentino di Cultura - Instituto per la Ricerca Scientifica e Tecnologio) and ISTC-SPFD CNR (Instituto di Scienza e Tecnologi della Cognizione - Sezione di Padova “Fonetica e Dialettologia” - Consiglio Nationale della Ricerche) and is freely available at `www.pd.istc.cnr.it/Software/It-Festival/2.0/lex_ifd.zip` (last accessed 9 March 2007).

Each entry provides spelling, part-of-speech, pronunciation, stress, and syllabification information. A total of 440,084 entries exist in the original lexicon. Because there is such similarity in Italian between different forms of the same words, we endeavoured to reduce the lexicon to only one form of each word. For this reason, proper nouns, plurals, verb forms (apart from the infinitive), superlatives and comparatives, and homophones and homographs were removed from the lexicon.

Because syllabification information was given in the pronunciation domain and the three Italian rule-based syllabification algorithms operate on the spelling domain (see Section 3), all words which did not have the same number of letters as phonemes were also removed. This allowed syllable boundaries to be transferred to the spelling domain without any need of complex alignment processing. Using this simple alignment approach, only 8,697 words were removed. The resulting lexicon (referred to as the *Full* Italian lexicon below) consisted of

44,720 entries, which formed the basis for all experiments unless otherwise stated.

3. Rule-based algorithms

Three Italian-specific rule-based algorithms for automatic syllabification were tested. The algorithms selected were: Cioni's algorithm for the syllabification of written Italian [22], an implementation of Hall's ordered rules for Italian syllabification [23], and Bergamini's SYL-LABE syllabification algorithm [24]. None of these methods have been tested or compared in order to determine accuracy. All three algorithms operate on the spelling domain.

3.1. Cioni's algorithm for the syllabification of written Italian

Cioni [22] presents a method using what he claims to be a "minimal set of rules" developed with the assistance of Italian linguists. The source code (written in C) for this program is available from `www.di.unipi.it/~lcioni/AltroSoftware/sillabatore.tar.gz` (last accessed 9 February 2007).

A subset of these rules is provided by Cioni [22] and is listed below (where C denotes a consonant and V denotes a vowel):

1. CVCV → CV | CV;
2. $VC_1C_2V \rightarrow VC_1 | C_2V$, if $C_1 = C_2$;
3. $VC_1C_2V \rightarrow V|C_1C_2V$, if $C_2 = h$;
4. $VC_1C_2V \rightarrow V|C_1C_2V$ or $VC_1|C_2V$, if $C_1 = s$ and depending on the value of C_2;
5. VCCV → VC|CV;
6. $VC_1C_2C_3V \rightarrow VC_1|C_2C_3V$, if $C_1 \neq s$;
7. $VC_1C_2C_3V \rightarrow V|C_1C_2C_3V$ or $VC_1|C_2C_3V$, if $C_1 = s$ and depending on the values of C_2 and C_3;
8. VCCCCV → VCC|CCV in most cases;
9. $V_1V_2 \rightarrow V_1|V_2$, if $V_1 \in \{a,e,o\}$ and $V_2 \in \{a,e,o\}$;
10. $V_1V_2V_3V_4 \rightarrow V_1V_2V_3|V_4$, if $V_1V_2V_3$ is a triphthong;
11. $V_1V_2V_3 \rightarrow V_1|V_2V_3$ if $V_1 \in \{a,e,o\}$;
12. $V_1V_2V_3 \rightarrow V_1V_2|V_3$, if $V_1 = i$ and $V_2 \neq u$, or $V_1 = u$ and $V_2 \neq i$.

Rules are also included to specify which pairs of vowels form diphthongs and therefore cannot be separated into different syllables. All rules are applied recursively by searching through the given word from left to right.

3.2. Hall's ordered rules for Italian syllabification

Hall [23] lists six ordered rules for breaking single Italian words into syllables:

1. $C_1C_2 \rightarrow C_1|C_2$, if $C_1 = C_2$;
2. $C_1C_2 \rightarrow C_1|C_2$, if $C_1 = c$ and $C_2 = q$;
3. $C_1C_2 \rightarrow C_1|C_2$, if $C_1 \in \{m,n,r,l\}$;
4. VCC → V|CC;
5. VCV → V|CV;
6. never divide a sequence of vowels into multiple syllables.

He provides two additional rules for division across word boundaries. The first states that a syllable boundary should never be placed immediately following an apostrophe which connect two words; for example, "l'albero" (*the tree*). The second concerns placement of syllables in musical scores. In this environment when a final syllable in a word ends in a vowel and the next word begins with a vowel and they must be both sung on the same note or over tied notes, it is necessary to indicate that they form a single syllable. This is done by decreasing the space between the two syllables.

These rules are given with the intent of assisting Italian instructors in teaching students how to divide Italian words in the spelling domain. Because they are fully described, it was possible to implement a rule-based automatic syllabification program[1] using these rules for the purpose of evaluation.

3.3. Bergamini's SYL-LABE program

Bergamini's rule-based syllabification algorithm is called SYL-LABE and was implemented in C (available at `http://www.pierotofy.it/pages/sorgenti/C/Utility/`[2] - last accessed 15 March 2007). The results of this algorithm were used as a gold standard in work on the automatic syllabification of Italian [24]. Two versions (1.0 and 3.3) of the algorithm are available and both were tested but only the results of the best algorithm (version 1.0) are reported. Implementation details are given in an Italian file which accompanies the download of version 3.3 of the program. One sample rule used in this system is VCV → V | CV.

The SYL-LABE program, as it was originally built, syllabifies only one word each time it runs. In addition to the word itself, stress information is required by the SYL-LABE algorithm in order to obtain syllabified output. Stress information was provided as given in the Italian lexicon. For example, to syllabify the word "sempre" (*always*), the input required is: `sempre` and `2` at the prompts given, where 2 is the location in the input string of the vowel in the stressed syllable. A simple loop program was written in order to obtain the syllabification of a list of words using the SYL-LABE program.

4. Data-driven algorithms

The data-driven algorithms used in this comparison were the same two that performed best on the syllabification of English words [14]: Syllabification by Analogy (SbA) and the Look-Up Procedure.

4.1. Syllabification by Analogy

Syllabification by Analogy is adapted from Pronunciation by Analogy, a method for automatic grapheme-to-phoneme transcription [25]–[28].

To compute the syllabification of an unknown word, it is first broken into substrings. Comparing these to substrings of syllabified words in the lexicon determines each segment's syllabic structure. This information is then used to construct the syllabification of the entire word.

Matching substrings are found by comparing the input word to all words in the dictionary. For each entry, the initial character in the input string is aligned with the final character in the syllabified word. The input string is then shifted left un-

[1] A Python implementation of Hall's rules is available from the authors upon request.

[2] The program may not be listed on the first page of the site.

til its final character is aligned with the initial character of the syllabified word. Before each shift all aligned characters are checked to determine whether there are any matching substrings at this point. If a match is found, syllabification information for this substring (obtained from the syllabified dictionary entry) is stored in a syllabification lattice. Using this procedure, the input word is compared to all words in the dictionary.

For example, the word “able” is represented as the input string `a?b?l?e` where `?` represents each position between letters (juncture) at which a syllable boundary may occur. The syllabified word “syl | la | ble” is represented as `s*y*l|l*a|b*l*e` where `*` and `|` represent non-syllable and syllable boundaries, respectively. When compared to find matching substrings, `?` may be matched with either `|` or `*`. Using “able” as the unknown word and “syllable” as the word from the lexicon, the matching process for them is shown in Table 1.

<table>
<tr><th>Step</th><th>Matching Procedure</th></tr>
<tr><td>1</td><td>a?b?l?e</td></tr>
<tr><td></td><td>s*y*l|l*a|b*l*e</td></tr>
<tr><td>7</td><td>a?b?l?e</td></tr>
<tr><td></td><td>s*y*l|l*a|b*l*e</td></tr>
<tr><td>Final</td><td>a?b?l?e</td></tr>
<tr><td></td><td>s*y*l|l*a|b*l*e</td></tr>
</table>

Table 1: *Example matching process using “syllable” (an entry in the lexicon) to syllabify “able” (an unknown word).*

The resulting syllabification lattice is a graph for which information from matching substrings form the nodes and arcs. Nodes represent the beginning and ending substring characters. Arcs are labeled with any intermediate substring characters along with the number of occurrences of this substring within the matches found in the dictionary. In the case of the above example, the nodes and arc inserted into the lattice from the substring `a|b*l*e` (found in step seven) would be $\bullet_{\mathtt{a}} \xrightarrow{\mathtt{|b*l*:1}} \bullet_{\mathtt{e}}$, along with all other subelements of this substring (for example, $\bullet_{\mathtt{a}} \xrightarrow{\mathtt{|:1}} \bullet_{\mathtt{b}}$ for `a|b` and $\bullet_{*} \xrightarrow{\mathtt{l*:1}} \bullet_{\mathtt{e}}$ for `*l*e`).

A decision function is used to find the all possible shortest paths through the lattice from the first to the last character of the input word. Syllabification is obtained from a given path by concatenating the node and arc labels (aside from the frequencies). If only one shortest path is found it is used to infer the syllabification of the unknown word.

When two or more shortest paths exist, a set of scoring strategies are used to determine the best syllabification. The three scoring strategies that gave the highest performance on the English language [29] were:

1. the maximum product of the arc frequencies along the shortest path;
2. the maximum frequency of the same syllabification within the shortest paths;
3. the maximum weak link value where ‘weak link’ is the minimum of the arc frequencies.

For the sake of consistency, these same scoring strategies were used to determine the syllabification of Italian words.

4.2. Look-Up Procedure

The Look-Up Procedure was also originally used for grapheme-to-phoneme transcription [30]. It has since been modified to perform automatic syllabification [11], [14]. This method uses N-grams (each consisting of a left context, right context and central letter) to learn and determine syllable boundaries.

During training, an N-gram is generated for each possible syllable boundary location in a word. Each N-gram is stored in a table along with how often a syllable occurs and does not occur following the central letter. Table 2 shows the table entries for the word `s*y*l|l*a|b*l*e`, using a left and right context of three letters (N = 7).

<table>
<tr><th></th><th colspan="2">Frequencies</th></tr>
<tr><th>N-grams</th><th>|</th><th>*</th></tr>
<tr><td>---syll</td><td>0</td><td>1</td></tr>
<tr><td>--sylla</td><td>0</td><td>1</td></tr>
<tr><td>-syllab</td><td>1</td><td>0</td></tr>
<tr><td>syllabl</td><td>0</td><td>1</td></tr>
<tr><td>yllable</td><td>1</td><td>0</td></tr>
<tr><td>llable-</td><td>0</td><td>1</td></tr>
<tr><td>lable--</td><td>0</td><td>1</td></tr>
</table>

Table 2: *Table entries for the word “s*y*l | l*a | b*l*e” used by the Look-Up Procedure.*

During testing, the closest matches to the N-grams from the test words are found in the table. Similarity between N-grams is determined using an N-element weight vector. For a given N-gram, if the frequency of a syllable boundary occurring after the central letter is higher than the frequency of no syllable boundary, a syllable boundary is placed in the test word.

For example, using the 7-grams stored in Table 2 to syllabify the word “able” requires finding the closest match to each of four 7-grams (`---able`, `--able-` and `-able--`) within the table. Using $[1, 4, 16, 64, 16, 4, 1]$ as the weight vector, the closest match to the first 7-gram in given above (`---able`) is the entry `yllable` with a similarity value of 85 $(64 + 16 + 4 + 1)$. The frequency of a syllable boundary occurring after the central letter in this pattern is greater than the frequency of no syllable boundary and therefore a syllable boundary is placed following the `a` in “able”.

The Look-Up Procedure was tested with all 15 weight vectors (given in Table 3) that were used in the comparison of automatic syllabification methods for English [14] and the study in which this technique was originally described [30].

5. Results

To compare the syllabification algorithms described above, a leave-one-out procedure was used whereby each word was removed from the lexicon in turn and its syllabification was inferred from all other words.

Results were computed using word and juncture accuracy. Word accuracy is simply the number of words syllabified by the method in exactly the same way as is given by the standard used (in this case, the Italian lexicon). Juncture accuracy compares syllabification at the sub-word level. Each position between letters is assessed to determine whether it was classified correctly. For example, the Italian word “sempre” has five junctures (denoted by a ‘`?`’) and can be shown as “`s?e?m?p?r?e`”. The accepted syllabification, according to the Italian lexicon, is “sem | pre”. If an algorithm syllabifies the word as “semp | re”, this is considered entirely wrong in terms of word accuracy, however it is 60% (3/5) correct in terms of juncture accuracy, as shown in Table 4 in which C and I correspond to correctly and incorrectly syllabified junctures.

Version	Left Context				Central Letter	Right Context				
	-4	-3	-2	-1	0	+1	+2	+3	+4	+5
1					1					
2				1	4					
3					4	1				
4				1	4	1				
5			1	4	16	4				
6				4	16	4	1			
7			1	4	16	4	1			
8			1	4	16	4	2			
9		1	4	16	64	16	4	1		
10		1	4	16	64	16	5	1		
11	1	4	16	64	256	64	17	4		
12		4	16	64	256	64	16	4	1	
13		4	16	64	256	64	17	4	1	
14		16	64	256	1024	256	64	16	4	1
15		16	64	256	1024	256	65	16	4	1

Table 3: *Weight vectors used in the Look-Up Procedure.*

<table>
<tr><th></th><th colspan="11">Syllabification of “sempre”</th></tr>
<tr><td>Output</td><td>s</td><td>⋆</td><td>e</td><td>⋆</td><td>m</td><td>|</td><td>p</td><td>⋆</td><td>r</td><td>⋆</td><td>e</td></tr>
<tr><td>Lexicon</td><td>s</td><td>⋆</td><td>e</td><td>⋆</td><td>m</td><td>⋆</td><td>p</td><td>|</td><td>r</td><td>⋆</td><td>e</td></tr>
<tr><td>Junctures</td><td></td><td>C</td><td></td><td>C</td><td></td><td>I</td><td></td><td>I</td><td></td><td>C</td><td></td></tr>
</table>

Table 4: *Juncture accuracy example.*

The results for all automatic syllabification algorithms are presented in Table 5. Although all 15 weight sets were used for the Look-Up Procedure, only the top five are reported. The difference in performance between the best rule-based method (SYL-LABE) and the best data-driven method (SbA) is approximately 10%. A Chi-squared test (χ^2) reveals $\chi^2_{obt} = 2977.0$ for words and $\chi^2_{obt} = 5030.3$ for junctures. These differences between SbA and SYL-LABE are highly statistically significant ($p < 0.01$ in both word and juncture accuracy).

Discrepancies in performance amongst the rule-based methods are attributed to differences in the rule sets used by each. Although some rules are consistent between methods, others are vastly different. For example, Hall’s rules [23] state that no vowel cluster should ever be separated by a syllable boundary while Cioni [22] states that when the vowels ‘a’, ‘e’, and ‘o’ are adjacent within a word (e.g. ‘ae’ or ‘eo’), they are not in the same syllable. Simple analysis of the lexicon reveals that, for bigrams involving the vowels ‘a’, ‘e’, and ‘o’, Hall’s rule is nearly always wrong while Cioni’s rule is often correct.

	Percentage Correct	
Algorithm	Word	Juncture
Cioni	86.59	97.78
Hall	81.59	97.24
SYL-LABE	89.77	97.89
Syllabification by Analogy	97.70	99.67
Look-up Procedure		
version 10	96.43	99.54
version 11	96.04	99.49
version 8	96.02	99.49
version 13	95.93	99.48
version 15	95.82	99.46

Table 5: *Syllabification results on the* Full *Lexicon.*

In addition, the overall results of all methods appear to be better in Italian than syllabification results in English, previously reported in [14]. This could be due to the fact that the *Full* Italian lexicon contained many more words than any of the lexicons used when comparing syllabification methods in English.

Because the correct syllable boundaries in a word are sometimes disputed, the English comparison used three lexicons: one from *Webster’s Pocket Dictionary* (19,596 entries), another from the *Wordsmyth English Dictionary-Thesaurus* (18,016 entries), and a third (called the *Overlap* database) which consisted of the 13,594 words with the same syllabification in both of the other lexicons [14]. To determine whether Italian is indeed easier to syllabify automatically than English, a randomized reduced set of the *Full* Italian entries, which matched *Overlap* lexicon size and, as closely as possible, the word length distribution was selected. The resulting *Reduced* Italian database also consisted of 13,594 entries.

	Words Correct (%)		
Algorithm	Italian	English	χ^2_{obt}
SbA	95.33	85.43	764.0
Look-up Procedure	91.60	73.53	1541.6
Rule-based	89.77	36.88	8186.0

Table 6: *Comparison of syllabification results for* Overlap *English [14] and* Reduced *Italian lexicons.*

The performance of the data-driven methods on the *Reduced* Lexicon was slightly less than on the *Full* Lexicon, as would be expected given that significantly fewer words were provided for training. However, the difference between the best data-driven algorithm (SbA - 95.33% for words) and best rule-based method (SYL-LABE - 89.77% for words) is still significant ($\chi^2_{obt} = 563.3, p < 0.01$).

Furthermore, from a computational perspective, these results quantitatively confirm linguistic and psychological findings, which indicate that Italian is simpler and more consistent in syllable structure than English, as stated by [17], [20], [31] and [32]. Although the CV syllable has been found to be most common in both English and Italian, this constituted only 34% of the syllables in English [17] as opposed to a full 60% in

Italian [31]. Such a marked difference should result in Italian being easier to syllabify than English. Table 6 compares the results for the Italian and English [14] languages, showing that, for SbA, the best Look-Up Procedure weights (version 10 for both languages) and the best rule-based methods in each of the two languages, the results obtained for Italian are significantly higher ($p < 0.01$). Indeed, although data-driven methods provide the most accurate results for Italian, rule-based methods still perform well with the best word accuracy at a full 89.77% in comparison to the poor performance of rules on English syllabification.

6. Conclusion

Previous studies show that data-driven methods outperform rule-driven methods in English syllabification tasks [14]. The purpose of this study was to extend the comparison of these two approaches to the syllabification of a language known to have lower syllabic complexity, namely Italian.

When the results from Italian syllabification methods are compared to those from English, it is evident that, regardless of the method used, performance on the Italian lexicon is significantly better. This indicates that syllabification must be a more straightforward task in Italian which is not surprising due to the fact that Italian exhibits lower syllabic complexity.

This comparison on a set of about 44,000 Italian words also confirms the superiority of the data-driven algorithms in terms of both word and juncture accuracy. Overall, all the algorithms presented attain at least 80% word accuracy. The best data-driven method (SbA) reaches a word accuracy of 97.70%, whereas the best rule-based method (SYL-LABE) achieves 89.77%.

In conclusion, these results suggest that, when a syllabification procedure is included as a component of a TTS system, a data-driven method is a more appropriate choice than a rule-based approach, even for languages with low syllabic complexity.

7. Acknowledgements

The authors wish to thank Piero Cosi for providing help with the Italian lexicon. This work was supported in part by funding from the Natural Sciences and Engineering Research Council of Canada (NSERC). In addition, the first author was funded by the National Research Council (NRC) Graduate Student Scholarship Supplement Program (GSSSP), and an Izaak Walton Killam Predoctoral Scholarship.

8. References

[1] S. Fitt, “The generation of regional pronunciations of English for speech synthesis,” in *Proceedings of Eurospeech 1997*, 1997, pp. 2447–2450.

[2] S. P. Kishore and A. W. Black, “Unit size in unit selection speech synthesis,” in *Proceedings of Eurospeech 2003*, 2003, pp. 1317–1320.

[3] I. Kopecek, “Syllable based speech synthesis,” in *Proceedings of the 2nd International Workshop on Speech and Computer (SPECOM ‘97)*, 1997, pp. 161–165.

[4] I. Kopecek, “Automatic segmentation into syllable segments,” in *Proceedings of the First International Conference on Language Resources and Evaluation*, 1998, pp. 1275–1279.

[5] C. Oliveira, L. C. Moutinho, and A. Teixeira, “On European Portuguese automatic syllabification,” in *Proceedings of Interspeech 2005*, 2005, pp. 2933–2936.

[6] B. Möbius, “Word and syllable models for German text-to-speech synthesis,” in *Proceedings of the Third ESCA/COCOSDA Workshop on Speech Synthesis (SSW3-1998)*, 1998, pp. 59–64.

[7] M. Libossek and F. Schiel, “Syllable-based text-to-phoneme conversion for German,” in *Proceedings of the Sixth International Conference on Spoken Language Processing (ICSLP-2000)*, 2000, pp. 283–286.

[8] P. B. de Mareüil, “Linguistic-prosodic processing for text-to-speech synthesis in Italian,” in *Proceedings of the Sixth International Conference on Spoken Language Processing (ICSLP-2000)*, volume 1, 2000 pp. 697–700.

[9] M. Refice and M. Savino, “Automatic grapheme-to-phoneme conversion for Italian,” *Archives of Control Sciences*, vol. 15, pp. 415–428, 2005.

[10] D. Burileanu and C. Negrescu, “Prosody modelling for an embedded TTS system implementation,” presented at the 14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, 2006.

[11] W. Daelemans and A. van den Bosch, “Generalisation performance of backpropagation learning on a syllabification task,” in *Proceedings of Twlt3: Connectionism and Natural Language Processing*, 1992, pp. 27–37.

[12] A. van den Bosch and W. Daelemans, “Data-oriented methods for grapheme-to-phoneme conversion,” in *Proceedings of EACL ’93*, 1993, pp. 45–53.

[13] R. I. Damper, Y. Marchand, M. J. Adamson and K. Gustafson, “Evaluating the pronunciation component of text-to-speech systems for English: A performance comparison of different approaches,” *Computer Speech and Language*, vol. 13 pp. 155–176, April 1999.

[14] Y. Marchand, C. R. Adsett, and R. I. Damper, “Automatic syllabification in English: A comparison of different algorithms,” *Language and Speech*, to be published.

[15] M. Bruck, R. Treiman, and M. Caravolas, 1995. “Role of the syllable in the processing of spoken English: Evidence from a nonword comparison task,” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 21, pp. 469–479, June 1995.

[16] M. Conrad and A. M. Jacobs, “Replicating syllable frequency effects in Spanish in German: One more challenge to computational models of visual word recognition,” *Language and Cognitive Processes*, vol. 19, pp. 369–390, March 2004.

[17] R. M. Dauer, “Stress-timing and syllable-timing reanalyzed,” *Journal of Phonetics*, vol. 11, pp. 51–62, 1983.

[18] P. M. Bertinetto, L. Cioni, and M. Agonigi, 1994. “The hierarchical structure of the syllable in Italian (Or: Does one really start from the beginning?),” *Quaderni del Laboratorio di Linguistica*, vol. 8, pp. 1–21, 1994.

[19] C. J. Alvarez, M. Carreiras, and Marcus Taft, “Syllables and morphemes: Contrasting frequency effects in Spanish,” *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 27 pp. 545–555, March 2001.

[20] P. H. K. Seymour, M. Aro, and J. M. Erskine, "Foundation literacy acquisition in European orthographies," *British Journal of Psychology*, vol. 94, pp. 143–174, May 2003.

[21] P. Cosi, F. Tesser, R. Gretter, and C. Avesani, "Festival speaks Italian!" in *Proceedings of Eurospeech 2001*, 2001, pp. 509–512.

[22] L. Cioni, "An algorithm for the syllabification of written Italian," in *Proceedings of the 5th International Symposium on Social Communication*, 1997, pp. 22–24.

[23] R. A. Hall Jr., "Ordered rules for Italian syllabification," *Italica*, vol. 51, pp. 305–307, Autumn 1974.

[24] R. MacKinney-Romero and J. Goddard, 2006. "Syllabification using decision trees, early results on three languages," in *3er Taller de Tecnologías del Lenguaje Humano*, 2006, pp. 282–287.

[25] M. J. Dedina and H. C. Nusbaum, 1991. "PRONOUNCE: A program for pronunciation by analogy," *Computer Speech and Language*, vol. 5, pp. 55–64, January 1991.

[26] F. Yvon, "Grapheme-to-phoneme conversion using multiple unbounded overlapping chunks," in *Proceedings of Conference on New Methods in Natural Language Processing (NeMLaP-2'96)*, 1996, pp. 218–228.

[27] R. I. Damper and J. F. G. Eastmond, "Pronunciation by Analogy: Impact of implementation choices on performance," *Language and Speech*, vol. 40, pp. 1–23, March 1997.

[28] Y. Marchand and R. I. Damper, "A multistrategy approach to improving pronunciation by analogy," *Computational Linguistics*, vol. 26, pp. 195–219, June 2000.

[29] Y. Marchand and R. I. Damper, "Can syllabification improve pronunciation by analogy of English?" *Natural Language Engineering*, vol. 13, pp. 1–24, March 2007.

[30] A. J. M. M. Weijters, "A simple look-up procedure superior to NETtalk," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN-91), volume 2*, 1991, pp. 1645–1648.

[31] S. Frota and M. Vigário, "On the correlates of rhythmic distinctions: The European/Brazilian Portuguese case," *Probus*, vol. 13, pp. 247–275, 2001.

[32] C. Burani, L. Barca, and A. W. Ellis, "Orthographic complexity and word naming in Italian: Some words are more transparent than others," *Psychonomic Bulletin & Review*, vol. 13, pp. 346–352, April 2006.

Spoken Language Conversion with Accent Morphing

Mark Huckvale & Kayoko Yanagisawa

Department of Phonetics and Linguistics
University College London, London, U.K.
m.huckvale@ucl.ac.uk, k.yanagisawa@ucl.ac.uk

Abstract

Spoken language conversion is the challenge of using synthesis systems to generate utterances in the voice of a speaker but in a language unknown to the speaker. Previous approaches have been based on voice conversion and voice adaptation technologies applied to the output of a foreign language TTS system. This inevitably reduces the quality and intelligibility of the output, since the source speaker will not be a good source of phonetic material in the new language. This article contrasts previous work with a new approach that uses two synthesis systems: one in the source speaker's voice, one in the voice of a native speaker of the target language. Audio morphing technology is then exploited to correct the foreign accent of the source speaker, while at the same time trying to maintain his or her identity. In this paper we construct a spoken language conversion system using accent morphing and evaluate its performance in terms of intelligibility. Encouraging results tell us more about the challenges of spoken language conversion.

1. Introduction

Corpus-based speech synthesis systems can now be built from the voice of any individual and are capable of producing good quality spoken realisations of any utterance in the voice of the speaker in the language of that speaker. An interesting challenge is to further develop such systems so that they can produce convincing spoken realisations of any utterance in the voice of the speaker but in a language unknown to the speaker. We call this the *spoken language conversion* problem, to distinguish it from the speech-to-speech translation problem (which aims to recognise and convert the utterance text, too) and the voice conversion problem (which aims to keep the utterance the same, but change the speaker). An earlier term was Foreign Language Synthesis [1], but this doesn't capture the idea of preserving speaker identity. Spoken language conversion systems could be used as the output component of a speech-to-speech translation system, but they could also have other applications. They might be used to produce talking phrasebooks, to dub films in a foreign language, to speak embedded foreign language phrases in a text, or to provide pronunciation targets for language learning. For the purposes of discussion, let us call the source speaker S1, the language of the source speaker L1, and the required output language L2.

What are the challenges of spoken language conversion (SLC)? Firstly the aim must be to produce L2 utterances that in the minds of impartial listeners, *could have been* produced by speaker S1. Of course the spoken language of the speaker is one of the defining characteristics of his or her identity, so we don't expect that a speaker will necessarily be recognisable when speaking L2. Anecdotal evidence is that bilingual speakers can sound like different people in their two languages. It seems likely that individuals speaking an L2 with a poor accent are more identifiable, but we don't know of evidence for this. Nevertheless, the first challenge of SLC is to preserve in L2 those aspects of the identity of the speaker that are not related to their L1 accent.

A second challenge for SLC is to generate convincing phonetic forms in L2 using knowledge only of the speaker's spoken L1. Some L1 phonetic units may make perfectly satisfactory analogues for L2 units. Most languages seem to use vowel qualities close to [i], [a] and [u] for example, and have consonants similar to [p], [t] & [k], see [2]. Other L2 units may be found by selection from a range of occasional allophonic variants exhibited by S1 in L1 – for example, a required alveolar tap [ɾ] might be found by searching through an English speaker's realisations of /t/. Some L2 units might be generated by mixing or blending sounds in L1; for example new vowel qualities might be formed by a process of interpolation between forms found in L1. Lastly, however, there may be phonetic units in L2 that have no parallel in L1 - for example retroflex stops found in Hindi - and these need to be generated by a process of extrapolation beyond forms found in L1.

A third challenge for SLC is how to deal with differences (across languages) in the phonetic interpretation of phonological units in context. The realised form of a given phonological unit will vary according to the segmental and supra-segmental environment: for example, in English, /t/ has different allophones in different syllable positions, and vowels may be reduced in different stress positions. However these very contextual variations can themselves be different across languages. Some languages do not use aspirated stops, others may or may not velarise /l/; plosives undergo lenition in some languages but not others; some languages do not exhibit vowel reduction; others may allow voiceless vowels, and so on. Languages also vary phonotactically, such that phonetic sequences found in one language might be missing from another, which in turn may lead to poorly articulated clusters. So while it may be easy to find a commonality of phonetic forms across languages in some instances, each phonetic unit also has a range of contextual variants and these variants may be different in different languages. Thus an SLC system needs to be concerned with phonetic detail at a level below that normally considered in monolingual synthesis.

A fourth challenge comes from how we ought best to evaluate the performance of SLC systems. The recent tendency for the evaluation of monolingual synthesis systems has been the use of a mean opinion score (MOS), using a rating scale from 1-5. Such an approach is not without problems when applied to SLC. If we used MOS to evaluate SLC systems we would, of course, need to use native listeners of L2 for the rating. However SLC systems also need to be evaluated in terms of how well speaker identity is preserved, and this raises issues about how well individuals can be recognised when speaking another language anyway. In addition, if we seek to compare different SLC systems, it may be hard to disentangle the perceptual consequences of processing artefacts from the assessment of speaker similarity. Listeners may be more critical of a clean and precise synthesis of S1 in L2 that does not express S1's identity exactly, than a noisy, messy synthesis where identity is less easy to establish anyway. Finally, MOS experiments require a large pool of listeners, which make them expensive to perform. They can also be insensitive to small variations in system performance [3].

In this paper we will review previous technological approaches to the spoken language conversion problem. We will try to highlight what we see as their limitations. We then introduce a new approach based on accent morphing – a process that involves interpolation between two versions of a spoken utterance. We demonstrate the potential of accent morphing within the context of spoken language conversion by showing how well it improves the intelligibility of foreign-accented TTS synthesis to native listeners. We conclude by drawing some implications for the construction of future spoken language conversion systems.

2. Previous approaches to Spoken Language Conversion

Any synthesis system that can be controlled at a phonetic level can be made to simulate a foreign language simply by selection of appropriate units from the L1 inventory. We don't consider such approaches here since they will have severe foreign accents, although they might function as control conditions in SLC experiments. Perhaps the first approach to SLC that went beyond phonological selection from L1 was Campbell's foreign language synthesis system [1]. This system was based on the CHATR corpus-based synthesis system, but modifications were made at the level of unit-selection so as to choose corpus units for synthesis (from L1) that were best suited to implementing the required phonetic forms in L2. In conventional unit-selection, candidate units are selected on the basis of a phonological match to the target utterance. It is assumed that the phonetic detail in the selected speech signal sections is appropriate because of the match in phonological labels. For foreign-language synthesis, we can map the phonological labels, but this does not guarantee the appropriateness of the phonetic detail. Campbell's approach was to use a phonetic target for unit selection based on acoustic analysis of a synthesized native version of the utterance. Unit-selection then becomes a process to choose among phonetic units rather than phonological units. In terms of how well Campbell's system meets the challenges of SLC, we note that S1's voice is used in an unmodified form, and so in one sense S1's identity is maximally preserved. However since the process only selects from S1's available units, it does not address the problem of L2 units which are poorly realised or missing in the source system. While the acoustic matching to L1 might provide some appropriate contextual variants, it can't deal with contexts or variants that are missing in L1. Evaluation of the system was very limited, and performed only in terms of MOS on isolated words with no control condition.

The advent of speech-to-speech translation systems in the 1990s encouraged the development of speaker-adaptable text-to-speech systems: synthesis systems which were implemented in language L2 using some different speaker S2, but which could be modified to sound like S1. The dominant technique for this adaptation was then, and remains today, *voice conversion*. In voice conversion, an utterance is modified by some signal processing techniques to change the identity of the speaker, but to leave the linguistic content of the utterance unchanged. A number of voice conversion approaches have been proposed, e.g. [4,5,6]. All these techniques have at their heart a statistical model which maps spectral details across two speakers. An utterance spoken by speaker S2 is broken down into spectral vectors, then each of these is substituted by vectors estimated as representative of speaker S1 and the utterance resynthesised. The training of the mapping from S2 to S1 is performed by aligning equivalent speech signals in training data produced by S2 and S1. Gaussian mixture modelling of LPC-derived spectral envelopes is a common technique.

Voice conversion as described above is really only suited for mapping between speakers that speak the same language – this is because the mapping is learned from a training corpus of matched signals, and the matching relies on a phonetic equivalence of the signals. Attempts have been made to adapt voice conversion across languages, for example [7,8,9]. Mashimo [8] used a trick based on a bilingual speaker S2 who could speak both L1 and L2. A text-to-speech system was implemented in S2's voice in language L2, but then the voice conversion mapping was learned between S1 and S2 speaking L1. This allowed for the mapping to be learned from matched sentences spoken by both S2 and S1. Sündermann et al [9] adapted the idea so that the matched sentences in L1 were generated by unit-selection from a corpus of speaker S2 speaking L2. To understand the performance of these cross-language voice conversion systems, we need to understand more about how phonetic equivalence across languages is established. If for example, the mapping is learned from materials that are the same only in terms of phonological transcription using a phoneme-level association across languages, then it is likely that this mapping will fail to accommodate differences in phonetic detail. If, for example, voice conversion changed a native [ɾ] to a foreign [ɹ] to implement /r/, then intelligibility of the L2 utterances may suffer. This is just one example of a general issue about context sensitivity in cross-language voice conversion. Since the whole approach is based on estimating a single best spectral slice in S1 for a spectral slice found in S2, then there is no mechanism for the mapping to be made sensitive to the phonetic, phonological or prosodic context of the utterance. The 'best' mapped spectral slice may be different in different contexts: whether this is part of an /l/ or an /r/, whether it is in a stressed syllable or an unstressed one, whether it is phrase final or phrase initial, and so on. Evaluation of Sündermann's system indeed shows that MOS ratings after conversion are much lower than before. The process of cross-language voice conversion reduces the rating of the synthetic speech from 4.7 to 3.5. Worse, this reduction in quality does not seem to be matched by a large increase in the rating of S1 speaker similarity, here the MOS only increased from 1.6 to 2.0 after voice conversion. This may be because current voice conversion technology finds it easier to map overall spectral envelopes rather than details of the speaker's source signal [10].

Recently a third technology has been developed that could be capable of spoken language conversion. Latorre et al [11] describes an HMM synthesis system which is trained using multiple voices, and adapted using a single target voice. If such a synthesis system were trained with multiple languages, using an extended phone set to achieve a consistent labelling, then the approach could be used to generate a number of languages in one new target voice. The key difference to voice conversion is that adaptation is performed at the level of phones rather than at the level of spectral slices. This provides a level of context sensitivity, whereby the same spectral detail in two different phones might be mapped to different values. To perform the adaptation, a set of phonologically labelled utterances from S1 in L1 are fed into the system to adapt all the phone models even though only some of those phones in only some contexts will be present in the adaptation utterances. It seems that within the system, phones (across all languages) are clustered into groups, and a linear transformation of spectral means are applied to all units within a cluster, estimated from the adaptation material. It is not clear how this process affects the foreign language phones not present in L1, and the impact these have on intelligibility. In terms of preserving the identity of S1, Latorre's system is somewhat hampered by the relatively poor voice quality of HMM synthesis compared to corpus synthesis. However, HMM synthesis could use samples of S1's LPC residual to excite each phone model, and this could improve the identifiability of the speaker. Once again, the use of "equivalent" phonetic forms across languages, even when their precise realisation will be different in context, means that Latorre's system will also replace correct L2 forms with L1 approximations, leading to a reduction in intelligibility. Consider an L2 which uses [t^h] in one environment and [t] in another, if the adaptation process replaced both with a particular implementation of /t/ in L1, then the adapted speech will end up with incorrect detail. This type of effect could explain the reduction in the MOS of the L2 speech after adaptation (from 4.3 to 3.8), even when the MOS rating of identity improves (from 2.6 to 3.1).

In this section we have seen three approaches to spoken language conversion. We suggest that all have some weaknesses, many related to the use of an overly simplistic model of the phonetic relationships between languages. A table of phonological equivalences is not going to be good enough when the realisations of those units depends on the contexts in which they occur and in which language they are produced. The aim of our research is to explore these mismatches in more detail, and to that end we have developed another approach to spoken language conversion which provides more control over the phonetic mapping between L1 and L2.

3. Accent Morphing

The long term objectives of our research are to give a quantitative account of the differences between accents, both regional accents and foreign language accents. Spoken language conversion is a convenient testing ground for ideas about what aspects of accent are most salient to listeners. For any language pair, we can use the technology to generate and compare arbitrary utterances, then we can evaluate the consequences of differences in phonetic detail between them. Particularly we want to study how differences in phonological inventory and phonological interpretation across languages have an impact on the intelligibility and acceptability of a speaker S1 producing L2. To do this we needed a model of L2, a model of speaker S1 and the ability to control the phonetic composition of new utterances.

Our first insight was that the best knowledge we have for how to produce an utterance in L2, complete with all appropriate contextual variation, is through the use of a synthesis system in L2. So we use an L2 text-to-speech system as a knowledge source for how to speak L2, just like Campbell [1]. Similarly, the best knowledge we have about speaker S1, complete with how they produce different phonetic forms in different contexts, is through a synthesis system built in the voice of speaker S1. Inevitably this latter system will be in language L1, since we assume that speaker S1 does not speak L2.

Using our two text-to-speech systems, we can now generate a foreign-accented version of some target utterance U1 using system S1L1, and we can generate a native-accented version of the utterance U2 using system S2L2. If we could establish which aspects of U1 are inappropriate or inadequate, say by comparing it to U2, we can perform a signal processing transformation on only those aspects of U1 which need to be changed. The advantage of this is that U1 remains in the voice of speaker S1, and those aspects that are satisfactory are unmodified in the procedure. We call this technique *accent morphing*, because it takes as input two versions of the same utterance and generates a third version which borrows speaker information from one and accent information from the other. In other words, we implement a spoken language conversion system by generating the target L2 utterance using S1's voice, and then "patching up" the inevitable foreign accent in such a way as to minimise the impact on his or her identity.

How can we establish which aspects of U1 need to be changed? We have two sources of information: general information about the phonetics and phonology of the two languages, and specific information about the spectral qualities used in the utterances U1 and U2. We might, for example, simply identify particular phones which are likely to be problematic. On the other hand we might be able to use knowledge of accent variability and human perception to judge whether the existing implementation of a phone in U1 is within an acceptable range. The work done by Huckvale on the ACCDIST metric for comparing accents across speakers [12] might be used to establish which segmental qualities are furthest from the norm for the target accent.

How can we perform the signal modifications appropriate for this utterance? We might do this by "borrowing" temporal and spectral information from U2 and blending it with U1. For example, we might match vocal tract sizes across S1 and S2, so that we can predict target spectral envelopes for some phone in L2 in this context. A number of possible technical approaches could be taken to perform accent morphing. Techniques based on LP analysis and residual excitation seem practical [13]. We describe one particular implementation in the next section, although we are sure that better methods will be developed in the future. The concept presented here is not specific to some particular form of signal processing. However the spectral manipulation is performed, it only needs to be applied in some phonetic contexts and can be made sensitive to the requirement to preserve the identity of speaker S1.

How does this approach meet the challenges of SLC? Firstly it aims to re-use the speech of S1 in all places where it is satisfactory, this may mean re-use of the source signal, or of some whole segments or even of some frequency regions within segments. Information about phonetic units missing in L1 can be borrowed from U2, and furthermore, these will have appropriate contextual forms for L2. Lastly, we know that foreign accents are less intelligible to native listeners, therefore we can evaluate success by measuring the increase in intelligibility brought on by accent morphing. The next section evaluates one implementation of the idea.

4. Intelligibility Experiment

4.1. Aims

This experiment was designed to see if it is possible to implement an accent morphing system as part of a spoken language conversion application, and to assess the intelligibility of its output. Specifically, we addressed the following questions: (i) Can accent morphing improve the intelligibility of foreign-accented TTS output to native listeners? (ii) What are the relative contributions of morphed pitch, timing and segmental content to any change in intelligibility? (iii) Are there any interactions between changes in segmental content and changes in pitch and timing? This experiment did not address the impact of accent morphing on speaker identity, which is left for a further study. However we have tried as far as possible to minimise the impact of the processing on identity.

4.2. Source materials

The speech material consisted of 40 semantically unpredictable Japanese sentences, each containing 4 key words. These were adapted from [14]. Semantically unpredictable material was chosen to make the test difficult, so as to avoid ceiling effects in intelligibility scores, without requiring the addition of noise. Audio realisations of the utterances were acquired from (i) a native Japanese speaker, (ii) a Japanese TTS, and (iii) an English TTS using a custom dictionary. All versions were produced in a female voice in Standard Tokyo Japanese, at 16 kHz sampling rate. The Japanese TTS was the NeoSpeech VoiceText system using the Miyu voice. The English TTS was the AT&T Natural Voices system using the Audrey UK English voice. To make the English TTS system speak Japanese, romanised orthographic forms of the Japanese words were added to a custom dictionary. The Japanese pronunciations were entered using the best available phonetic units present in the English voice.

4.3. Accent Morphing

The accent morphing system takes two phonetically annotated and pitch-marked versions of an utterance: one from the source speaker and one from the model speaker. These are analysed and aligned and then used to generate a new target version of the utterance by selecting and combining characteristics from them. In this experiment, phonetic labelling and pitch period marking could not be obtained from the TTS systems (because we were using the SAPI interface to the systems), so phonetic labelling was performed through automatic alignment using an HMM tool (analign, in the SFS toolkit [15]). These were subsequently hand-corrected. Pitch period marking was performed using an automatic tool (SFS txanal). The best settings for this tool were optimised over the 40 sentences, but no hand correction was used.

Analysis consisted of pitch synchronous linear predictive coding (LPC) on windows centred on each glottal impulse and of a size equal to two pitch periods. In voiceless regions, the analysis window size was chosen on the basis of a smooth interpolated pitch contour, so as to provide continuity in analysis window size from frame to frame through the utterance. The LPC coefficients were then converted to a line spectral pair (LSP) representation, to make the coding of the spectral envelope more amenable to interpolation across speakers. The excitation residual was extracted from the source speaker for each separate glottal cycle and stored to complement the spectral information.

Alignment of the utterances was performed using a dynamic programming procedure working from an MFCC spectral representation of the speech, but constrained by the phonetic annotations. This gave an accurate cycle-by-cycle alignment between source and model speaker versions of the utterance, even within individual segments.

Morphing was then performed by generating the target utterance one glottal cycle at a time by selecting and interpolating pitch, timing and spectral characteristics from the set of aligned glottal cycle pairs. For some output time *t*, the corresponding source time is found from the required target timing. Similarly the synthesis window offset from the previous output cycle is found from the required target pitch. The required spectral envelope is found by interpolation of the envelopes of the matched cycles, while the required residual is just copied from the source speaker. Resynthesis from the interpolated LSP parameters and residual is then performed by overlap-add. In general, successful copying of spectral information from one speaker to another requires that the speakers have similar vocal tract sizes. However, normalisation of vocal tract size was considered unnecessary in this experiment, since both TTS voices appeared to have similar vocal tract sizes (assessed in terms of their mean F4 and F5 frequencies).

4.4. Experimental Conditions

Table 1: Description of each condition

E	Unmodified English TTS (source)
A	Segmental morphing alone (from J)
P	Pitch morphing alone (from J)
R	Rhythm morphing alone (from J)
PR	Pitch & Rhythm morphing (from J)
APR	Segment, Pitch & Rhythm morphing (from J)
J	Unmodified Japanese TTS (model)
N	Natural Japanese (control)

The conditions used in the experiment included the unmodified English TTS (E), Japanese TTS (J) and natural Japanese (N) versions of the sentences, together with accent-morphed variants of the English TTS. Details of the morphed conditions follow. In the 'A' conditions, target forms with a modified spectral envelope were morphed from the Japanese TTS as model speaker and the English TTS as source speaker. The only parts of the model spectral envelope that were used were regions below 3.5kHz in voiced parts of the sentence. Spectral information above 3.5kHz, spectral information in voiceless regions, and the excitation residual all came from the source speaker. This was to preserve the identity of the source speaker as much as possible, consistent with modifying phonetic quality towards the model. Previous studies (e.g. [16]) have shown that the residual and the high frequency spectrum contain important information about speaker identity. To limit artefacts arising from the switching of speaker data across and within frames, windowing was applied. Time windowing occurred across a single glottal cycle at the start and end of each voiced section, while frequency windowing extended from 3000 to 4000Hz, both using a linear interpolation.

In the ‘P’ conditions, the relative fundamental frequency (F0) changes for the phonetic segments were taken from the model speaker, while mean and variance of F0 were taken from the source. This ensured that the pitch contour was copied over but that the mean F0, important to speaker identity, was unmodified. In the ‘R’ conditions, the relative durations of the phonetic segments in the target were taken from the model, while the overall utterance duration was taken from the source. Thus the target had the same speaking rate as the source, but modified rhythm.

As well as the individual conditions, we also looked at the combination of pitch and rhythm morphing (PR), and the combination of segment, pitch and rhythm morphing (APR). Unfortunately, practical limitations in the size of the experiment prevented us from exploring all possible combinations. Table 1 provides a summary of the different conditions used.

4.5. Intelligibility Test

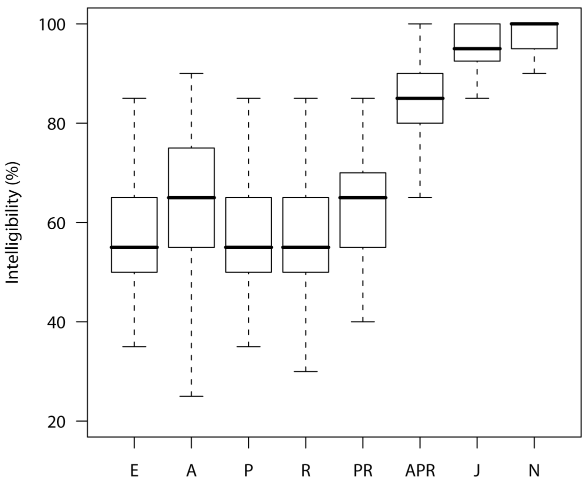
Recordings of the 40 sentences across the 8 different conditions were randomised in a Latin-square design into 8 lists, such that each list contained 5 sentences from each condition in random order. 56 native Japanese speakers each listened to one of the lists assigned randomly, such that each list was recognised 7 times overall. Thus for each condition, word intelligibility is based on 1120 observations. The listening experiment was conducted over the Internet, using specially-written web pages containing JavaScript functions and Java applets to prevent each sentence being played more than once. Listeners typed their responses into a web form where the sentence frame was provided and only 4 keywords needed to be completed for each sentence. Listeners were asked to input their responses using kanji and kana as appropriate, in order to disambiguate homophones which differ in pitch pattern. A brief practice session preceded the collection of actual intelligibility data, which were collected on our web server. Responses were marked in terms of percentage keywords correct. Exact homophones with the same pitch pattern were considered as acceptable forms.

Table 2: Mean intelligibility of each condition (N=1120)

Cond	%Intelligibility
E	56.96
A	64.46
P	58.04
R	58.30

Cond	%Intelligibility
PR	63.21
APR	84.20
J	94.91
N	95.71

Figure 1: Word intelligibility by condition



4.6. Results

The distribution of intelligibility scores across conditions is shown in Fig 1, and the means are summarised in Table 2. Conditions were compared in a pairwise manner using a Wilcoxon signed-rank test.

Unmodified conditions: E, J & N

As expected, the human Japanese speaker (N) gave almost perfect intelligibility scores. This control condition showed that the task and methodology were essentially satisfactory. The Japanese TTS system (J) also showed very good performance. A lower score would have been ideal to avoid problems with ceiling effects. Nevertheless it confirms that the Japanese TTS contains good quality segmental and suprasegmental information, adequate for use as a pronunciation target. The English TTS system speaking Japanese (E) showed considerably worse performance, as might be expected. This confirms that there is the potential for an accent morphing system to improve intelligibility.

Suprasegmental conditions: P, R & PR

Morphing just the pitch of the English TTS towards the Japanese TTS (P) did not trigger a significant increase in intelligibility. This is somewhat surprising considering Japanese does use pitch information for lexical access [17]. However in this experiment, the use of sentence materials rather than isolated words may have reduced the importance of pitch information. Morphing just the rhythm of the English TTS towards the Japanese TTS (R) also did not produce a significant increase in intelligibility. However the combined manipulation of pitch and rhythm (PR) did show a small but significant increase in intelligibility (p=0.03) over the unmodified condition (E). These facts might be explained if pitch information useful for lexical access was more readily available to listeners once it was placed in the right rhythmical framework. The interaction of pitch and timing like this has also been observed in studies such as [18].

Segmental conditions: A & APR

The modification of low-frequency spectral information in voiced regions (A) had a significant effect (p=0.007) on intelligibility over the unmodified condition (E). This change, which predominantly affects vowel realisations, clearly helps listeners identify words. However, the change caused by segmental quality change alone is rather small. One explanation for this might be due to morphing artefacts. For example an incomplete source-filter separation in the analysis could lead to some vowel colour being retained in the source residual.

The combination of segmental and suprasegmental morphing caused a large increase in intelligibility, from 57% to 84% (E to APR), reducing the gap between condition E and condition J by two thirds. Perhaps it is important to emphasise here that in the APR condition, much of the source speaker characteristics were still retained, as explained in 4.4. The combination of A and PR had a considerably greater impact on intelligibility than either factor separately. This suggests that the segmental changes necessary to improve the intelligibility are different in different prosodic contexts, so that using the segmental quality of the model voice is only suitable if the prosodic environment is also correct. Finally, the remaining gap between conditions APR and J could have a number of causes. It could be related to the segmental information present in the voiceless regions, in the excitation residual or in the spectrum above 3 kHz. Or it may be that the morphing process itself has a deleterious effect on the signal.

4.7. Discussion

We have described an experiment in the application of accent morphing to improve the intelligibility of foreign-accented Japanese TTS. The significant findings are as follows. Firstly the experiment showed that an accent morphing procedure can significantly improve intelligibility, despite any degradation in signal quality that may have been caused by signal processing. In this experiment segmental and suprasegmental information were taken from a Japanese TTS version of the source utterance, and we targeted morphing on the low-frequency spectral envelope in voiced regions, together with pitch and rhythm. A drop of 60% in word error rate (from 43% to 16%) was achieved using this procedure.

A second finding of the experiment is that morphing pitch or rhythm or segmental quality separately has surprisingly little effect on intelligibility. The lower intelligibility of the English TTS system speaking Japanese is not due to just one of these factors.

A third finding is that the combination of segmental and suprasegmental changes has a superadditive effect on intelligibility over the changes individually. This clear demonstration of an interaction between the segmental and suprasegmental properties of the signal is further evidence that phonetic differences between languages are contextually conditioned. It is only when the Japanese segmental forms are used in the right Japanese prosodic contexts that they significantly improve intelligibility.

5. CONCLUSIONS

In this paper, we have introduced a new approach to building a spoken language conversion system: a TTS system in L1 is used to produce L2 then the worst aspects of its foreign accent are corrected using accent morphing. The experiment we have presented did not evaluate a complete SLC system but concentrated on how phonetic differences between languages can have an impact on intelligibility. We have shown that the technique can produce highly intelligible Japanese utterances from an English TTS system. Detailed results also show that there are segmental and suprasegmental differences and segmental-suprasegmental interactions which need to be accommodated in a spoken language conversion system. For this particular language pair, we find that segmental quality changes alone do not have a large benefit. This suggests that the spectral mapping of the kind employed in voice conversion systems - which is applied separately from a change in prosody - may limit their ability to improve intelligibility. We have also shown that phonetic details need to be matched to the prosodic context – only when the two are in step do we see a significant improvement in the output. This suggests that a speaker adaptable TTS system that operates across languages may need to condition segmental adaptations on the prosodic context in which they occur.

We hope to extend this work in two directions: firstly to investigate in more detail which specific phonetic aspects of the speech most need to be modified to improve intelligibility. The fewer elements of the source signal that we need to change, the smaller will be the impact on speaker identity. Secondly, we hope to directly compare voice transformation and accent morphing techniques on the same data, in terms of the intelligibility of the resulting speech as well as the preservation of speaker identity.

6. REFERENCES

[1] Campbell, N., "Foreign Language Speech Synthesis", *3rd Speech Synthesis Workshop*, Australia,, 1998, 177-181.

[2] Ladefoged, P., Maddieson, I., "Sounds of the world's languages", *Blackwell Press*, 1996.

[3] Vazquez-Alvarez, Y., Huckvale, M., "The Reliability of the ITU-P.85 Standard for the Evaluation of Text-to-Speech Systems", *Proc. ICSLP-2002*, Denver, 2002.

[4] Abe, M., Nakamura, S., Shikano, K., Kuwabara, H., "Voice conversion through vector quantisation", *J. Acoust. Soc. Japan*, 11(2), 1990, 71-76.

[5] Stylianou, Y., Cappé, O., Moulines, E., "Statistical Methods for Voice Quality Transformation", *Proc. EuroSpeech 1995*, Madrid, Spain, 1995.

[6] Toda, T., Lu, J., Saruwatari, H., Shikano, K., "Voice conversion algorithm based on Gaussian mixture model with dynamic frequency warping of STRAIGHT spectrum", *Proc ICSLP-2000*, Beijing, 2000, 279-282.

[7] Abe, M., Shikano, K., Kuwabara, H. "Cross-language voice conversion". *Proc. ICASSP-90*, Albuquerque, 1990, 345–348.

[8] Mashimo, M., Toda, T., Kawanami, H., Kashioka, H., Shikano, K., Campbell, N., "Evaluation of Cross-Language Voice Conversion using Bilingual and Non-Bilingual Databases", *Proc EuroSpeech-2001*, Aalborg, 2001, 361-364.

[9] Sündermann, D., Höge, H., Bonafonte, A., Ney, H., Hirschberg, J., "Text-Independent Cross-Language voice Conversion", *Proc. ICSLP 2006*, Pittsburgh, USA, 2006.

[10] Sündermann, D., Höge, H., Bonafonte, A., Ney, H., and Black, A., "Residual Prediction Based on Unit Selection", *Proc. of 9th IEEE Automatic Speech Recognition and Understanding Workshop*, San Juan, Puerto Rico, 2005.

[11] Latorre, J., Iwano, K., Furui, S., "New approach to the polyglot speech generation by means of an HMM-based

speaker adaptable synthesizer", *Speech Communication*, 48, 1227-1242, 2006.

[12] Huckvale, M., "ACCDIST: a metric for comparing speakers' accents", *Proc. ICSLP-2004*, Jeju, Korea, 2004.

[13] Ye, H., & Young, S. "High quality voice morphing". *Proc. ICASSP 2004*. Canada, 2004, Vol 1, 9-12.

[14] Japan Electronics and Information Technology Industries Association, 2003. Speech Synthesis System Performance Evaluation Methods. JEITA IT-4001.

[15] Speech Filing System Tools. http://www.phon.ucl.ac.uk/resource/sfs/. Visited 5-Mar-07.

[16] Lin, Q., Jan, E.-E., Che, C. W., Yuk, D.-S., Flanagan, J., "Selective use of the speech spectrum and a VQGMM method for speaker identification", *Proc. ICSLP 1996*, Philadelphia, 1996, 2415–2418.

[17] Sekiguchi, T., Nakajima, Y., "The use of lexical prosody for lexical access of the Japanese language", *J. Psycholinguistic Research*, 28(4), 1999, 439–453.

[18] Ulbrich, C., "Interaction of timing and pitch in cross-varietal data", *Proc. 11th Australasian International Conference on Speech Science and Technology,* Auckland, 2006.

Comparative Investigation of Peak Alignment in Polish and German Unit Selection Corpora

Grazyna Demenko[1], Agnieszka Wagner[1], Matthias Jilka[2], Bernd Möbius[3]

[1] Dept. of Linguistics, Adam Mickiewicz University, Poznan, Poland
[2] Dept. of English Linguistics, University of Stuttgart, Germany
[3] Institute of Natural Language Processing, University of Stuttgart, Germany
lin@amu.edu.pl, wagner@amu.edu.pl, jilka@ifla.uni-stuttgart.de,
moebius@ims.uni-stuttgart.de

Abstract

This paper presents a comparative study on the temporal alignment of pitch peaks of H*L accents in Polish and German. Speech material used in the study came from the unit selection synthesis corpora of the Polish voice module of the BOSS system and the IMS German Festival TTS system. The major factors investigated were concerned with the influence of syllable structure on the one hand, as well as phrasal and tonal environment on the other hand. For the analysis of Polish falling accents, the effects of accent type, phrase type, and word position were also taken into account. Results show that in both languages, pitch peak placement is consistently affected by onset and coda type and by the tonal context (H or L tonal target preceding or following). Also, the position of the accent in the phrase is found to have a significant influence. Additionally, the results also reveal the difference between the two Polish falling pitch accents (static and dynamic).

1. Introduction

The alignment of pitch peaks is one of the key issues in the generation of natural prosody of synthetic speech. It is generally acknowledged that an F0 peak position on the syllable constitutes a distinctive feature of pitch accents.

In various intonation models the issue of peak alignment is dealt with in different ways. Phonological models distinguish between early and late peaks represented by two bi-tonal pitch accents: L*+H with a low target on the accented syllable followed by a rise or jump up to a peak on the post-accented syllable, and L+H* with a low target on the pre-accented syllable followed by a rise or jump up to a peak on the accented syllable (e.g. [5]).

In phonetic models [18] the alignment of pitch peaks is carefully controlled and determined on the basis of syllable and tonal environment. Contrary to those studies, there is, however, no control of the segmental and prosodic environment of the H*L peaks in the corpora used in this study (for example in [16] only H* peaks in the phrase-final syllable followed by a low phrase accent and a low boundary tone are considered) and the languages examined are Polish and German, not (American) English.

Proper identification of an F0 peak position is essential for the correct approximation and stylization of intonation contours, e.g. the Momel method [6] is based on the detection of target points – F0 maxima and minima – for application of the approximation function. At the moment we are developing a tool for automatic stylization of Polish intonation contours [4]. We hope that the analyses presented in this paper will bring more insight into the factors which influence peak alignment and that their results can also be used to help the automatic detection of pitch peaks for the purpose of intonation stylization.

2. Speech material and annotation

For the purpose of the investigation of peak alignment of Polish falling pitch accents 1150 phrases from the unit selection corpus of the Polish module for the BOSS speech synthesis system were used [2]. They were created by a linguist in order to provide speech samples that include accented syllables in different segmental (e.g. sonorant vs. voiceless obstruent onset and coda) and suprasegmental contexts (e.g. statement vs. exclamation). The phrases are of different length (varying from 1 to 13 words) and include 7136 accented syllables altogether (3031 instances of falling pitch accents).

The speech material was labelled using a tool *Annotation Editor* developed in the Institute of Linguistics AMU and at Poznan University of Technology. Labelling at the segmental level (transcription, segmentation into phonemes, syllables and words) was carried out automatically and with respect to prosodic features – semi-automatically. We distinguished two rising and two falling pitch accents differing with respect to whether the rise/fall is realized on the accented or the post-accented syllable (LH* and L*H for the rising accents, and H* and ΔH for the static and dynamic falling accents respectively), one rising-falling accent (L*HL), one accent realized by F0 interval between pre-accented and accented vowels (LI) and one realized by duration rather than pitch (LD). For the purpose of the current study the two types of falling accents were merged into a single class (H*L)[1].

At the prosodic phrase level information concerning phrase type was provided: statement, exclamation, question, minor intonation phrase (i.e. minor continuation and minor cadence). The speech database labelling made it possible to extract the following features for each accented syllable and include them in the analysis of peak alignment: 1) preceding and following accent type, 2) number of phonemes in the onset and coda, 3) onset and coda type (sonorant vs. voiced obstruent vs. voiceless

[1] An accent can be induced by two different mechanisms, a jump to a new pitch level in the syllable nucleus, and a change within the syllable nucleus. The use of a jump rather than a glide or vice versa is often dependent on the make-up of the syllables over which the accent spreads. If there is only one syllable a glide is more likely to be used. Differences between static and dynamic accent realizations are related to semantic function [3].

obstruent), 4) syllable position in the word, 5) number of syllables in the word, 6) foot position in the phrase (measured as a distance from the phrase end), 7) number of feet in the phrase and 8) prosodic phrase type.

For German the speech database of the IMS German Festival synthesis system [7] served as a corpus for the investigation. It mainly consists of sentences that were selected from a newspaper corpus by means of a greedy algorithm in order to ensure good coverage. The corpus was recorded by a professional male speaker, contains approximately 160 minutes of speech (2601 utterances with 17489 words [13] and was prosodically labeled using the GtoBI(S) System [11] (2681 instances of the H*L pitch accent).

3. Procedure

A *Praat* script was written which enabled to automatically obtain the information necessary for the peak alignment analysis in the Polish corpus. For each file the F0 contour was extracted and smoothed with a median filter with a window of 7 points which is useful for elimination of faulty F0 values and microprosodic effects on the F0 contour (see e.g. [12]). For each syllable the script provided: the F0 value at the syllable start and end, position and height of F0 maximum and minimum, mean F0 over the length of the syllable, F0 standard deviation and syllable duration. All this information was very useful for detection of possible errors in the F0 extraction or prosodic annotation and elimination of faulty data from the analysis.

The Festival synthesis system [1], which was used for the investigation of German, includes the “Festival feature functions” which can be used to describe a multitude of aspects of the segmental, syllabic, and prosodic structure of the utterances in the database. The measurement of the peaks themselves was done in a straightforward, automatic fashion by locating the F0 peak in a syllable labeled with a H*L pitch accent. In the case of H*L, the assumption that the peak is indeed in the same syllable is not problematic, but complications due to microprosody or voiceless regions cannot be avoided.

Throughout the analyses presented in this paper, syllable start was used as a reference point for measuring peak location, which is expressed as percentage of total syllable duration. Alternative reference points, such as start of voicing or start of rhyme, had previously been shown to yield less consistent results [8].

4. Analysis: Polish

4.1. Effects of segmental factors

4.1.1. *Onset and coda type*

Our analysis confirms previous findings concerning the influence of the onset and coda type - according to the van Santen & Hirschberg classification [16]: -V (voiceless obstruent), +V-S (voiced obstruent), +S (sonorant) - on peak alignment, i.e. that peaks occur earlier in the syllable if there is a sonorant in the onset and later if there is a sonorant in the coda (e.g. [8], [16]). On the basis of our data we found out that peaks occur relatively early in the syllable if there is no onset (mean: 34.22%, median: 23.05%), around the middle of the syllable if it starts with a sonorant (mean: 49.11%, median: 56.37%) or voiced obstruent (mean: 49.53%, median: 56.82%). The F0 peak moves towards the end of the syllable if the onset includes a voiceless obstruent (mean: 59.91%, median: 66.06%). As far as codas are concerned the effect is just the opposite: the peak occurs the earliest in the syllable if it includes a voiceless obstruent in the coda (mean: 39.41%, median: 45.47%), it moves towards the syllable center if there is a voiced obstruent or sonorant in the coda (mean: 43.26%, median: 48.42% and mean: 45.95%, median: 52.48% for the two coda types respectively). Peaks have the latest position if there is no coda (mean: 56.59%, median: 68.12%).

4.1.2. *Number of phonemes in the onset and coda*

The number of phonemes in the onset and coda also has a significant influence on peak alignment. In general, the more segments the onset includes the later the peak occurs and the less segments the coda consists of the later the peak is aligned in the syllable. The effects discussed here are illustrated in Figures 1 and 2:

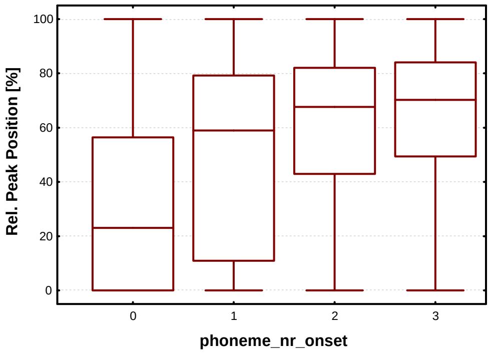


Figure 1: The effect of the phoneme number in the onset on relative peak position: 0 (median: 23.05%), 1 (median: 58.94%), 2 (median: 67.68%) and 3 (median: 70.24%)

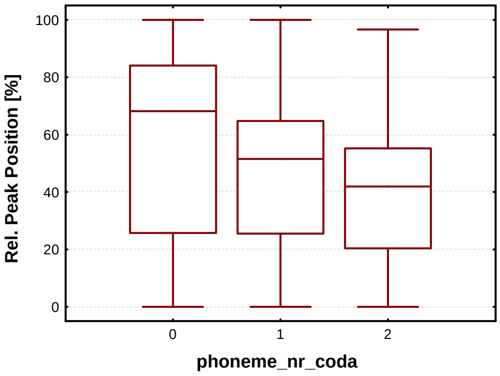


Figure 2: The effect of the phoneme number in the coda on relative peak position: 0 (median: 68.12%), 1 (median: 51.49%), 2 (median 41.93%)

4.2. Effects of suprasegmental factors

Suprasegmental factors investigated in the current study concerned phrasal and tonal environment of the accented syllable and word, and prosodic phrase type.

4.2.1. *Syllable position in the word*

The position of word accent in Polish is most often the penultimate syllable (see e.g. [14]). In our database there were only three instances of word-final accented syllables and four instances of word-medial accented syllables (in polysyllabic words): they were excluded from the analysis in order to avoid uncertain results. Thus, the effect of syllable position in the word was investigated on the basis of a two-way distinction between word-initial syllables and syllables pre-final in the word. An insignificant effect of syllable position in the word has been found: F0 peaks occur relatively earlier in word-initial syllables (mean: 53.02%, median: 60.24%) than in pre-final accented syllables (mean: 54.32%, median: 64.43%).

4.2.2. *Features of feet*

Our results on the influence of syllable number in the feet on peak alignment confirm previous findings. It was shown in [17] that in monosyllabic feet pitch peaks occur somewhere in the middle of the syllable, whereas in polysyllabic feet they are located earlier (i.e. towards the end of the foot-initial syllable). Additionally, in [9] it was observed that this effect concerns only first three syllables in the foot. Therefore, in the current analysis feet consisting of than three syllables and more were merged into one class. The results obtained for our data are illustrated in Figure 3:

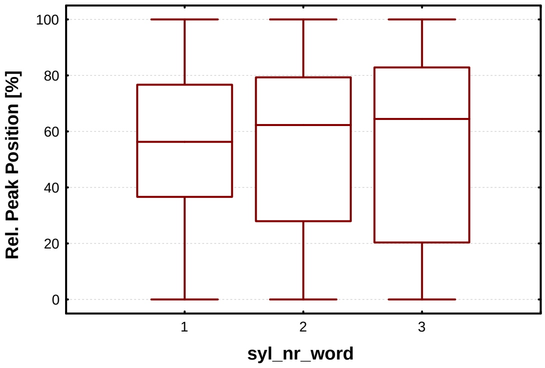


Figure 3: The effect of syllable number in the feet on relative peak position: 1 (median: 56.37%), 2 (median: 62.33%), 3 and more (median: 64.38%)

Considering feet position in the phrase an observation was made that the peak shifts towards the beginning of the syllable as the feet position changes from phrase-initial to phrase-final. In feet at the beginning of major intonation phrases the mean peak position is 76.45% into the syllable (median: 78.94%) and at the beginning of minor intonation phrases: 65.96% (median: 74.25%). The results suggest that intonation phrase type (major vs. minor) also plays a role in peak alignment. Syllables in phrase-medial feet have peaks located relatively later than syllables in feet of a pre-final position in the phrase (mean: 65.45%, medial: 69.94 vs. mean: 61.96%, median: 65.65%). In phrase-final feet F0 peaks occur in the first-third of the accented syllable (mean: 29.94%, median: 16.49%). It can be seen in Figure 4 that the accent type influences relative peak position as well: it seems that even though the mean peak position was the same for the two accent types when all instances of accented syllables were taken into account, the separation of falling accents depending on whether the fall is realized on the accents or post-accented syllable was justified (see section 2). The analysis of variance has proved that peak position is influenced significantly by both falling accent type (F [4,3031]=86.005, p<0.001) and distance of the accented syllable from phrase boundary (F [4,3031]=49.34, p < 0.001).

4.2.3. *Features of phrases*

On the basis of our data an influence of prosodic phrase type on peak alignment was found. We distinguished among three sentence modes: statements, questions and exclamations, and two types of minor prosodic phrases: ending with a cadence and signalling continuation. It has to be explained that in our database questions beginning with a question pronoun received the same phrase type label as statements, because they both.

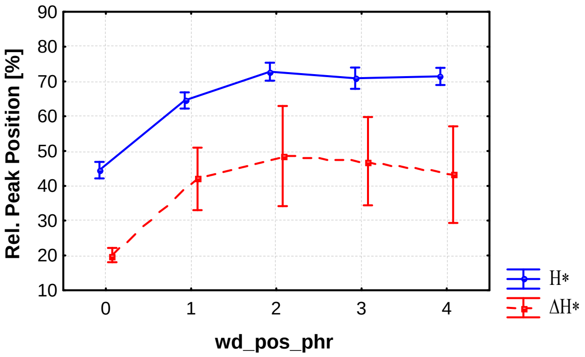


Figure 4: The effect of foot position (regarded as a distance from phrase boundary, i.e. 0 is phrase final) on relative peak position depending on the accent type H* or ΔH*. Mean peak position for phrase final static accented syllables: 43.2% and for dynamic accents: 20.85%.

have falling nuclear melody. Only yes-no questions were marked as representing the interrogative mode

Median pitch peak position in syllables in different types of prosodic phrases is illustrated in Figure 5. It can be seen that pitch peaks are located much earlier in exclamations compared to other phrase types. In statements peaks occur somewhere in the middle of the accented syllable; in questions and minor intonation phrases they are located in the second half of the syllable.

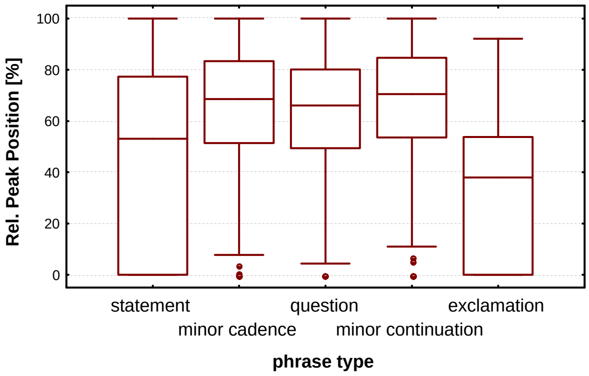


Figure 5: The effect of the phrase type on relative peak position: statement (median: 53.05%), minor cadence (median: 64.24%), question (median: 66.02%), minor continuation (median: 70.5%) and exclamation (median: 36.54%).

Another factor related to phrase structure examined in the study was the number of words in the phrase. This factor appeared to have an opposite effect on peak alignment to the one of syllable number in the word, i.e. with an increasing number of words in the phrase the peak shifts towards the start of the syllable. In single-word phrases the mean peak position is 68.15% (median: 72.71%), in two-word phrases it is 52.67% (median: 60.,89%). But from 3 words up peak location remains fairly constant: the mean position is 53.69% (median: 61.78%) in three-word phrases and 53.55% (median: 62.42%) in phrases consisting of more than three words.

4.2.4. *Tonal environment*

The influence of neighbouring tonal targets on peak location investigated in [8] concerned the number of syllables from/to the preceding/following pitch accent as well as type of the preceding/following tonal target (H vs. L). In the study of F0 peaks in Polish we examined the influence of the preceding/following pitch accent type on peak alignment. A general observation is that the type of preceding pitch accent has smaller impact on peak location than the following pitch accent type.

4.3. Comments

Both for segmental and suprasegmental factors the statistical analyses shows large differences between median and mean values. For the interpretation of the statistical significance of the analysed data further research is needed. It is necessary to a) make more precise analyses for each individual factor, b) to examine the interactions between the factors and c) to carry out a multivariate analysis.

5. Analysis: German

5.1. Factors relating to syllable structure

5.1.1. Effects of onset and coda class

Similar to the Polish results (see section 4.1.1) peak placement is significantly influenced by onset and coda type in German as well. These types are again defined according to the van Santen & Hirschberg classification [16]: -V (voiceless obstruent), +V-S (voiced obstruent), +S (sonorant).

With respect to the three onset types (see also Figure 6), the peak is earliest when there is a sonorant in the onset (mean: 32.8% of syllable duration) and latest when a voiceless obstruent forms the onset (mean: 42.0%). Peaks are generally located in-between (mean: 37.0%), if the onset consists of a voiced obstruent.

For the different coda types (see Figure 7) there is a significant difference of peak position between sonorant codas (mean: 41.7% of syllable duration) and codas solely made up of obstruents (mean 28.5% for voiced obstruents; 27.7% for voiceless obstruents). The peak thus occurs clearly later when a sonorant coda is present, whereas there is no significant difference between the two obstruent classes.

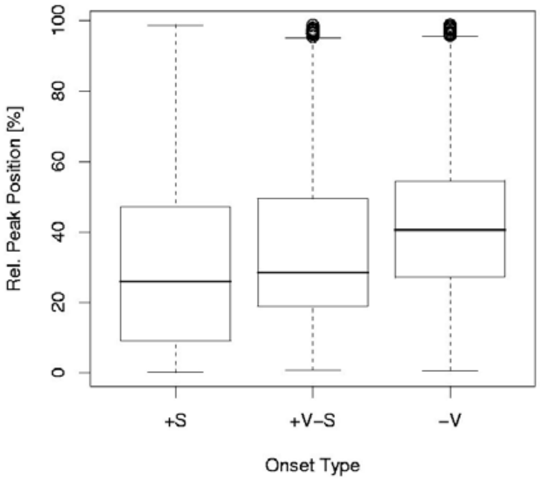


Figure 6: Boxplot showing relative peak position depending on onset type: sonorant (+S, median: 25.95%), voiced obstruent (+V-S, median: 28.4%), voiceless obstruent (-V, median: 40.5%)

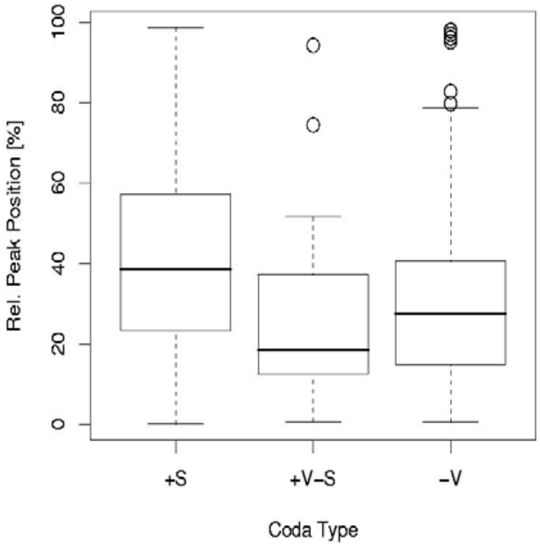


Figure 7: Boxplot showing relative peak position depending on coda type: sonorant (+S, median: 38.4%), voiced obstruent (+V-S, median: 18.45%), voiceless obstruent (-V, median: 27.7%)

If onset type varies but coda type remains unchanged (sonorant), a significant movement of the peak can be observed (F [2, 2667] = 31.526, p < 0.001), as the peak occurs successively later when the onset is a sonorant (36.23% of syllable duration), a voiced obstruent (39.23%), or a voiceless obstruent (44.97%). Variation of coda type (sonorant vs. voiceless obstruent) has a less distinct effect on the peak locations. Most peaks apparently occur in the same locations. However, the greater frequency of late peaks near the syllable boundary in sonorant codas leads to a significantly later mean value (F [2, 2667] = 65.005, p < 0.001) for sonorant (36.23%) vs. obstruent codas (25.15%).

5.1.2. Two types of sonorant coda

In the Festival feature functions' classification of syllable structure types, coda type +S covers both closed syllables with actual sonorant coda consonants and open syllables. Differences between the two types must therefore be expected.

Examining the absolute interval between syllable start and peak location, it turns out that there is virtually no difference between open and closed (+S) syllables. For the former, peak location is on average 97.59 ms after the beginning of the syllable compared to 97.24 ms for the latter. Peak position is thus stable in terms of absolute timing in this very specific context. As syllables with actual sonorant codas can be expected to be longer despite possible compensatory effects concerning vowel length (vowels in accented open syllables are unlikely to be short), this has the consequence that, in relation to syllable duration, peaks occur later in open syllables. Indeed, in open syllables the mean value for peak location is 45.87% compared to 37.98 % for syllables with one sonorant coda consonant and 35.08% for syllables with two coda consonants. If the coda consists of only one obstruent, the peak occurs even earlier, at 30.83%. Figure 8 attempts to visualize these results.

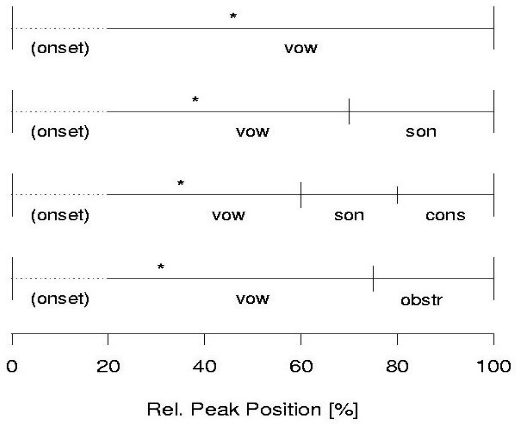


Figure 8: Relative peak position (*) for different coda compositions: open syllable (top row), coda with one sonorant consonant, coda with two consonants, coda with one obstruent consonant (bottom row).

5.2. Influence of the phrasal and tonal environment

The alignment of F_0 peaks is known to be affected by the proximity of other tonal events (pitch accents or phrase boundaries) which may lead to effects of tonal repulsion (e.g., [14]).

5.2.1. Position of the accented syllable in the phrase

A first interesting issue regarding the position of the H*L pitch accent in the intonation phrase is its distance to the preceding or following phrase boundary in number of accents. This amounts to the question whether it is significant if the pitch accent is the first, second, third, next-to-last, last etc. accent of the phrase. As determined by an analysis of variance this distance is shown to be significant (F [1, 2668] = 475.87, p < 0.001) when looking in the direction of the final phrase boundary. This is not the case with respect to the initial phrase boundary (F [1, 2668] = 0.2437, p = 0.6216).

Considering this result it is especially interesting to see whether pitch accents in the extreme positions of the phrase, i.e. the first or last (nuclear) accent, behave accordingly.

Indeed we find that peak alignment in the final pitch accent of the intonation occurs significantly earlier (F [1, 2668] = 21.591, p < 0.001) than in those accents which are not final (mean: 38.5% of syllable duration vs. 53.4% of syllable duration). Peak alignment in the first pitch accent of the phrase is on the other hand not significantly different (F [1, 2668] = 2.5009, p = 0.1139) from that of the other H*L pitch accents in the phrase.

The early alignment in nuclear pitch accents as opposed to non-final accents raises the question whether this is an effect that is facilitated by those instances that occur in the final syllable of the phrase and are thus being pushed forward by the following boundary tone. The comparison of H*L pitch accents in the phrase-final syllable with all other H*L pitch accents does in fact show a significant difference (F [1, 2668] = 496.56, p < 0.001). The peaks of final syllable accents are aligned quite early in the syllable (mean: 21.1% of syllable duration vs. 44.0%). The difference remains significant (F [1, 2668] = 104.98, p < 0.001) also when the nuclear phrase-final pitch accents are compared to nuclear pitch accents that are not in the phrase-final syllable (mean: 21.1% of syllable duration vs. 41.7%). In accordance with these results it is not surprising that similarly to a pitch accent's distance to the next phrase boundary in number of accents, its distance in number of syllables is also significant (F [1, 2668] = 420.01, p < 0.001), and that, correspondingly, there are no significant results with regard to distance to the preceding phrase boundary (F [1, 2668] = 0.0194, p = 0.8892).

5.2.2. Influence of neighboring tonal targets

If boundaries can affect peak alignment, then it is reasonable to assume that other neighboring tonal events, i.e., pitch accents, will do so as well. In a comprehensive analysis of the influence of such adjacent tonal events three major questions are of interest, namely whether the adjacent target is high or low, whether it is preceding or following and how far in terms of number of syllables it is away from the examined pitch accent.

High (H) or low (L) targets are defined by the target point closest to the examined pitch accent, a preceding L*H pitch accent would thus be registered as H, a following one as L.

A first analysis shows that peak alignment is not influenced by the type of target preceding it (F [1, 2461] = 1.643, p = 0.2000). It is, however, of weak significance whether a H or L target follows (F [1, 2665] = 6.0593, p < 0.05; mean peak position when H target follows: 36.4% of syllable duration vs. mean peak position when L target follows: 39.3% of syllable duration).

Alignment occurs significantly earlier (F [1, 2665] = 80.584, p = 0.001) when the next tonal target follows immediately in the next syllable (mean peak position: 33.6% of syllable duration vs. 42.2% of syllable duration).

This result is confirmed when comparing the influence of H and L targets at a distance of either one or two syllables from the accented syllable. Here, peak alignment is significantly different for all four possibilities (mean H following after 1 syllable: 26.1%; mean H following after 2 syllables: 39.3%; mean L following after 1 syllable: 34.5% mean L following after 2 syllables: 43.2%). In this case there is thus also a difference depending on whether a high or low target is following (see also Figure 9).

6. Conclusions

Analyses presented in this paper confirmed results of previous studies (e.g. the influence of factors related to syllable and feet structure, tonal environment) and revealed more factors that play significant role in peak alignment such as prosodic phrase structure and type, phoneme number in the onset and coda.

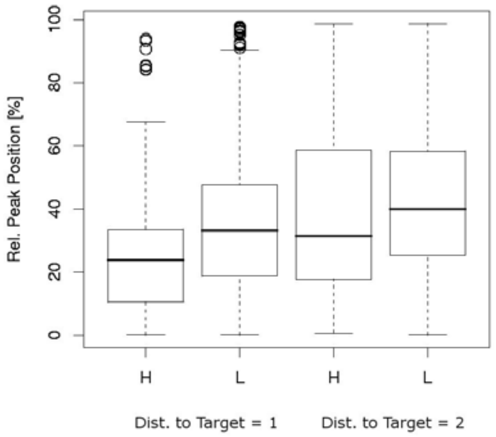


Figure 9: Boxplot showing relative peak position depending on distance and type of following target:H target after 1 syllable (median: 23.8%), L target after 1 syllable (median: 33.2%), H target after 2 syllables

(median: 31.4%), L target after 2 syllables median: 39.9%).

The comparison of peak alignment in Polish and German has shown that a) peaks occur generally later in Polish than in German, b) considering the effects of onset/coda type the same tendency can be observed in the two languages, c) in Polish stronger effect of following tonal target can be observed, d) phrase-final accent has special status in both languages (Polish: final word position, German: nuclear accent).

With respect to speech synthesis it can be added that even very general measurements such as the most frequent peak location in all H*L pitch accents of the corpus may have their use as defaults to fall back on, should more complex rules not apply. In fact, for unit selection the procedure offers the possibility of adapting to the potential prosodic idiosyncrasies of the individual speaker who provides the voice.

The general analysis of all labeled H*L accents must also disregard the fact that timing differences can either be phonetic or phonological in nature. As a consequence, differences in peak alignment that are not caused by the segmental and/or prosodic environment but are actually the expression of a different communicative function (as shown in [10] for early, medial, or late peaks in German) are not captured. From the point of view of speech synthesis, this problem is not too pressing as the prediction of such differences in meaning is not yet possible anyway. Also, this kind of phonological variation is arguably less likely to occur in a corpus that mainly contains readings from newspaper articles. Similarly, the method of detecting peaks may have to be refined, if the current approach is extended to other types of accents which are more likely to have peaks outside of the accented syllable. It may be emphasized again that the approach taken in this study has the advantage of allowing for the effective analysis of large amounts of data. It does, however, not create a controlled environment in which influences from parameters other than the one investigated are excluded. This apparent disadvantage can be dealt with by targeting interactions between specific parameters.

7. Acknowledgements

This research has been realized within the framework of the Project No. R00 035 02 “Technologies of processing and recognizing verbal information in the systems of internal security” financed by the Polish Ministry of Science and Higher Education. It has also been supported by a Alexander von Humboldt Polish Honorary Research Fellowship awarded to the fourth author.

8. References

[1] Black, A. W., Taylor, P. and Caley, R., “The Festival Speech Synthesis System – System documentation”, CSTR Edinburgh, 1999 [http://www.cstr.ed.ac.uk/projects/festival/manual/].

[2] Breuer, S., Stober, K., Wagner, P., Abresch, J., “Bonn Open Synthesis System BOSS“.[http://www.ikp.uni-bonn.de/boss/].

[3] Cruttenden, A., “ Intonation“, Cambridge University Press, 1986

[4] Demenko, G. and Wagner, A., “The stylization of intonation contours”. Proceedings of Speech Prosody 2006, Dresden 2006

[5] Grice, M., Baumann, S., Benzmüller, R., “German Intontion in Autosegmental-Metrical Phonology”. In Jun, Sun-Ah (ed.) Prosodic Typology: The Phonology of Intonation and Phrasing. Oxford University Press 2005

[6] Hirst, D. and Espesser, R., “Automatic modelling of fundamental frequency using a quadratic spline function”. Travaux de l'Institut de Phonétique d'Aix 15, 71-85

[7] “IMS German Festival homepage,” [http://www.ims.uni-stuttgart.de/phonetik/synthesis/index.html].

[8] Jilka, M. and Möbius, B., “Towards a comprehensive investigation of factors relevant to peak alignment using a unit selection corpus”. Proceedings of Interspeech/ICSLP 2006 (Pittsburgh, PA), 2054-2057.

[9] Klabbers, E. and Van Santen, J., “Clustering of foot-based pitch contours in expressive speech”. ISCA Speech Synthesis Workshop 5, Pittsburgh, PA, 2004

[10] Kohler, K. “Macro and micro F_0 in the synthesis of intonation”. In Kingston, J. and Beckman, M. (eds.), Papers in Laboratory Phonology I. CUP, Cambridge, 115-138, 1990.

[11] Mayer, J., *Transcription of German Intonation – The Stuttgart System*, Technical Report, University of Stuttgart, 1995 [http://www.ims.uni-stuttgart.de/phonetik/joerg/labman/STGTsystem.html].

[12] Taylor, P., “Analysis and synthesis of intonation using the Tilt model”. J. Acoust. Soc. Am. 107 (3), 1697–1714

[13] Schweitzer, A., Braunschweiler, N., Dogil, G., Möbius, B. 2004. Assessing the Acceptability of the SmartKom Speech Synthesis Voice. *Proceedings of the 5th ISCA Speech Synthesis Workshop*, 1-6.

[14] Silverman, K. and Pierrehumbert, J. “The timing of prenuclear high accents in English”. In Kingston, J. and Beckman, M. (eds). Papers in Laboratory Phonology I. Cambridge: CUP, pp. 72-106, 1990

[15] Steffen-Batóg, M., “Accentual structure of the Polish language”. Polish Scientific Publishers (PWN) 2000.

[16] Van Santen, J. and Hirschberg, J. “Segmental effects on timing and height of pitch contours”. Proceedings of *ICSLP Proc.*, 719-722, 1994.

[17] Van Santen, J. and Möbius, B. “A Quantitative Model of F0 Generation and Alignment”. In Botinis, A. (ed.), Intonation – Analysis, Modelling and Technology, Kluwer, Dordrecht, 269-288, 2000.

[18] Van Santen, J., Möbius, B., Venditti, J., Shih, C., “Description of the Bell Labs intonation system”. Proceedings of Third ESCA Workshop on Speech Synthesis,1998.

Optimization of Polish Segmental Duration Prediction with CART

Katarzyna Klessa, Marcin Szymański**, Stefan Breuer***, Grażyna Demenko**

Institute of Linguistics, Dept. of Phonetics
Adam Mickiewicz University, Poznań, Poland*
Poznań University of Technology, Poland**
Institute of Communication Sciences
University of Bonn, Germany***
katarzyna.klessa@amu.edu.pl, marcin.szymanski@cs.put.poznan.pl,
breuer@ifk.uni-bonn.de, lin@amu.edu.pl

Abstract

This paper describes results of the investigation of Polish segmental duration for the purpose of speech synthesis. The experiment is a continuation of the previous work of the same authors [1] aiming at improving the outcome of the duration prediction mechanism to enhance the overall quality of synthesized speech.

1.Introduction

Duration prediction models for speech synthesis range from the more traditional, rule-based techniques to trainable, corpus-based techniques. Nowadays, it is often the case that the two approaches overlap and careful linguistic feature extraction usually is an important stage preceding the actual statistical processing [2, 3, 4, 5, 6] The unit that should be regarded as the base for segmental duration modeling is also a subject of discussion. Most frequently, phone is used as the unit, however there are also other proposals e.g. Campbell's syllable-based model [7]. Linguistic knowledge may be used not only in the data preparation process but also in the modeling process itself which is postulated and tested for various languages by Van Santen's sum-of-product models e.g. [8, 9, 10]. In the present study, phones were the base units for prediction, though the influence of other units was considered. In our experiment, the influence of two sets of factors on phone duration was investigated. Then we compared the present results with the ones we obtained from similar tests done with a smaller corpus of data [1]. Both experiments were performed using BOSS technology [11].

2. Corpora and annotation procedure

For the most recent analyses, we used a corpus of two hours of continuous speech read by a male professional speaker. The results were then compared to those obtained from a corpus of almost 50 minutes of speech produced by the same speaker [1].

- The 2-hour corpus contains utterances prepared especially for the database to provide coverage for the most frequent Polish triphones (most importantly CVC triphones in the contexts of sonorants), as well as the most frequent diphones, and consonant clusters and the 600 most common Polish words.
- The texts in the 50-minute database consist of fragments of prose, newspaper articles, short dialogs, (some of them having a potential emotional load), and a list of railway enquiry entries.

Both corpora were annotated according to the same labeling procedure. First, the recordings were labeled automatically with “Salian” [12]. (the segmentation accuracy of the software is 10 [ms]). In the second step, the labels were corrected manually based on visual inspection of spectrograms. Both corpora were revised by the same group of experts following the same guidelines. The 50-minutes corpus was corrected mainly using Wavesurfer, and for the other corpus “Annotation Editor”, a tool designed specifically for the purposes of the speech synthesis project for the Polish language (see acknowledgments) was used. SAMPA for Polish was used as the transcription alphabet with the following modifications:

- Palatalized variants of [k][g] were added to the label set and marked with: [c][J]
- Labels for the Polish nasalized vowels i. e. [e~][o~] were removed and instead sequences of vowels and nasalized consonants [w] or [j] marked with [w~][j~] were used

The second modification will be subject of further investigation in the future as there are reasons to suspect that better results for synthesis could be obtained by connecting the oral and nasal component into compound items [13]. Syllable boundaries were inserted automatically as well as word stress labels. Word stress was assigned to the penultimate syllable, which is its most common position in Polish. Afterwards, the actual placement of stress was manually verified and corrected, if needed. Phrase boundaries were established according to linguistic cues and then verified on the basis of perceptual evaluation of intonation contours, intensity and pauses.

3. Features for training CART

Initially, the list of features for duration prediction included the following information:

- Sound (which particular phone is the phone in question)
- Sound's properties. The following features were included as sound properties: the manner of articulation, the place of articulation, the presence of voice, the type of sound (consonant or vowel)
- Properties of the preceding and of the following context. The properties were exactly the same as those listed above as the properties of the sound in question. A 7-element frame was used as the context information, i. e. the same properties were used as features for three preceding and for three following phones as well as for the phone in question.
- Position within the higher unit of speech organization structure. Feature space included: position of the phone in question relative to pause; the distance of the syllable containing the phone to the left and right word boundary; the position of the syllable within the foot (in the anacrusis, head or tail of the foot), the position of the foot within the intonation phrase. Position of the sound within syllable structure (onset, nucleus, coda).
- Identical neighborhood. The information if the phone in question occurred in the neighborhood of an identical phone in the directly preceding or following context within the same word.
- The position of sound relative to consonant clusters (within cluster, before or after cluster or with no cluster in the direct neighborhood).
- Word length and foot length.
- Word stress and sentence stress.

4. Results

To obtain our results, we used the CART implementation “wagon” [14]. The set of features corresponding to the properties listed in the previous paragraph was used to predict segmental duration with the 50-minute database and with the 2-hour database. The results are shown in the first two columns of Table 1. Each time, the results were obtained with 5 % held out data.

Table 1: Comparison of CART results for two corpora

50-min Corpus	2-h Corpus	2-h Corpus & Features Modified
RMSE 19.1973 **Correlation**: 0.7284 Mean (abs) **Error** 14.1092 (13.0182).	**RMSE** 15.5178 **Correlation** : 0.8047 Mean (abs) **Error** 11.4132 (10.5139)	**RMSE** 15.4010 **Correlation**: 0.8080 Mean (abs) **Error** 11.3451 (10.4154)

As can be seen, the results for the 2-hour corpus are significantly better. The difference in RMSE (the root mean squared error) appeared to be more than 4 milliseconds, the overall mean correlation also improved from almost 0.73 to 0.8. In search for further improvement of the prediction the list of features was extended by the following new items:

- The same or different place of articulation of the phone in question and the phone in its direct left or right context
- The same place of articulation across word boundary. The information if the phone in question occurred in a neighborhood of a phone with the same place of articulation across word boundary
- Syllable length, phrase length, and the length of the whole source utterance

After including this information into the feature space, there was another improvement of the results, yet it was less substantial than the one obtained after switching to a bigger and better controlled database in terms of the coverage of phonetic-acoustic properties of the read speech.

The next step was to check how the contribution of particular features to the overall result obtained by the whole feature set was influenced after adding new features. In order to get the information, we used the stepwise option of “wagon” [14]. With the stepwise option enabled, results are expressed as cumulative correlation.

The number of features in the two experiments differed by six items (51 features in the initial test, and 57 features after adding the new features). As it appeared, the order of features in order of their contribution was very similar in both of the two experiments as far as the most contributing features are concerned. The first difference was observed on the 12th position in the ranking. In the experiment with the modified feature list the 12th feature was one of the new features: the length of the utterance. As for the other new features, the phrase length was on the 16th position, the syllable length on the 18th position, and the information of the same or different place of articulation across word boundaries was on the next position, immediately followed by the feature “same or different place of articulation” for the right context.

Table 2 shows the 15 most contributive features in the test performed with various number of features. In the first two columns the results for the two-hour database are presented for experiment with the two different feature lists. The last column shows the similar results for the 50-minute database that were obtained using the shorter feature list.

Despite the differences between the corpora and various number of features taken into account four features out of the first fifteen in the rankings appear exactly on the same position in each of the three sets (marked in bold).

Another observation is the that in the results for the 50-minute database, the first feature in the ranking is the “phone in question”, which is consistent with the results reported for many similar studies for various languages e. g. [6, 15, 16]. The second feature in terms of importance was the immediate right context. For the two-hour database the order appeared to be reverse: the phone in question was placed on the second position just behind the “right context” feature. First, we thought one of the possible reasons might be the fact that Polish diphthongs were treated as two separate units in the 2-hour database while in the 50-minute database they were regarded as compounds. In order to check if that was the case an additional test using stepwise option of “wagon” was run. This time, two parts of diphthongs were again joined into compound units, however in the resulting order of features the right context was still more contributive than the phone in question.

Table 2: Feature ranking comparison (stepwise) – the most important features, cumulative correlation.

2-h corpus features modified (Dataset of 98835 vectors of 57)	2-h corpus (Dataset of 98835 vectors of 51)	50-min corpus (Dataset of 98835 vectors of 51)
1. Right context: 0.5062	1. Right context: 0.5062	1. Phone in question: 0.4774
2. Phone in question: 0.7004	2. Phone in question: 0.7004	2. Right context: 0.6396
3. Left context: 0.7375	**3. Left context: 0.7375**	**3. Left context 0.6625**
4. **Foot distance to the right phrase boundary: 0.7613**	4. **Foot distance to the right phrase boundary: 0.7613**	**4. Foot distance to the right phrase boundary 0.6793**
5. Articulation manner in the 3rd right context: 0.7703	5. Articulation manner of the 3rd right context: 0.7703	5. Articulation manner in the left context 0.6859
6. Articulation manner in the left context: 0.7749	6. Articulation manner in the left context: 0.7749	6. * Syllable position within the foot 0.6919
7. Nuclear stress: 0.7787	7. Nuclear stress: 0.7787	7. * Articulation place (the phone in question) 0.6959
8. Presence of voice (the phone in question): 0.7846	**8. Presence of voice (the phone in question): 0.7846**	**8. Presence of voice (the phone in question) 0.7033**
9. Articulation manner in the right context: 0.7893	**9. Articulation manner in the right context: 0.7893**	**9. Articulation manner in the right context 0.7098**
10. Articulation manner (the phone in question): 0.7934	10. Articulation manner (the phone in question): 0.7934	10. Articulation manner in the 3rd right context: 0.7146
11. Articulation manner of the 2nd left context: 0.7963	11. Articulation manner of the 2nd left context: 0.7963	11. Articulation place in the right context 0.7168
12. Utterance length: 07893	12. Articulation manner of the 2nd right context: 0.7982	12. Syllable distance from the beginning of the word: 0.7189
13. Presence of voice in the left context: 0.8003	13. Syllable distance from the beginning of the word: 0.7995	13. Articulation manner in the 2nd left context: 0.7208
14. Word length: 0.8025	14. Sound type in the right context: 0.8003	14. Articulation place in the 2nd right context 0.7219
15. Articulation manner of the 2nd right context: 0.8038	15. Presence of voice in the left context: 0.8011	15. Phone position in the syllable onset, nucleus or coda: 0.7231

It should be noticed that in the classification and regression tree the top-most rule generated for the 2-hour database was "CRIGHT is sil" (i.e. right context is silence). Due to the fact that one of the sub-bases of the 2-hour database consisted of short, 3-4-word phrases, the stronger influence of the following context might possibly be explained by prepausal lengthening effect.

Most features in the first fifteen positions occur in all three tests with two exceptions: the features "syllable position within the foot" and "Articulation place (the phone in question) " appear on the 6th and 7th position in the test for the 50-minute corpus, these two are marked with a star.

Additionally, two more experiments were performed. First, the "new" set of features was used to test a corpus composed of all data, i.e. both the 50-minute and the 2-hour database. The results appeared to be slightly better than for the 50-minute database but worse than those for the 2-hour database. The numbers were as follows:

- no heldout: RMSE 16.3432, Correlation 0.7878, Mean (absolute) Error 11.5945 (11.5182)
- 5% heldout data: RMSE 16.3977 Correlation is 0.7862 Mean (abs) Error 11.6520 (11.5377).

For the second experiment, fifty minutes of recordings were randomly selected from the 2-hour database to compare the outcome of the two corpora using a similar amount of speech data. The results were characterized by the following values of correlation and errors:

- no heldout: RMSE 15.9093 Correlation is 0.7929

  Mean (abs) Error 11.7416 (10.7352)
- 5% heldout data: RMSE 15.9644 Correlation is 0.7912 Mean (abs) Error 11.7652 (10.7909)

The above values are an improvement as compared both to the 50-minute database and to the results obtained for the combined databases, however they are slightly worse than those for the 2-hour database. The latter observation seems to be an obvious effect of enlarging the speech corpus. The deterioration of the results after combining both corpora might be due to differences between the types of texts recorded in the two databases. The speaker tended to accelerate while reading longer texts as compared to short separate phrases even though he was supposed to keep the same rate for all the recordings. However this relation requires further investigation.

5. Conclusions

Correlation and RMSE improved substantially when we used the larger corpus containing sentences prepared to cover the most frequent clusters, diphones and triphones as the training data set. The modification of the list of features provided a further (slight) improvement of the results. The comparison of the results obtained with the two feature sets (Table 2) shows that the first difference in the order of contribution appears on the 12th position (per 51 or 57 items), the first eleven items are ordered identically in the two tests.

This stability of feature order, together with the improvements in correlation and RMSE suggests that our choice of features comprises the chief linguistic and phonetic determinants of segmental duration. The problem that needs further examination is the fact that in the recent tests the feature "phone in question" was the second most contributive feature and not the first one which seems to be the obvious expected result and was always the case in the previous experiments of the authors as well as reported in other studies. The values of correlation and the RMSE after the modification of the corpus and the list of features for duration prediction provided comparably good results. The next step in order to obtain further improvement should be a closer investigation into the cost function in the unit selection algorithm, which will be performed in the near future, as well as more detailed analyses of the statistical relevance of the results obtained for correlation and RMSE

6. Acknowledgements

This paper was supported by Alexander von Humboldt Foundation and by Polish Scientific Committee (KBN Project no. 2 H01D 003 24) .

7. References

[1] Breuer, S., Francuzik, K., Demenko, G., Szymański, M. *Analysis of Polish Duration with CART*, Proceedings of Speech Prosody, Dresden, 2006.

[2] Klatt, D. H. Linguistic uses of segmental duration in English; Acoustic and perceptual evidence. JASA 59 (5), 1976, pp. 1208 - 1221

[3] Olaszy, G., Predicting Hungarian Sound Durations for Continous Speech. Acta Linguistica Hungarica, Budapest, vol. 49 (3-4), 2002, str. 321-345.

[4] Riedi, M.P. Controlling segmental duration in speech synthesis systems. PhD thesis, TIK-Schriftenreihe (26), ETH Zürich, 1998.

[5] Vainio, M. Altosaar, T; Karlajainen, M; Aulanko, R; Werner, S. Neural network models for Finnish prosody. Proceedings of ICPhS'99, California, 1999.

[6] Batusek, R. A Duration Model for Czech Text-to-Speech Synthesis, Proceedings of Speech Prosody, Aix-en Provence, 2002.

[7] Campbell, N., 1992. Multi-level timing in speech University of Sussex . PhD Thesis. (Exp. Psychol): Brighton, UK.

[8] V. Santen, J.P.H., Quantitative Modeling of Segmental Duration. [in:] Proceedings of Human Language Technology Conference, Princeton, New Jersey, 1993, str. 323-328.

[9] V. Son, R.J.J.H., V. Santen, J.P.H., Strong interaction between factors influencing consonant duration. [in:] Proceedings of Eurospeech '97, Rhodos, 1997.

[10] Moebius, B., van Santen, J.P.H. Modeling segmental duration in German text-to-speech synthesis, Proceedings of the International Conference on Spoken Language Processing (Philadelphia, PA). (4), 1996, pp. 2395-2398.

[11] Breuer, S., Wagner, P., Abresch, J., Bröggelwirth, J., Rohde, H., Stöber, K., Bonn Open Synthesis System (BOSS) 3. Documentation and User Manual. http://www.ikp.uni-bonn.de/boss/BOSS_Documentation.pdf 2005.

[12] Szymański M. and Grocholewski S., Transcription-based automatic segmentation of speech. [in:] Proceedings of 2nd Language & Technology Conference, Poznań, 2005, pp. 11-15.

[13] Demenko, G., Wypych, M., Baranowska, E. Implementation of Grapheme to Phoneme Rule and Extended Sampa Alphabet In Polish Text-to-Speech Synthesis. Speech and Language Technology (7), pp. 79-95. Poznań, 2003.

[14] King, S., Black, A.W., Taylor, P., Caley, R., Clark, R., Edinburgh Speech Tools. System Documentation Edition 1.2, for 1.2.3 24th Jan 2003.

[15] Krishna, N.S., Murthy, H.A., Duration Modeling of Indian Languages Hindi and Telugu. [in:] Proceedings of 5th ISCA Speech Synthesis Workshop, 2004.

[16] Chung, H., Huckvale, M., Linguistic factors affecting timing in Korean with application to speech synthesis. [in:] Proceedings of Eurospeech 2001, Scandinavia. http://www.tsi.enst.fr/~chollet/Biblio/Congres/Audio/Eurospeech01/CDROM/papers/page815.pdf

Utilization of an HMM-Based Feature Generation Module in 5 ms Segment Concatenative Speech Synthesis

Toshio Hirai† *Junichi Yamagishi*‡ *Seiichi Tenpaku*†

† Arcadia, Inc., Japan
‡ The Centre for Speech Technology Research, University of Edinburgh, UK
thirai@arcadia.co.jp

Abstract

If a concatenative speech synthesis system uses more short speech segments, it increases the potential to generate natural speech because the concatenation variation becomes greater. Recently, a synthesis approach was proposed in which very short (5 ms) segments are used. In this paper, an implementation of an HMM-based feature generation module into a very short segment concatenative synthesis system that has the advantage of modularity and a synthesis experiment are described.

1. Introduction

Speech synthesis is a technique for converting text into speech. Currently, concatenative speech synthesis systems are entering the mainstream because they can achieve high quality speech without great difficulty. In a concatenative speech synthesis system, an amount of recorded speech samples and their features are stored in a database ("corpus.") At synthesis, an appropriate speech segment sequence is selected from the corpus and concatenated smoothly. The selection is executed according to the feature time series (target) that is generated from input text. A concatenative speech synthesis system increases its potential to generate natural speech if the system uses shorter speech segments, because the concatenation variation becomes greater. Recently, a synthesis approach was proposed in which very short segments (5 ms) were used [1, 2, 3, 4].

We have been trying to improve the naturalness of the synthesized speech. At the same time, we sought the feature generation module since our system lacked the module [1]. Another 5 ms segment synthesis method proposed by Ling et al. uses HTS (HMM-based Triple S (speech synthesis system)) [5] as a feature generation module [2]. HTS has the ability to generate a time series of vectors (mel-cepstrum or "mcep") from Linguistic-/Prosodic-Information(LPI) that is produced from input text. In our system, HTS is also adopted as the feature generation module.

The method proposed by Ling et al. requires the time series of the mean and variance of features as the target value for synthesis. Since it is difficult to get paired information which guarantees "natural" synthesized speech, it is also difficult to isolate the cause (a problem of the feature generation module *or* of the feature-to-speech module) when the synthesized speech is not natural enough. On the other hand, in our method, only the mean value is required as the target to synthesize. It would be an advantage since if the feature (which corresponds to the mean value and must have the information to synthesize natural speech) extracted from natural recorded speech is used as the target in our method but the synthesized speech is not natural enough, it can be said that the cause rests with the feature-to-speech module.

This paper is organized as follows. The next section (Section 2) introduces the processing outline in the 5 ms segment concatenative speech synthesis including the utilization of the feature generation module in HTS. The following two sections (Sections 3 and 4) present a synthesis experiment and its results, in which 450 Japanese utterances were used. Section 5 discusses the findings in the experiment, and Section 6 summarizes this paper.

2. Concatenative speech synthesis using 5 ms segments with an HMM-based feature generation module

2.1. Analysis stage

2.1.1. Corpus construction

Speech data are analyzed every 5 ms, and extracted features are stored in a speech corpus during the analysis stage for corpus construction. The extracted features are the speech fundamental frequency (F_0), power, and spectrum. As the spectral information, mcep is adopted. These features are used to get the target cost in the synthesis stage. To construct a corpus automatically and to avoid contamination of the F_0 extraction errors, it is not the scalar F_0 value (as in conventional speech synthesis systems including Ling's method [2]) but the lower frequency part of a power spectrum (the upper frequency bound is set by the highest F_0 of the speech data) which is treated as the F_0 information in our method. Hereafter, the lower frequency part of a power spectrum is also denoted as "F_0 information." **Figure 1** shows an example of the F_0 information. As can be seen from the figure, the "ridge" of the power spectrum contour corresponds to the scalar F_0 pattern ('$+$') very well.

In the synthesis stage, FFT-based power spectrum is used for the calculation of concatenation distortion at the concatenation point. Therefore, FFT analysis centered at the ends of 5 ms segments is also executed. The processing flow is illustrated in **Figure 2**.

2.1.2. Feature generation module building

In HTS, the relationship between LPI and features (F_0 and mcep) is analyzed in order to construct a feature generation model. In our method, it is required to generate F_0 information by the model. In this report, the F_0 information is merged into the mcep and analyzed as one stream in model training for simplicity.

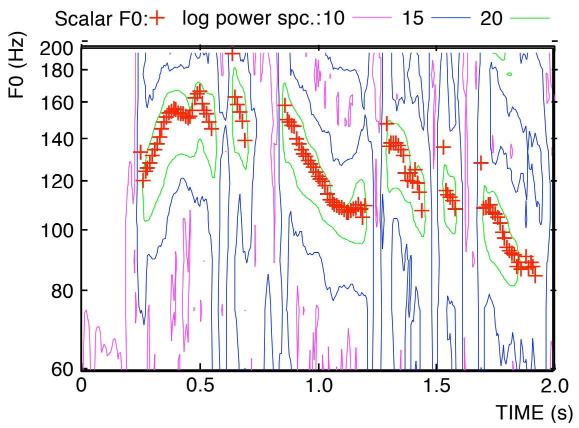


Figure 1: *Contour of the lower frequency part of power spectrum with scalar F_0.*
Initial part of sentence J21 in ATR503:
“nyu:gakushIkeNo ukeru tokiyori (than
an entrance examination)”

2.2. Synthesis stage

2.2.1. Feature generation

In the synthesis stage, input text is used to generate the feature time series by HMMs trained in the analysis stage as target vectors that are used to synthesize the required speech sound. The generated feature is decomposed into F_0 information, power, and mcep.

2.2.2. Feature-to-speech processing

Speech segments similar to the generated feature are searched for in each segment in the speech corpus using a target cost function, and the N-best segments are selected as candidates in each frame. Next, all combinations of the candidate connections are evaluated, and the segment sequence exhibiting the lowest connection distortion per concatenation point is concatenated in order to generate synthesized speech.

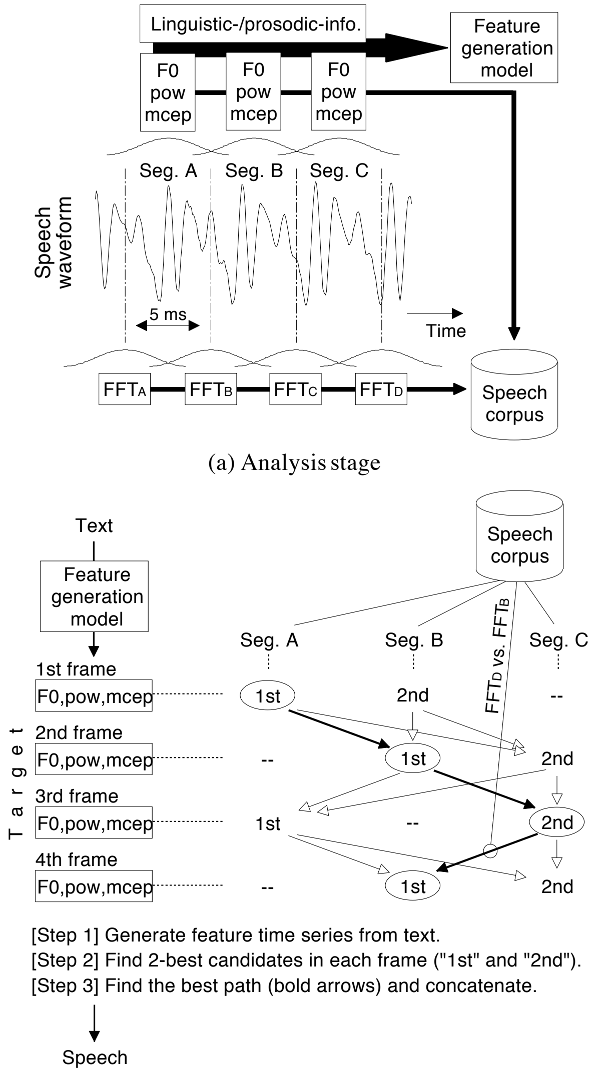
3. Experiment

3.1. Speech material

We used a phonetically balanced Japanese speech database (“ATR 503 sentences [6],” ATR503) spoken by a male speaker that is attached to HTS [5]. Speech data in the database were sampled at 16 kHz and quantized with 16 bits. For corpus construction and feature generation module building by HTS in the analysis stage, we used the 450 utterances of the database (the groups A, B, ..., and I). The total number of 5 ms segments was 481,207.

3.2. Analysis conditions

These are the segment analysis conditions: frame length, 1,024 points (64 ms) for F_0 analysis and 512 points (32 ms) for power and mcep analysis; Hanning windowing; frame step width, 80 points (5 ms); F_0 analysis, from 1st to 19th orders of the power spectrum (the frequency of the 19th channel is 296.875 Hz ($= 16,000/1,024 \times 19$)); the order and α in mcep analysis, 24 and 0.42. The zero-th term of mcep was not used as the power. Instead, power was calculated from the windowed signal directly. For the extraction of the mcep parameter, we used the “mcep” command in Speech Signal Processing Toolkit [7].



(a) Analysis stage

(b) Synthesis stage

Figure 2: *Processing flowchart of analysis/synthesis stages.*

In paper [1], the lower frequency part of a power spectrum was normalized in each segment in order to eliminate power information by dividing each original value by the summation. However, if the normalized value is used in the HTS training, the generated feature sometimes shows negative values though it should be positive. To avoid this problem, the normalization was not executed in this report. Because the velocity (“Δ”) and the acceleration (“Δ^2”) of the value are considered in parameter generation of HTS [8], the smoothness of them was ensured.

These are the analysis conditions for distortion measurement at the segment edge: Frame length was 256 points (= 16 ms) with 0.97 pre-emphasis and Hanning windowing.

3.3. HTS processing conditions

The version of HTS was 2.0 [5]. From the remaining 53 sentences that were not used for model building, five were chosen for a synthesis experiment. Scalar F_0 information was also included in the training features since the training became unstable without it. The feature sequences were directly

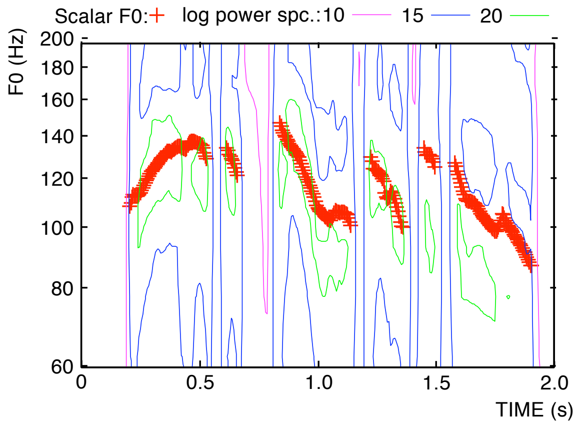


Figure 3: *Parameters generated by HTS.*
Refer to the note of Figure 1.

generated from HMMs that was trained with HTS. The HMMs are 5-state left-to-right context-dependent HMMs, and each state has 2 Gaussian probability density functions. We utilized an EM-based iterative parameter generation algorithm which is detailed in Section 2.3 of [8] (Case 3) and the information is stored in the directory "gen/qst001/ver1/2mix/2" in the HTS system.

3.4. Synthesis conditions

N in N-best was set at 300. This is the procedure to calculate the distance between a target segment and a segment in the corpus: (1) For all features (F_0, power, and mcep), execute (1–1) and (1–2). (1–1) For each target segment, calculate the Euclid distance of the feature from all segments in the corpus. (1–2) Each distance is normalized by the mean and standard deviation of all distances. (2) The summation of weighted previous-/current-/post-positional distances of all features is treated as the definitive distance. The weights for previous-/current-/post-position were 1, 3, and 1.

For the distortion measure at concatenation points, the Kullback-Leibler distance of the FFT-based power spectra [9] with the consideration of the powers was adopted. The Dijkstra's shortest path search algorithm [10] was used for the full path search. Consideration for previous-/post-target distance and concatenation distortion ensures the smoothness of synthesized speech indirectly. Finally, the cross-fade technique for frame sized segment concatenation was used to generate the speech waveform.

4. Results

It took about 7 hours to complete the learning of HMMs by a computer with a 2.4 GHz CPU and 768 MB memory, excluding the feature extraction time from recorded speech. A sample of the generated F_0 information is shown in **Figure 3** with the generated scalar F_0. For 5 ms segment synthesis, it took about 1.6 hours for a sentence. The spectrum, F_0, etc. of the synthesized speech are shown in **Figure 4** with those of the recorded speech. Synthesized speech samples can be listened to at: http://www.arcadia.co.jp/~thirai/ssw6.

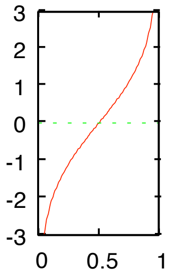


Figure 5: *An example of a finite-infinite transformation (equation (1)).*

5. Discussion

The speech synthesized by the proposed method seems to have better voice quality (speaker's individual reproducibility) than that generated by the original HTS. In 5 ms segment speech synthesis, sometimes a buzz noise appears (e.g. /N/ in "he:kiN" of speech sample J11). It might be caused by the monotonous target value in a part where a segment sequence is used repeatedly in the part in synthesized speech. It would be suppressed by adding a small random noise to the target feature.

The intonation of the speech synthesized by the 5 ms segment synthesis and by the HTS has problems. For example, the intonation pattern of 5 morae accentual phrase "unagiyani (at an eel restaurant)" of sample J01 should be "LHHHH," but the patterns realized in the synthesized speech were "LHHHL." (H: High, L: Low.) It might be caused by the mismatch between recorded speech and the LPI of it in the training data. Therefore, if LPI is corrected according to the recorded speech, the intonation quality of synthesized speech would be improved.

As mentioned in section 3.2, the normalization of the lower frequency part of a power spectrum was not executed in this report. In order to suppress the appearance of the negative value in a generated feature, it would be effective to use a function that transforms finite-domain $[0, 1]$ to infinite-range $(-\infty, \infty)$, such as the logistic transformation:

$$y = \log \frac{x}{1-x}, \qquad (1)$$

where x is one of the normalized value in the F_0 information, and y is a transformed value, which is used in the model training. **Figure 5** shows the part of the function (y range is $[-3, 3]$). In the synthesis stage, its inverse function is used for the transformation from the value generated by HTS to target value for speech synthesis. By the way, it is necessary to study the meaning of such non-linear conversion for speech parameters, and if such pre-processing is appropriate for the analysis in HTS. Not only the transformed value, but also the original value has not been the target for such consideration. For example, the agreement between the Euclid distance of a pair of F_0 information and the auditory perceptual distance in F_0 in these segments has not been studied deeply yet. We have tried to find the optimal distance measure by changing it to another one (correlation between the F_0 information [11]), but clear improvement of synthesized speech quality was not confirmed.

It is known that the speech synthesized from the target feature extracted from recorded speech has a "vibration" sounding problem which appears at /no o/ of "hiQshino omoi" in sample J21. (This phenomenon also appears in the speech synthesized from the HTS generated features, around 2.7 s in Figure 4.) These are possible reasons: the low resolution of

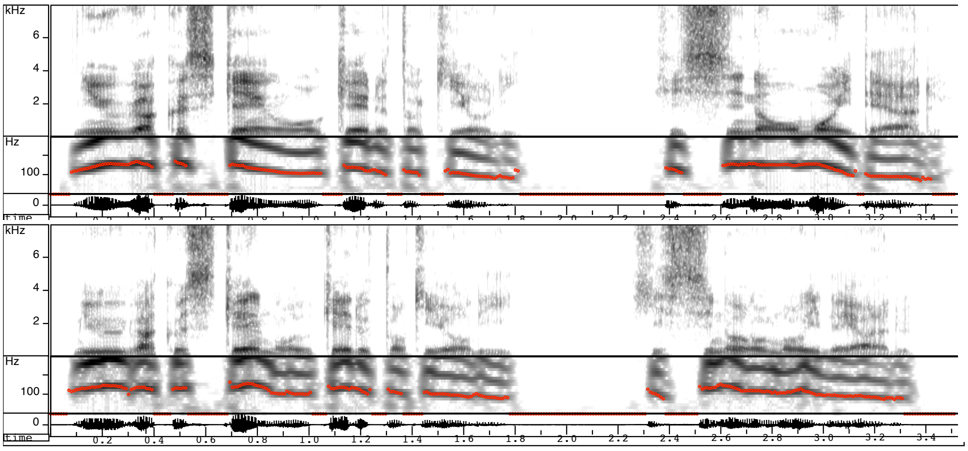


Figure 4: *Synthesized speech (bottom) with original speech (top).*
The utterance is J21: "nyu:gakushIkeNo ukeru tokiyori hiQshino omoide aru (I made more desperate efforts than an entrance examination)." In each panel, the spectrum, enlarged part of narrow band spectrum (0–300 Hz) for F_0 pattern displaying with automatically detected scalar F_0, and waveform are drawn from top to bottom. The unit of time axis is second.

the F_0 information to represent F_0; the limitation of contextual information (currently, only the adjacent segment's distances are considered). It is necessary to investigate if the resolution precision of F_0 affects the speech quality seriously by changing the order of FFT in F_0 analysis.

In the proposed method, the shortest path that exhibits the lowest mean concatenation distortion is searched for exhaustively in the N-best candidates. Such a full search is effective if the distortion measure corresponds to the perceptual measure very well. However, the measure used in this method might not have such correspondence. For this reason, some kind of pruning in the segment search would be effective for improving the naturalness of synthesized speech.

6. Summary

In this paper, we presented a concatenative speech synthesis system in which 5 ms segments are used and an HMM-based feature generation function of HTS is introduced as an LPI-to-feature transformation module. It was confirmed that the reproducibility of the speaker's voice quality was better than that generated by the original HTS.

Since speech synthesized from the extracted features of recorded speech has a vibration sounding problem, the priority for the solution of it should be higher than that of speed-enhancement for synthesis.

7. Acknowledgments

The authors greatly appreciate the discussion with Professor Takayuki Arai (Sophia University, Japan) and Dr. Tomoki Toda (Nara Institute of Science and Technology).

8. References

[1] T. Hirai and S. Tenpaku. Using 5 ms segments in concatenative speech synthesis. In *Proc. 5th ISCA Speech Syntheis Workshop*, June 2004. http://www.ssw5.org/papers/1032.pdf.

[2] Z. Ling and R. Wang. HMM-Based unit selection using frame sized speech segments. In *Proc. ICSLP*, 2006.

[3] T. Hirai. Optimization of target cost weights in concatenative speech synthesis with very short segments of 5-ms duration. In *Proc. 4rd Joint Meeting of ASA and ASJ*, Nov. 2006. Abstract: JASA, Vol. 120, No. 5, Pt. 2, p.3037, 1pSC7.

[4] Z. Ling and R. Wang. HMM-based hierarchical unit selection combining Kullback-Leibler divergence with likelihood criterion. In *Proc. ICASSP*, 2007.

[5] HTS version 2.0, 2006. http:// hts.ics.nitech.ac.jp/.

[6] Y. Sagisaka, K. Takeda, M. Abe, S. Katagiri, T. Umeda, and H. Kuwabara. A large-scale Japanese speech database. In *Proc. ICSLP*, pages 1089–1092, 1990.

[7] K. Tokuda. Reference manual for Speech Signal Processing Toolkit ver. 3.0, 2002. http://kt-labics.nitech.ac.jp/˜tokuda/SPTK/.

[8] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. In *Proc. ICASSP*, May 2000.

[9] Y. Stylianou and A. K. Syrdal. Perceptual and objective detection of discontinuities in concatenative speech synthesis. In *Proc. ICASSP*, May 2001.

[10] A. V. Aho, J. E. Hopcroft, and J. Ullman. *Data Structures and Algorithms*. Addison-Wesley Pub. Co., 1982.

[11] T. Hirai and S. Tenpaku. Refinement of F0 distance measure in 5 ms segment concatenative speech synthesis. In *Rec. Spring Meeting, Acoust. Soc. Jpn.*, March 2007.

Clustering Algorithm for F0 Curves Based on Hidden Markov Models

Damien Lolive, Nelly Barbot, Olivier Boeffard

IRISA / University of Rennes 1 - ENSSAT
6 rue de Kerampont, B.P. 80518, F-22305 Lannion Cedex
France

{damien.lolive,nelly.barbot,olivier.boeffard}@irisa.fr

Abstract

This article describes a new unsupervised methodology to learn F_0 classes using HMM on a syllable basis. A F_0 class is represented by a HMM with three emitting states. The unsupervised clustering algorithm relies on an iterative gaussian splitting and EM retraining process. First, a single class is learnt on a training corpus (8000 syllables) and it is then divided by perturbing gaussian means of successive levels. At each step, the mean RMS error is evaluated on a validation corpus (3000 syllables). The algorithm stops automatically when the error becomes stable or increases. The syllabic structure of a sentence is the reference level we have taken for F_0 modelling even if the methodology can be applied to other structures. Clustering quality is evaluated in terms of cross-validation using a mean of RMS errors between F_0 contours on a test corpus and the estimated HMM trajectories. The results show a pretty good quality of the classes (mean RMS error around 4Hz).

Index Terms: prosody, fundamental frequency, unsupervised classification, Hidden Markov Model

1. Introduction

Technologies linked to speech processing widely use intonational speech models. We can particularly consider Text-to-Speech Synthesis (TTS) or a more emerging field as Voice Transformation. A TTS system needs prosodic models in order to create intelligible speech from text and elocution style. Most of works on this subject rely on a strong expertise in phonology and acoustic phonetics. A great challenge for a TTS sytem would be to offer a wide variety of prosodic models so as to diversify voice catalogs.

Nowadays, the majority of voice transformation systems use global prosodic adjustment (elocution rate and melody) [1]. An important issue would be to transform prosodic models between source and target speakers, notably of melodic contours. In order to easily adapt these models from various speakers and to limit manual expertise, an unsupervised methodology is necessary.

Although intonation is a combination of numerous linguistic factors, this article focuses on the acoustic parameter recognized to be the most prominent suprasegmental factor, the fundamental frequency or F_0. F_0 contours, extracted from the speech signal, represent the vibration of the vocal folds over time. A wide range of publications have reported on efforts in modelling F_0 evolution. We can particularly cite MoMel [2], Tilt [3], B-spline models [4], as well as Sakai and Glass's work [5] which use regular spline functions. Such stylizations offer a direct or parametric description of the F_0. A consequent literature deals with the fundamental frequency prediction problem from linguistic information [6]. This kind of modelling is supervised insofar as a segmentation in prosodic units is imposed and associated to F_0 curves.

As for the melodic contour classification issue, few works deal with an *unsupervised* F_0 clustering. The problem is to derive a set of basic melodic patterns from a set of sentences from which F_0 has been previously computed. The idea is that concatenation of elementary F_0 contours can characterize a complete melodic sentence [7]. We assume here that an atomic element of the melodic space is linked to the syllable. Thus, the objective is to learn a coherent set of melodic contour classes at the syllabic level. The major difficulty is to take into account the syllable duration. Two melodic contours with different temporal supports can represent the same elementary melodic pattern. Consequently, we choose to use Hidden Markov Models (HMM) which intrinsically integrate the elasticity of the representation support of an elementary form.

In this article, an unsupervised classification methodology for melodic contours is described. This methodology is based on the use of HMM in an unsupervised mode. The increase of the number of classes is realized thanks to a variant of Gaussian splitting on a HMM set.

The HMM structure and the procedure carried out to split a class are introduced in section 2. In section 3, the unsupervised learning algorithm applied to determine a set of melodic contour classes is described. The experimental methodology is then presented is section 4, as well as the evaluation method of class quality. The results are discussed in section 5.

2. Unsupervised HMM modelling

2.1. The model

In this article, we are interested in finding out a partition of a set of syllable melodic contours thanks to HMM. In our approach, a HMM characterizes a class and models F_0 contours which are monodimensional signals. Figure 1 shows the topology of the HMM used. Their construction is based on a syllable structure. Indeed, linguistics teaches us that a syllable can be divided into three parts: onset, nucleus and coda. This structure leads us to consider a model with three emitting states. Moreover, as onset and coda are optional, the state transition graph includes jumps which allow to avoid the first and last emitting states.

A HMM M_j is composed of five states and does not have any backward state transitions. States q_{0j} and q_{4j} are respectively the start and end nodes of the HMM. These two states are non-emitting and have a null sojourn time. As for the states q_{ij}, for i from 1 to 3, their output values are distributed according to a Gaussian law with mean μ_{ij} and variance σ^2_{ij}.

For a contour class M_j, the associated HMM parameters are trained using a standard Baum-Welch algorithm. Melodic

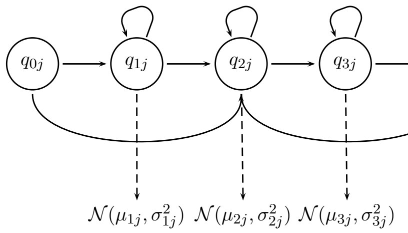

q_{0j}
q_{1j}
q_{2j}
q_{3j}
q_{4j}
$\mathcal{N}(\mu_{1j},\sigma^2_{1j})$
$\mathcal{N}(\mu_{2j},\sigma^2_{2j})$
$\mathcal{N}(\mu_{3j},\sigma^2_{3j})$

Figure 1: *Structure of HMM M_j: it is composed of three emitting states q_{1j}, q_{2j}, q_{3j}. Their output values are supposed to be Gaussian. States q_{0j} and q_{4j} are start and end nodes.*

contours are labeled thanks to the Viterbi algorithm that proposes an unsupervised decoding. The grammar used for decoding permits to respect the syllable indivisible nature. No loop is enabled and only one HMM can be chosen among the whole HMM set $\mathcal{M}$.

This work takes place in an unsupervised framework, the number of classes is *a priori* unknown. We then propose to increase the number of classes by dividing the existing classes. The strategy presented in paragraph 2.2 answers this problem and also provides an initialization of the HMM training after the division process.

2.2. Gaussian splitting

In the previous section, we have introduced the model used to describe a class. We now propose a solution to divide a class, that is to say a HMM, into two distinct classes based on Gaussian splitting.

In [8, 9], we find two different applications of Gaussian splitting. It is a practical method that enables to increase the number of classes and to initialize the new class parameters for the retraining phase. This method consists in slightly perturbing the mean of the Gaussian law associated to each state of a HMM. In this article, we use this method to create two HMM from a single one.

For a class in the training corpus (a set of syllables), we denote M_j the associated HMM which is estimated according to the maximum likelihood criterion. To obtain two classes, we split the HMM M_j by perturbing the means μ_{ij} of the Gaussians associated to the states q_{ij}. The means are modified along the standard deviation direction σ_{ij} of the corresponding Gaussian:

$$\mu^+_{ij} = \mu_{ij} + \epsilon * \sigma_{ij} \tag{1}$$

$$\mu^-_{ij} = \mu_{ij} - \epsilon * \sigma_{ij} \tag{2}$$

where ϵ is a constant fixed to 0.001 in our experiments. The specialization of the two new HMM is done using the Baum-Welch algorithm.

3. Unsupervised learning algorithm

The learning process of the set of melodic contour classes is realized in an unsupervised manner. We do not have classes already defined from which we can train the HMM. Under the proposed model assumption, the main goal is to cluster forms that look alike.

The strategy described in figure 2 builds a set of classes from three elements: the set of contours, the method to split classes and a measure allowing to decide which classes must be divided.

```
Input: $NbToSplit$ the number of HMM to split at each
       step
Output: $\mathcal{M} = \{M_1, \ldots, M_p\}$
$\mathcal{M} = \{M_1\}$;
$e_{prev} = +Inf$;
$\epsilon = 1e^{-4}$;
converged = false;
repeat
    foreach HMM $M_i \in \mathcal{M}$ do
        - learn $M_i$ using the Baum-Welch algorithm on
          the training corpus
    end
    - re-label all syllables of the validation corpus with
      the new HMM $\mathcal{M}$ (Viterbi);
    - re-compute the mean RMS error $e_{cur}$ between
      each syllable and its HMM class model;
    if $e_{prev} - e_{cur} < \epsilon$ then
        converged = true;
    else
        - divide $\mathcal{M}$ into two HMM sets $\mathcal{M}_1$ and $\mathcal{M}_2$
          with card($\mathcal{M}_1$) = $NbToSplit$;
        - split each HMM of $\mathcal{M}_1$ into $\mathcal{M}_1^{new}$;
        - merge $\mathcal{M}_1^{new}$ and $\mathcal{M}_2$ into a new HMM set
          $\mathcal{M}^{new}$;
        - re-label all syllables according to the new
          HMM set $\mathcal{M}^{new}$;
        $\mathcal{M} = \mathcal{M}^{new}$;
        $e_{prev} = e_{cur}$;
    end
until converged = true ;
```

Figure 2: *Unsupervised algorithm used to learn the melodic contour classes*

The algorithm first considers one class to which a HMM is associated. At each step of the algorithm, we split a subset of the existing classes to create new classes. Considering the algorithm has done a certain number of iterations, we then have a HMM set $\mathcal{M}$. After the learning step of the models in $\mathcal{M}$, the global mean RMS error (Root Mean Square error) is computed on a validation corpus. For a F_0 contour of length d, the RMS error calculation is done in the following way:

- We compute the optimal state sequence $(T_t)_t \in \{q_{1j}, q_{2j}, q_{3j}\}^d$ of the HMM M_j using the Viterbi algorithm.
- To each state T_t, we associate the mean value $\mu_{T_t j}$ of the Gaussian in the state T_t of the HMM M_j.
- The RMS error is then computed between the F_0 observations and that sequence of mean values:

$$RMS^2 = \frac{1}{d}\sum_{t=1}^{d}(F_0(x_t) - \mu_{T_t j})^2 \tag{3}$$

The algorithm convergence is then evaluated in function of the mean RMS error on the validation corpus: we consider that the convergence is achieved if the mean RMS increases or is stable. If the algorithm has not converged at this step, we construct the subset $\mathcal{M}_1$ constituted by the $NbToSplit$ HMM that have the highest value for a criterion, four are tested in section 4.3. These HMM are then split each one into two HMM, in order to obtain more accurate classes in terms of mean RMS error. The number of HMM to split $NbToSplit$ is a parameter of the algorithm.

Once we have the new set of classes $\mathcal{M}^{new}$ coming from the splitting of $\mathcal{M}_1$, the Viterbi algorithm is applied to modify the F_0 contour labels in the training corpus and to make them correspond to the new classes. Thenceforth, we can learn the new HMM on the modified training corpus. The Gaussian splitting process is repeated until the algorithm reaches a convergence threshold. During the splitting step, if a HMM does not capture a sufficient number of contours, then the algorithm goes on without splitting it.

4. Experimental methodology

4.1. F0 corpus

Experiments are conducted on a set of syllables randomly extracted from a 7,000 sentence corpus. The acoustic signal was recorded in a professional recording studio; the speaker was asked to read the text. Then, the acoustic signal was annotated and segmented into phonetic units. The fundamental frequency, F_0, was analyzed in an automatic way according to an estimation process based primarily on the autocorrelation function of the speech signal. Next, an automatic algorithm was applied to the phonetic chain pronounced by the speaker so as to find the underlying syllables. The corpus of the selected syllables is divided into a training corpus ($8,000$ syllables) and a validation corpus ($3,000$ syllables).

4.2. Data pre-processing

The first step concerns the conversion of the F_0 values in cent. The cent, which is the hundredth of a semi-tone, is a unit that makes a parallel with the logarithmic scale of the ear. The conversion from Hertz to cent is given by equation 4, where $F_0^{ref} = 110$Hz.

$$F_0^{cent} = 1200 * \log_2 \left(\frac{F_0^{hertz}}{F_0^{ref}} \right) \tag{4}$$

The second step is similar to the process achieved in [10]. It realizes a linear interpolation of unvoiced parts of the F_0 curves at the sentence level. This interpolation comes from the hypothesis according to which a continuous melodic gesture exists, the fundamental frequency value is then masked during unvoiced parts. Moreover, a linear regression is done on the interpolated F_0 curves in order to suppress microprosodic variations.

4.3. Experiments

The main goal of this study is to establish unsupervised classes from a speech corpus. Thus, the use of common evaluation methodology in order to evaluate the quality of the classes is impractical.

In our case, we propose to evaluate the overall quality of the clustering in relation to the similarity of the contours grouped according to their shape and independently of their duration. To do that, we use a RMS error calculation between a syllable and the optimal trajectory of the associated HMM. We can obtain a RMS error for an entire class, that we want as small as possible and notably smaller than the common JND threshold for the F_0 (about 4Hz).

Moreover, to be able to compute the RMS error and compare the results to the JND threshold (for F_0), we convert the melodic contours and the mean trajectory of the associated HMM into hertz.

In the next section, three experiments are presented. The first one shows an example of a curve and its class HMM trajectory. The aim of this experiment is to show how a curve and its duration are represented by a HMM. The second experiment shows the evolution of the RMS error for the CMSE (Cumulative MSE) criterion in function of the number of classes for three $NbToSplit$ values. The third experiment compares the four following class selection criteria:

1. mRMSE: for each class we compute the mean RMS error, classes are then sorted according to this value,
2. RMSEv: we compute the RMS error variance for each class, so low variance classes are kept while high variance classes are split,
3. CMSE: the global error for a class is calculated by summing the squared RMS values (Cumulative MSE),
4. CMSE_n: the Cumulative MSE divided by the number of curves in the class is computed for each class. In this case, the global error is equally distributed over the curves.

They are compared in terms of RMS error and number of HMM at each iteration of the algorithm.

5. Results and discussion

5.1. F0 contour example

Figure 3 shows an example of a melodic contour and the trajectory of the HMM associated to its class. We can observe the sequence of the HMM states over time. For this example, the HMM stays in state q_1 during the first four observations. The Gaussian mean that corresponds to this state is approximately 107Hz. In this example, the RMS error between the F_0 contour and the HMM trajectory is around 1Hz. The analysis of this figure shows that the states of the HMM capture the general shape of the contour. The time evolution and thus the length of the contour is catched by the loops at the level of each HMM state. Consequently, each HMM reflects a particular form which is independent of duration and enables the modelling of melodic contours of different lengths but of similar shape.

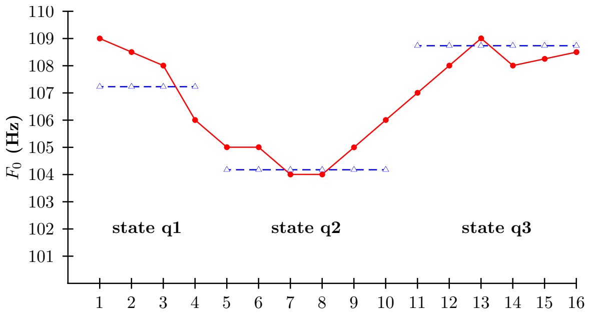


Figure 3: *Example of an HMM class and a F_0 contour taken within this class. The melodic contour (red line) is superposed to the mean values of the Gaussians associated to the states of the HMM (dashed blue line). The sequence of the three HMM states for this syllable is written below the curves.*

A HMM state models a constant melodic segment and the first derivative could be useful to better follow the evolution of the melodic contour. For practical purposes, this could be realized by the joint use of the F_0 values and the first derivative

Table 1: *Mean RMS error (Hz) with* 95% *confidence intervals for the three split variants on the validation corpus*

N. of HMM	Split-1	Split-2	Split-n
1	11.44 ±0.18	11.44 ±0.18	11.44 ±0.18
2	9.87 ±0.16	9.87 ±0.16	9.87 ±0.16
4	9.23 ±0.15	9.30 ±0.15	9.30 ±0.15
8	7.25 ±0.15	7.87 ±0.12	8.26 ±0.14
16	5.48 ±0.12	5.79 ±0.11	6.74 ±0.13
32	4.86 ±0.11	4.82 ±0.10	5.76 ±0.12
64	4.56 ±0.10	4.54 ±0.11	5.15 ±0.11
128	4.27 ±0.10	4.25 ±0.11	4.68 ±0.11

Table 2: *Mean RMS error (Cent) with* 95% *confidence intervals for the three split variants on the validation corpus*

N. of HMM	Split-1	Split-2	Split-n
1	165.50 ±2.30	165.50 ±2.30	165.50 ±2.30
2	140.89 ±2.01	140.89 ±2.01	140.89 ±2.01
4	130.96 ±1.92	131.98 ±1.90	131.98 ±1.90
8	104.80 ±2.05	113.86 ±1.71	118.85 ±1.92
16	79.81 ±1.68	84.91 ±1.58	98.26 ±1.73
32	71.37 ±1.56	70.53 ±1.40	84.28 ±1.69
64	66.97 ±1.50	66.16 ±1.53	75.63 ±1.59
128	62.62 ±1.48	62.03 ±1.49	68.53 ±1.50

values. However, taking into account this problem is relatively complex and leads us to difficulties concerning the estimation of the class quality. Instead of taking into account explicitly the first derivative, we can also increase the number of the states. In this case, the estimation process turns out to be an overestimated solution considering the high number of parameters.

5.2. Results for the CMSE criterion

Mean RMS errors related to the number of classes are presented in tables 1 and 2. This experiment is carried out with three different $NbToSplit$ threshold values:

- *Split-1*: $NbToSplit = 1$, we divide only one HMM at each iteration.
- *Split-2*: $NbToSplit = 2$, two HMM are divided at each iteration.
- *Split-n*: all the HMM are split into two parts at each iteration.

In table 1, we can see that, on the validation corpus, the RMS error decreases while the number of HMM increases for the three split methods. However, the error does not evolve in the same manner for the three cases. Concerning *split-1* and *split-2*, the number of HMM split at each iteration is small. The consequence is a lower RMS error (around 4Hz) than the *split-n* case, on the contrary the number of iterations necessary to obtain 128 HMM is greater. A bigger value for $NbToSplit$ increases the convergence speed (*split-n* case), but the RMS error is higher (greater than 5Hz). Generally speaking, we can conclude that relatively few classes are necessary to obtain a RMS error near the F_0 JND threshold around 4 Hz.

In table 2, the mean RMS errors in function of the number of classes are expressed in cent. The evolution of the error is the same as in table 1. We can notice that, for at least 16 classes, the error is smaller than one semi-tone (100 cents). Moreover, for the *split-1* and *split-2* cases, with 128 classes, the error is near a quarter of tone.

The errors presented in these two tables enable us to conclude that the distance between a contour and the associated trajectory of the HMM is small. This implies that the shapes of the melodic contours inside a class are similar. So a class reflects a particular elementary form and the set of classes is a quite good partition of the melodic contour corpus.

5.3. Behavior of the class selection criteria

To select the classes to split, we have tested four criteria (see section 4.3). Figure 4 shows the evolution of the RMS error for each criterion considering the *split-1* case. We can observe that the error decreases quickly during the first 20 iterations. Indeed, during the first iterations, the number of classes is small and data are easily separable. Consequently, adding a new HMM, ie. increasing the number of classes by one, is very efficient when the number of classes is small. Moreover, the difference between the four criteria is not distinguishable, the 95% confidence intervals are not disjoint. Concerning the RMS error, the best criterion in this experiment is the Cumulative MSE (CMSE) which leads to a mean error near 4Hz.

As the number of classes is unknown a priori, the number of iterations for each criterion is variable. In the mRMSE case, we can notice that it is very small (smaller than 60) while in the other cases the number of iterations is over 150.

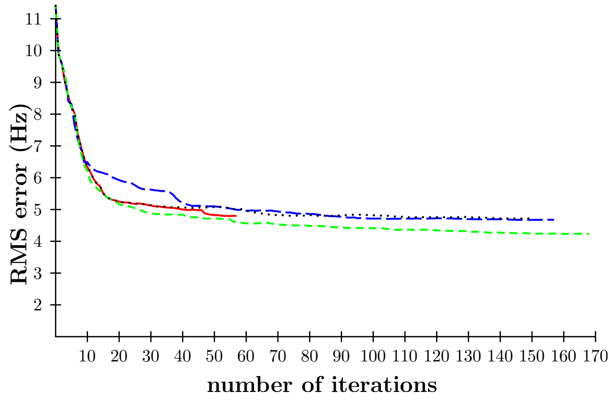


Figure 4: *Evolution of the RMS error for the four class selection criteria in the split-1 case: mRMSE (red line), RMSEv (long dashed blue curve), CMSE (dashed green curve) and CMSE_n (dotted black curve).*

Figure 5 represents the evolution of the number of HMM for the four criteria. As in figure 4, the number of iterations is varying from one criterion to another. Concerning the number of HMM, we can observe that its evolution is quite different between the four criteria. Indeed, in the CMSE case, the evolution of the number of HMM is nearly linear. On the contrary, the RMSEv and the CMSE_n cases have stages where the number of HMM is constant. During these stages, the algorithm can not split any of the HMM, but some iterations with a constant number of classes enable the algorithm to recompute new models and improve the set of classes. This processing continues until the RMS error increases significantly or stabilizes.

The comparison between figures 4 and 5 shows that when the number of classes is high, classes are specialized and their

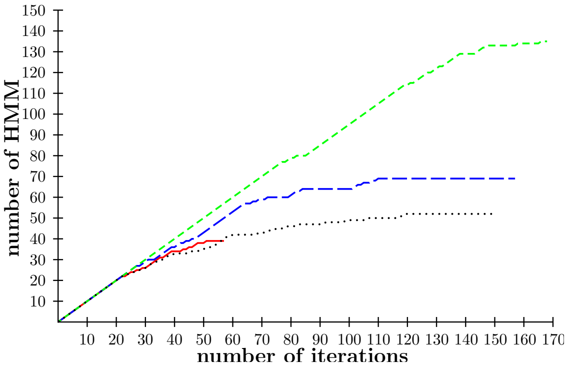


Figure 5: *Evolution of the number of HMM for the four selection criteria of the split-1 case: mRMSE (red line), RMSEv (long dashed blue curve), CMSE (dashed green curve) and CMSE_n (dotted black curve).*

mean RMS error is low. In this case, the addition of a new class enables classes to specialize a little bit more. The consequence is that higher the number of classes is, lower and more stable the mean RMS error is.

With the help of these figures, we can think about the more efficient criterion. First of all, we can select a criterion according to the RMS error. In this case, the most suitable of the four proposed criteria is CMSE which leads to an RMS error near 4Hz. We can also be more interested in the most efficient criterion, ie. we can then choose it as a compromise between the mean RMS error and the number of classes. This point of view makes the CMSE criterion the less efficient. Indeed, the number of classes for this criterion is twice the number of classes of the others while the error is not much less.

Despite of this fact, the CMSE criterion has good properties that the others do not have. Indeed, the mRMSE and the CMSE_n cases are mean values with respect to the number of F_0 contours in the classes. So the error is equally distributed over the contours which compose each class. Consequently, in a class, badly modelled curves are masked by well modelled curves. Concerning the RMSEv criterion, a class can have a low variance and even a high RMS error. Then, this criterion is not consistent, and the RMS error as well as the number of HMM quickly stabilize. Finally, as the global error for the CMSE criterion is not divided by the number of curves falling into each class, if a curve has a high RMS error, its class will be split. Moreover this criterion is coherent with the goal of a optimal RMS error in opposition to the RMSEv criterion.

6. Conclusion

In this article, a new unsupervised learning methodology based on HMM for melodic contour classes is described. The results show a pretty good precision of the classes. The mean RMS error is near 4Hz which is the common JND threshold for the F_0. Besides, HMM modelling enables to cluster contours of similar shape independently of their duration. Four class selection criteria were presented, we show that a CMSE criterion gives the most accurate results.

The experiments presented in this paper are based on melodic contours at a syllabic level. This methodology can be easily adapted to other temporal units like syllable sequences or intonational units.

Having a set of melodic contour classes for two speakers, we will try to estimate a conversion function enabling the transformation from one's speaker melodic contour classes (source speaker) into the classes of a target speaker. Moreover, the classification of melodic contours gives output labels corresponding to the F_0 patterns. These labels could be used in a TTS system to enhance it and diversify the possible synthesized voices at a prosodic level.

7. References

[1] Gillet, B. and King, S., “Transforming f0 contours”, Proceedings of the eurospeech conference, 2003.

[2] Hirst, D., Di Cristo, A. and Espesser, R., “Levels of representation and levels of analysis for the description of intonation systems”, Prosody : theory and experiment, kluwer academic pusblisher, M. Horne, Ed., vol. 14, 2000, pp. 51–87.

[3] Taylor, P., “Analysis and synthesis of intonation using the tilt model”, J. Acoust. Soc. America, 107:1697-1714, 2000.

[4] Lolive, D., Barbot, N. and Boeffard, O., “Comparing b-spline and spline models for f0 modelling”, Lecture notes in artificial intelligence - proceedings of the 9th international conference on text, speech and dialogue - brno, czech republic, P. Sojka, I. Kopecek and K. Pala, Ed., Berlin, Heidelberg: Springer Verlag, 2006, pp. 423–430.

[5] Sakai, S. and Glass, J., “Fundamental frequency modeling for corpus-based speech synthesis based on statistical learning techniques”, Proceedings of the ASRU conference, 2003, pp. 712–717.

[6] Traber, C., Talking machines : theories, models and designs. Elsevier B.V., 1992, ch. Fo generation with a database of natural F0 patterns and with a neural network, pp. 287–304.

[7] Mertens, P., “Automatic recognition of intonation in french and dutch”, Proceedings of the eurospeech conference, 1989, pp. 46–50.

[8] Sankar, A., “Experiments with a Gaussian Merging-Splitting Algorithm for HMM training for Speech Recognition”, DARPA Speech Recognition Workshop, Feb. 1998.

[9] Rabiner, L., Lee, C., Juang, B. and Wilpon, J., “HMM clustering for connected word recognition”, Acoustics, Speech, and Signal Processing, vol. 1, May 1989, pp. 405–408.

[10] Yamashita, Y., Ishida, T. and Shimadera, K., “A Stochastic F0 Contour Model Based on Clustering and a Probabilistic Measure”, IEICE Transactions on Information and Systems, vol. E86-D, no. 3, Mar. 2003, pp. 543–549.

Building a Better Indian English Voice using “More Data”

Rohit Kumar, Rashmi Gangadharaiah, Sharath Rao,
Kishore Prahallad, Carolyn P. Rosé, Alan W. Black

Language Technologies Institute
Carnegie Mellon University, Pittsburgh, PA, USA
{ rohitk, rgangadh, skrao, skishore, cprose, awb } @ cs.cmu.edu

Abstract

We report our experiments towards improving an existing publicly available Indian English voice using additional data. The additional data was used to create new duration and pronunciation models as well as to convert the existing voice to create a more Indian sounding voice. Two experiments along the above lines are reported. In the first experiment, we found that changing the pronunciation models has the potential to improve an existing Indian English voice. We conducted a second experiment to validate this finding. The second experiment shows the potential value in carefully investigating the separate effects of the different components of a pronunciation model in order to understand their unique contributions to improving an Indian English voice.

1. Introduction

English is the official language of India. Over 200 million people use Indian English. In this paper, we refer to the English used in news telecasts as Indian English. The English used in India, although originally acquired by native Indian speakers during the course of the British rule, is known to have undergone transformations along various dimensions of the language including its phonology, morphology, syntax and word usage [1]. While borrowing models from American or British English may be the right way to bootstrap Indian Language systems, it is essential that changes in the above mentioned aspects of Indian English are modeled appropriately in these systems.

Our motivation for this work is two fold. First, we want to develop a better Indian English voice. Second, we want to study whether additional data can be used either to improve a given Indian English voice or to build newer voices with very little data. We hypothesize that additional data can be used to improve multiple models used in any text to speech system (TTS). In particular we focus on three key components of a TTS, i.e., the duration model, the pronunciation model, and the voice data used to build the synthesis model.

The remainder of the paper is organized as follows. Section 2 discusses the design and results of the first experiment. Section 3 describes the second experiment along with our findings. Discussion of the results from both the experiments is found in Section 4 which is followed by conclusions and next steps.

2. Experiment 1: The new models

In the first experiment, we used additional data to create new duration, pronunciation and synthesis models. We experimentally evaluate their separate effects on two different response variables.

2.1. Data

We start with two baseline voices (KSP and BDL) distributed as a part of the CMU Arctic [2] set of voices. Both of these voices include recordings of 1132 optimally selected sentences. KSP is the voice of a native Indian who is a fluent speaker of Indian English. BDL is the voice of a standard American English speaker. Both KSP and BDL are male speakers.

The additional data we used is comprised of an Indian English pronunciation lexicon and speech recorded by five male Indian English speakers. Each of the five speakers recorded 100 sentences of the CMU Arctic set. These utterances were originally recorded for the ConQuest project to build acoustics models for an Indian English speech recognition system. Hence the recording was done in an office space unlike the CMU Arctic KSP and BDL voices which were recorded in a recording booth. Given the number of utterances per speaker and the quality of the recordings, the additional data by itself was not suitable for building high quality synthesis voices. Hence we use this data for building new duration models as well as for conversion as described later in this section.

2.1.1. Indian English Pronunciation Lexicon

The Indian English pronunciation lexicon was built specifically for this project. It is comprised of 3489 words derived from the 1132 CMU Arctic sentences and the 200 sentences from the SCRIBE Project [3]. An American English phoneme set was used to represent the pronunciation of these words in Indian English. Despite the differences between the American and Indian English, an American English phoneme set was used to represent the pronunciations in the Indian English lexicon because it allows us to bootstrap the Indian English dictionary from existing letter to sound rules as described ahead.

We used the CMU Dictionary [4] and a set of letter to sound rules built from the dictionary to generate American English pronunciations for the 3489 words. These pronunciations were then corrected by the authors to match the Indian English pronunciations. During corrections, if a desired phoneme was unavailable in the phoneme set, the nearest available phoneme (in terms of minimal mismatch of articulatory descriptors) was chosen.

After the manual corrections, the new Indian English phoneme sequences were syllabified and stress marked using a set of rules derived from characteristics of Indian Languages as discussed below.

The basic units of the writing system in Indian languages are referred to as “Aksharas”. The properties of Aksharas are as follows: (1) An Akshara is an orthographic representation of a speech sound in an Indian language; (2) Aksharas are syllabic in nature; (3) The typical forms of Akshara are V, CV, CCV and CCCV, thus have a generalized form of C*V; (4) An Akshara always ends with a vowel (which includes nasalized vowels); [5]. In view of these points, given a sequence of phones, one can consistently mark syllable boundaries at vowels. This heuristic is typically followed in building TTS systems for Indian languages [6]. At the same time, a simple set of rules are followed to assign stress to the syllables. A primary stress level is associated with the first syllable and to the other syllables which have non-schwa vowels. A secondary stress is associated with the rest of the syllables which have schwas. Assuming that Indian English speakers tend to borrow syllabification and stress assignment characteristics from their native languages, we wanted to investigate how the use of these rules would affect the quality of an Indian English TTS.

On analyzing the new Indian English Pronunciation lexicon we observed that only 918 (26.3%) words needed any correction at all. At the phoneme level only a 7.2% change was observed. The majority of these changes were phoneme substitutions. The most common substitution included vowel substitutions (like /aa/ → /ao/ e.g. h**o**stilities). Also, several common consonant substitutions like /z/ → /s/ and /w/ → /v/ were observed.

2.2. The New Models

We created 15 different voices using different combinations of converted voices, duration models and pronunciation models. We used the FestVox framework [7] to build all of these different models and voices.

2.2.1. The converted voices

We used the speech from two of the 5 speakers in the additional data to convert the KSP and BDL utterances. A converted set of utterance is represented as a 2-tuple <SOURCE, TARGET>. The SOURCE refers to the original speaker whose utterances are being converted. SOURCE can be KSP and BDL in our case. TARGET refers to the speaker to which SOURCE is being converted. One of the two target speakers we used from the additional data is a North Indian (NIE) speaker, and the other is a South Indian (SIE) speaker. Also it may be noted that KSP is a South Indian speaker too.

We use a GMM based Spectral conversion method [8] to create the converted voices. The 5 converted voices are <KSP, NIE>, <KSP, SIE>, <BDL, NIE>, <BDL, SIE> and <KSP, KSP> respectively. The <KSP, KSP> converted voice is used to compare the new voices with the existing Indian English voice and can be assumed to have the lowest distortion due to conversion.

2.2.2. The duration models

The duration models predict the duration of a phoneme during synthesis. The models are trained on phoneme segments obtained by automatically segmenting the given utterances. We use a publicly available Ergodic HMM based segmenter distributed with FestVox.

The baseline duration model was built using the 1132 utterances of the KSP voice. The experimental duration model in this case was built using the 1132 utterances of the KSP voice and the 500 utterances from the additional data. We refer to the experimental duration model as KSP++ which we contrast with the baseline duration model, namely KSP. Both the duration models are built using correlation and regression trees (CART) and are based on phonetic and syllabic features of the segment as well as its context.

2.2.3. The pronunciation models

A pronunciation model converts a given word to its pronunciation. The pronunciation of a word is comprised of the phoneme sequence corresponding to the sounds of the word and the syllabification of the phoneme sequence. Each syllable also carries information about its stress. A typical pronunciation model is comprised of a dictionary and a set of letter to sound (LTS) rules. The LTS rules may either be hand crafted or learnt from the dictionary. Given a word, a pronunciation model typically does a lookup in the dictionary. In case the dictionary does not contain the pronunciation of

Table 1. Results of the first Experiment (sorted by Mean Intelligibility)

Converted Voice	Duration Model	Pronunciation Model	Intelligibility		Indian-ness	
			Mean	Std. Dev	Mean	Std. Dev
KSP, KSP	KSP	IE	4.9	1.79	5.92	1.41
KSP, KSP	KSP++	IE	4.87	1.79	5.37	1.86
KSP, KSP	KSP++	CMU	4.48	1.83	5.3	1.89
KSP, SIE	KSP++	IE	4	1.78	5.4	1.69
KSP, SIE	KSP	IE	3.85	2.07	5.02	1.8
BDL, SIE	KSP++	CMU	3.78	1.97	2.77	2.15
KSP, NIE	KSP	IE	3.7	2.22	4.73	2.25
KSP, NIE	KSP++	IE	3.48	1.93	4.38	2.12
BDL, NIE	KSP++	CMU	3.48	2.07	2.53	1.79
KSP, SIE	KSP++	CMU	3.4	2.16	4.47	2.14
BDL, SIE	KSP++	IE	3.18	2.05	2.55	2.06
BDL, NIE	KSP	IE	3.17	2.06	2.83	1.63
BDL, NIE	KSP++	IE	3.13	1.88	3.17	1.7
KSP, NIE	KSP++	CMU	3.1	2	4.75	1.76
BDL,SIE	KSP	IE	2.87	2.04	2.72	1.78

the given word, the LTS rules are used to generate the pronunciation of the word.

We use two different pronunciation models in the first experiment. The baseline pronunciation model (CMU) is built from the CMU Dictionary consisting of over 105,000 words. The experimental pronunciation model which we refer to as IE, is built from the Indian English pronunciation lexicon of 3489 words described earlier. The LTS rules for both the models have been trained using CART [9].

2.3. The pilot experiment

To study the effect of (1) the different source and target voices, (2) the duration models and (3) the pronunciation models, we created 15 different festival [10] compatible voices. All voices are built to use a Unit Selection Synthesizer [11]. Table 1 lists the 15 different voices in terms of the models and converted voice they use.

In the first experiment, these 15 voices were subjectively evaluated for two different perceived measures: Intelligibility and Indian-ness. 15 subjects were asked to listen to 60 utterances and score each utterance for both the measures independently on a scale 0 to 7. For Intelligibility, they were instructed to score a zero if they did not understand even a single word of the utterance and to score a 7 if the utterance was perfectly understandable. For Indian-ness, they were instructed to score a 0 if the utterance did not sound like an Indian speaker at all and to score a 7 if the utterance sounded perfectly like an Indian speaker. Subjects were instructed to evaluate both the measures independent of each other.

15 subjects participated in this evaluation under controlled conditions. All subjects used the same equipment (laptop, speakers) and performed the listening task in the same office. All subjects are of Indian origin and are graduate students at Carnegie Mellon University. They have not been outside India for more than 4 years. The subjects were 21 to 27 years old.

The 60 utterances given to the subjects were composed of 4 utterances from each voice in random order in order to avoid ordering effects.

2.4. Preliminary evidence and directions

Table 1 enumerates the average scores for each of the voice on both the measures along with the corresponding standard deviations. The voice built from the KSP → KSP conversion performed best among all the other voices. The KSP Source voice was scored significantly higher than the BDL voice on both the measures. Further, the KSP voice as a target was significantly better than NIE. SIE was not significantly different from either KSP or NIE as a target voice. The <KSP, KSP> converted voice performed better than all the other converted voices because the distortion caused by conversion was minimal for that pair. However SIE not being significantly different from KSP shows the potential for creating new voices using a baseline voice and very little speech data from a target voice in the case where the source and target speakers have similar characteristics. Both SIE and KSP are South Indian English speakers of comparable age and educational background.

There was no effect of the duration model on either of the outcome measures. We found that both the duration models selected exactly the same sequence of units per utterance despite generating different targets. We understand that this is because of the low cost associated with duration mismatch as well as the restricted diversity of units in the inventory. The units matching the targets generated by both the duration models turn out to be the same in all cases.

Comparing across all the 15 experimental voices, we found no significant difference between the two pronunciation models. However, if we restrict our attention to the data from the <KSP, KSP> converted voice, we then see a significant difference in the average Intelligibility between the pronunciation models (p=0.008) when we included a variable in the model indicating for each judgment which sentence was spoken to account for variance caused by differences in the words included across sentences. A similar effect was observed for the voices based on the <KSP, SIE> converted voice (p=0.044).

Based on the evidence that <KSP, KSP> was the best of the converted voices and that <KSP, SIE> was among the better ones of the converted voices, ranking second according to the average intelligibility scores, we hypothesize that the improvements due to the IE pronunciation model were observable only in the good voices which were least distorted due to voice conversion. Based on this reasoning, we decided to further investigate the effect of the experimental pronunciation model using high quality voices like the unconverted CMU Arctic KSP voice.

3. Experiment 2: The field study

In the follow up experiment, we decided to focus on studying the contribution of the pronunciation model towards building a better Indian English voice. Unlike the first experiment, we conducted the second study in India.

In this experiment, we wanted to compare the two pronunciation models from the first experiment, CMU and IE, with high quality voices which have been built without any degradation due to voice conversion. We start with CMU Arctic KSP data and use two different synthesis techniques supported by Festival [10] to build the high quality voices: A unit selection approach referred to as CLUNITS [11] and a statistical parametric synthesis technique called CLUSTERGEN [12].

3.1. Three pronunciation models

To further study the contribution of the various components of the Indian English pronunciation model we introduce an intermediate pronunciation model derived from the CMU Dictionary. The intermediate pronunciation model (referred to as CMU+IESyl) was built by applying the Indian English syllabification and stress assignment rules to the baseline CMU Dictionary.

The intention of using this intermediate model was to study the individual contributions of two macro components of the Indian English pronunciation model i.e. the pronunciation (letter to sound rules) and the rules for syllabification and stress assignment. While CMU and CMU+IESyl pronunciation models can be compared to study the effect of the syllabification and stress assignment rules, the contrast between CMU+IESyl and IE pronunciation models can be used to study the contribution of the modified pronunciations for Indian English.

Table 2. Results of the field Experiment

Synthesis Technique	Pronunciation Model	Intelligibility		Naturalness	
		Mean	Std. Dev	Mean	Std. Dev
CLUNITS	CMU	3.83	1.18	3.37	1.1755
CLUNITS	CMU+IESYL	3.76	1.2	3.33	1.2368
CLUNITS	IE	3.88	1.16	3.48	1.1853
CLUSTERGEN	CMU	2.80	1.36	2.21	1.3597
CLUSTERGEN	CMU+IESYL	2.82	1.38	2.24	1.3972
CLUSTERGEN	IE	2.92	1.38	2.23	1.3737

3.2. Experimental Design

We built 6 different voices using all combinations of the 3 pronunciation models (CMU, CMU+IESyl, IE) and the 2 synthesis techniques (CLUNITS, CLUSTERGEN). All voices were built on the CMU Arctic KSP data.

Duration models were trained on the same data for all the voices. However, it must be noted that as the phoneme sequence for several words would be different for the different pronunciation models, the duration models will not be exactly the same for all the voices. We think that this is acceptable as building the duration model does not need any new knowledge engineering into the voice since they are built fully automatically given the KSP utterances and automatically generated segment labels. Table 2 enumerates the 6 voices.

23 participants evaluated all the 6 voices on two different measures: Intelligibility and Naturalness. Both these measures are similar to those used in the first experiment. We choose the term Naturalness instead of Indian-ness in this study as the participants in this study are resident in India. In this study a scale of 0 to 5 was used for both the outcome measures. The instructions for scoring each of the measures were similar to those in the first experiment.

The subjects used a web based interface to evaluate upto 6 sets of 30 utterances. Most of the subjects completed all the 6 sets in their evaluation. All subjects were 20 to 27 years old students at IIIT Hyderabad, India. Each set contained the same 30 sentences, 5 synthesized by each of the 6 voices. However in every set the 5 sentences synthesized by each voice were different. Further each set was randomized to avoid any ordering effects.

For our analysis, we consider a session to be the duration a single participant spends on evaluating one of the 6 sets. 128 sessions were completed among the 23 participants and in total 3840 utterances were evaluated.

3.3. Results

The results from the second experiment are shown in Table 2. We find a significant effect of the pronunciation model on the Intelligibility measure considering the session as a random factor in the analysis. $F(2, 3710) = 3.24$, $p < 0.04$. The IE pronunciation model proves to be better than the CMU+IESYL pronunciation model, although the effect size is very small ($p < 0.05$, effect size = 0.079).

In order to contrast between the different components of the 3 pronunciation models, we compared the CMU+IESYL and the IE pronunciation models. We found the Indian English pronunciation lexicon had a small but significant effect on Intelligibility as compared to the CMU dictionary when both of them use the same syllabification rules and stress marks.

On comparing the CMU and CMU+IESyl pronunciation models, we found no effect of the syllabification and stress marking rules in improving the intelligibility of Indian English. This observation leads us to conclude that the new pronunciation lexicon contributes to improving the Indian English voice. These studies also highlight that modifications in pronunciation lexicon provide better improvement in intelligibility than use of modified stress and syllable patterns on baseline CMU dictionary.

We also observe that the CLUNITS synthesis performs better than the CLUSTERGEN technique on both the measures ($p < 0.001$, effect size for intelligibility=0.71 and effect size for Indian-ness=0.85) for all the three pronunciation models.

4. Discussion

There have been other efforts in building an Indian English TTS. An Indian-accent TTS [13] uses a pronunciation model which does a morphological analysis to decompose a word and then looks up the pronunciation of the constituents in a dictionary containing about 3000 lexical items. If the pronunciation of any constituent is not found in the dictionary, it uses a set of hand crafted letter to sound rules [14] to obtain the pronunciation. [15] describes a method to build non-native pronunciation lexicons using hand-crafted rules in a formalism capable of modeling the changes in pronunciation from a standard (UK/US) pronunciation to a non-native pronunciation. [16] also describes a formalism and a set of rules for letter to sound transformation. However, unlike [14] and [15], [16] also discusses rules for syllabification as a part of pronunciation modeling.

Unlikely the above mentioned, we use automatic methods to derive the letter to sound rules. None of the mentioned work discusses stress assignment which we consider as an integral part of pronunciation modeling.

In this paper we have evaluated the contribution of pronunciation modeling in an Indian English TTS. This work reports our current finding and lays out directions for further investigation into the roles of pronunciation model and its components in building an Indian English TTS.

We believe the mismatch between pronunciations in the CMU Dictionary and the Indian English syllabification and stress assignment rules caused the CMU+IESyl pronunciation model to under perform. We are interested in improving the syllabification and stress assignment rules used for Indian Languages to be suitable for use with Indian English pronunciation modeling. Also, we would like to study the use of a larger manually modified pronunciation lexicon to improve the IE pronunciation model.

5. Conclusions

We conducted two experiments to evaluate new models for improving an existing Indian English voice. We found that voice conversion can be a useful technique for creating new voices with little data from an existing voice, particularly when the new voice and the existing voice share qualitative characteristics.

We also find that an Indian English pronunciation model can be the key to building a better Indian English voice. We experimented with a small manually corrected lexicon and found that it helps in improving the intelligibility of the voice. Further it may be noted that the Indian English lexicon was bootstrapped from American English letter to sound rules and only 26.3% words needed corrections. This can be an efficient technique for creating a non-native pronunciation lexicon.

While a better pronunciation lexicon is crucial in building a good pronunciation model, it may be worthwhile to further investigate the individual roles of syllabification and stress assignment. Also, the use of new phoneme set designed to incorporate the peculiarities of an Indian English phonology can be part of the next steps.

6. Acknowledgements

We thank our collaborators Raghavendra E. and Bhaskar T. at IIIT Hyderabad in helping us conduct the second study. Also we thank fellow graduate students at CMU and students at IIIT Hyderabad for participating in the experiments.

7. References

[1] Balridge, J. “Linguistic and Social characteristics of Indian English”, *Language in India, Vol. 2, 2002.*

[2] Kominek, J. and Black, A. W., “The CMU Arctic speech databases”, *5th ISCA Speech Synthesis Workshop, Pittsburgh, PA*, 2004.

[3] “SCRIBE – Spoken Corpus of British English,” http://www.phon.ucl.ac.uk/resource/scribe/, 1990

[4] Carnegie Mellon University, “The CMU pronunciation dictionary”, http://www.speech.cs.cmu.edu, 2000

[5] Prahallad, L., Prahallad, K. and GanapathiRaju, M., “A Simple Approach for Building Transliteration Editors for Indian Languages”, *Journal of Zhejiang University Science, vol. 6A, no.11, pp. 1354-1361, Oct 2005.*

[6] Prahallad, K., Kumar, R., Sangal, R., “A Data-Driven Synthesis Approach for Indian Languages using Syllable as a basic unit,” *International Conference on NLP, Mumbai, India, 2002.*

[7] Black, A. W. and Lenzo, K. A., “Building Synthetic Voices – for Festvox 2.1,” *2007*, http://festvox.org/bsv/.

[8] Toda, T., Black, A. W., Tokuda, K., “Spectral Conversion Based on Maximum Likelihood Estimation Considering Global Variance of Converted Parameter,” *Intl. Conf. on Acoustics, Speech and Signal processing, Philadelphia, Pennsylvania, 2005.*

[9] Black, A. W., Lenzo, K. A., Pagel, V. “Issues in Building General Letter to Sound Rules,” *3rd ESCA Workshop on Speech Synthesis, pp. 77-80, Australia, 1998.*

[10] Black, A. W., and Taylor, P. A., “The Festival Speech Synthesis System: System documentation,” *Technical Report HCRC/TR-83, Human Communciation Research Centre, University of Edinburgh, Scotland, UK, 1997.*

[11] Black, A. W. and Taylor, P. A., “Automatically clustering similar units for unit selection in speech synthesis,” *Proceedings of Eurospeech97, vol. 2 pp.601-604, Rhodes, Greece, 1997.*

[12] Black, A. W., “CLUSTERGEN: A Statistical Parametric Synthesizer using Trajectory Modeling,” *Interspeech 2006 - ICSLP, Pittsburgh, PA., 2006.*

[13] Sen, A. and Samudravijaya, K., “Indian accent text-to-speech system for web browsing,” *Sadhana, Vol. 27, Part 1, pp. 113-126 February, 2002.*

[14] Sen, A., “Pronunciation Rules for Indian English Text-to-Speech system,” *ISCA Workshop on Spoken Language Processing, Mumbai, India, 2003.*

[15] Kumar, R., Kataria, A., Sofat, S., “Building Non-Native Pronunciation Lexicon for English Using a Rule Based Approach,” *International conference on NLP, Mysore, India, 2003.*

[16] Mullick, Y. J., Agrawal, S. S., Tayal, S., Goswami, M., "Text-to-phonemic transcription and parsing into mono-syllables of English text," *Journal of the Acoustical Society of America, Volume 115, Issue 5, pp. 2544, 2004.*

Creating German Unit Selection Voices for the MARY TTS Platform from the BITS Corpora

Marc Schröder and Anna Hunecke

DFKI GmbH
Saarbrücken, Germany
marc.schroeder@dfki.de, anna.hunecke@dfki.de

Abstract

The present paper reports on the creation of German unit selection voices from corpora which had been recorded and annotated previously in the BITS project. We describe the unit selection mechanism of our MARY TTS platform, as well as the tools for creating a synthesis voice from a speech corpus, and their application to the creation of German unit selection voices from the BITS corpora. Because of reservations concerning the mismatch of phonetic chains predicted by the German TTS components in MARY and the manually corrected database labels, we compared voices based on the manually corrected labels with voices based on automatic forced alignment labelling. We compute the diphone coverage for both types of voices and show that it is a reasonable approximation of the German diphone set. A preliminary evaluation confirms the expectations: while the manually corrected versions show a higher segmental accuracy, the automatically labelled versions sound more fluent.

1. Introduction

Unit selection synthesis is becoming a mature technology. Introduced in the mid-1990s [1, 2], it has matured over the last decade to the extent that now a regular competition, the Blizzard Challenge is being organised, where different data-driven synthesis algorithms are compared based on synthesis voices prepared from the same data. The vast majority of commercial TTS systems are based on unit selection technology; they are covering an increasing number of languages and voices.

Research systems, and particularly open-source systems, are less numerous. By far the most well-known system is Festival [3]; it contains two unit selection implementations, a cluster unit selection [4] and a generic unit selection [5]. The admirable Festvox toolkit provides support for creating custom synthesis voices, in the form of source code and documentation. The FreeTTS system [6] is a Java based reimplementation of code derived from Festival, and contains an implementation of the cluster unit selection algorithm. The BOSS system [7] implements a non-uniform unit selection method, which uses phrase- or word-sized units when these are found in the corpus, and reverts to smaller units otherwise. The MARY platform [8] became open source in early 2006, but until recently could generate audio only using the MBROLA [9] diphone synthesiser. A first unit selection component was added for US English [10] and released as open source.

Research on German speech synthesis, and German unit selection technology, seems to be progressing rather slowly. Indeed, there seem to be only a very limited number of German unit selection systems developed purely in Academia – we could only find two. The unit selection system BOSS [7] is available as open source; it comes with the Verbmobil database Lioba, which is somewhat tilted towards the domain of appointment negotiation. A general-domain German unit selection system based on Festival [3] has been developed at IMS Stuttgart [11] and continues to be developed in the Smartweb project [12]. However, it does not seem to be publicly available.

One important factor slowing down the development of unit selection systems in research labs is the cost associated with the creation of unit selection corpora. In order to lower that barrier, the project BITS [13] was funded to create unit selection voice databases, annotate them, and make them publicly available.

The present paper reports on the creation of publicly available German unit selection voices for the MARY TTS platform, based on the BITS corpora. The paper is organised as follows. We start by presenting the basic properties of the unit selection system developed in the framework of the MARY platform, and report on work in progress on an open-source toolkit for creating unit selection synthesis voices. We then describe the BITS corpora used as speech material for voice creation in the present paper, and report on our experiences building synthetic voices from these corpora.

2. The MARY unit selection system

2.1. The open source MARY TTS platform

MARY (Modular Architecture for Research on speech sYnthesis) is a platform for research, development and teaching on text-to-speech synthesis. Originally developed for German [8], it was extended to US English by incorporating some TTS modules from the FreeTTS project, and, as the result of a student project, to Tibetan. MARY uses an XML-based representation format for its data, which makes it possible to access intermediate processing states, and to connect it to other XML-based processing components [14].

Apart from being a research platform, MARY is also a stable Java server capable of multi-threaded handling of multiple client requests in parallel.

The design is highly modular. A set of configuration files, read at system startup, define the processing components to use. For example, the file `german.config` defines the German processing modules, `english.config` defines the English modules, etc. If both files are present in the configuration directory, both subsystems are loaded when starting the server. Each synthesis voice is defined by a configuration file: `german-mbrola-de7.config` loads the MBROLA voice de7, `english-arctic-jmk.config` the unit selection voice built from the Arctic recordings of speaker jmk [15], etc.

Each synthesis module has an input and an output format, which can be flexibly defined. This makes it extremely easy to define pipeline architectures for processing any given input format into one or more output formats, without explicitly stating the required chain of modules. Starting from the input format specified for the system input (e.g., plain text, SSML [16], etc.), the TTS system searches a path through the available processing components until it arrives at the requested output format (e.g., audio). Although this is a very simple mechanism for specifying a component architecture, it seems to be sufficient for the processing requirements of a TTS system.

For the generation of audio, MARY includes the concept of a collection of waveform synthesisers; these are defined in an extensible way through the MARY configuration files. Currently, the list of available waveform synthesisers includes the MBROLA diphone synthesiser; an LPC-based diphone synthesiser provided by FreeTTS; the MARY unit selection synthesiser covered in the present paper; and an experimental interpolating synthesiser, creating intermediate voices from two existing unit selection voices [17] using a spectral interpolation algorithm [18].

The architecture of the MARY platform as well as the English and Tibetan processing components are available under a liberal BSD-style license. The German processing components are available free of charge under a research license. By permission from the MBROLA team, MBROLA binaries and voices are provided with MARY under the MBROLA license.

The system runs under Windows, Linux, Solaris, and Mac OS X. A comfortable graphical installer can be downloaded from the MARY website. During installation, users can indicate which components they want to install; only these components are downloaded from the MARY page.

In order to avoid misconfigurations, the configuration files define a number of dependencies, which are checked automatically at every system startup. If a component is found to be missing, the system offers to download it from the MARY website.

2.2. Unit selection in MARY

The unit selection system in MARY implements a generic unit selection algorithm, combining the usual steps of tree-based pre-selection of candidate units, a dynamic programming phase combining weighted join costs and target costs, and a concatenation phase joining the selected units into an output audio stream.

Units to concatenate are uniform. An early version of the system [10] used phoneme units. After getting feedback at the Blizzard Challenge Workshop 2006, we switched to diphone units, because joining in the mid-section of phonemes is expected to introduce less discontinuities than joining at phoneme boundaries. For each target diphone, a set of candidate units is selected by separately retrieving candidates for each halfphone through a decision tree, and retaining only those that are part of the required diphone. When no suitable diphone can be found, the system falls back to halfphone units.

The most suitable candidate chain is obtained through dynamic programming, minimising a weighted sum of target costs and join costs. Both are themselves a weighted sum of component costs. Target costs cover the linguistic properties of units, and the way they match the linguistically defined target. In addition, acoustic target costs can be used. These are currently used for comparing a unit's duration and F0 to the ones predicted for the target utterance by means of regression trees trained on the voice data. In the future, we intend to use acoustic target costs to also cover expressivity-related acoustic measures, such as spectral tilt or other robust measures of voice quality.

Join costs are computed as a weighted sum of F0 difference and of spectral distance, computed as the absolute distance in 12-dimensional MFCC space. We had experimented with a step function for the F0 penalty, based on the reasoning that small F0 deviations can be corrected by a smoothing algorithm [10]; currently, we are using a linear cost function instead and avoid signal post-processing as it seems to degrade the overall quality.

Like all unit selection systems, we face the challenge of determining appropriate weights for the individual target and join cost components. As we have not yet developed a principled way of determining these weights, we have set a number of ad hoc values through iterative listening and adapting. The resulting weights give equal importance to join costs and to target costs, a higher importance to F0 continuity than to spectral continuity, and a higher importance to duration and F0 targets than to phonetic context.

After the chain of units minimising these costs is determined, the units are retrieved from a timeline file and concatenated using overlap-add of one pitch period at the unit boundaries. The timeline file currently contains uncompressed PCM audio data, but is designed in a way that makes it easy to use more efficient encodings in the future.

The system is reasonably efficient: it synthesises speech about ten times faster than real-time on a recent Core 2 Duo processor. Decision trees and feature vectors required for the cost computation are held in memory; audio data is retrieved from a file after selection.

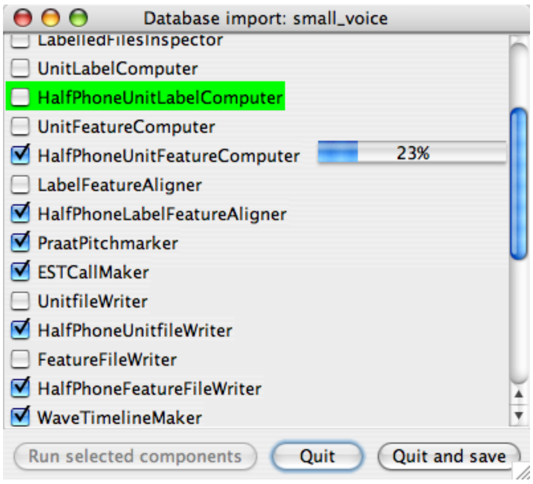
2.3. The voice creation toolkit in MARY

We are in the process of developing a toolkit for creating voices for MARY. We originally used the Festvox tools [19], and we continue to be deeply grateful to their creators for making them available to the community. However, it appears that some aspects of Festvox are tightly linked to the Festival system, and we felt that in the long run, the gain in control and flexibility justifies the development of our own voice creation toolkit.

The system combines an extensible list of "voice import components" in a graphical interface which is currently still very simple (see Figure 1). The user can select a series of import components, which are run in sequence. A progress bar is shown for the component which is currently running. After successful completion, the component is coloured in green; if processing fails, it is displayed in red, and processing of subsequent components is aborted. Configuration of non-default file system paths and special settings for the components is done via command-line options.

The voice import components that are currently available include components for automatic labelling using Sphinxtrain [20]; for importing text files in Festvox format; for predicting unit features with MARY; for making sure the unit labels and the feature chain predicted by MARY are properly aligned; for pitchmarking using Praat [21]; for the conversion of data into the compact format required by the MARY unit selection runtime system; for building classification trees for candidates using the wagon tool from the Edinburgh speech tools [22]; for pruning outliers from the generated trees; and for creating regression trees for duration and F0.

One of the most time-consuming tasks is the training of classification trees for the prediction of candidate units. Similarly to [4], we use acoustic distance between units as the impurity measure, and run wagon based on distance tables. In



□ LabelledFilesInspector
□ UnitLabelComputer
□ HalfPhoneUnitLabelComputer
□ UnitFeatureComputer
☑ HalfPhoneUnitFeatureComputer
□ LabelFeatureAligner
☑ HalfPhoneLabelFeatureAligner
☑ PraatPitchmarker
☑ ESTCallMaker
□ UnitfileWriter
☑ HalfPhoneUnitfileWriter
□ FeatureFileWriter
☑ HalfPhoneFeatureFileWriter
☑ WaveTimelineMaker

Figure 1: The MARY voice creation toolkit at work. In the situation shown, half-phone unit labels have been created successfully, unit features are being computed, and a number of components are scheduled for subsequent execution.

order to speed up the process on a multi-processor machine, the MARY CartBuilder component can run several wagon processes in parallel. Given the fact that the computation of acoustic distances is currently done in a single Java process, there is a limit to the number of wagon processes that should reasonably be started in parallel; we have experienced considerable speedup with running 3-5 wagon processes alongside one Java process on an 8-processor machine.

The MARY voice creation toolkit currently requires a considerable amount of expert knowledge in order to set paths correctly via command-line options and to select the right components for the task at hand. We intend to develop a more intuitive system providing groupings of the components that are usually required for a given task. For example, components working with halfphones are required for creating the necessary files to build classification trees for pre-selection of candidate units, but phone-sized units are needed for training regression trees for the prediction of duration and F0.

3. The BITS corpora

The BITS corpora were produced by the Bavarian Archive for Speech Signals (BAS) at Ludwig-Maximilians University, Munich, to provide a publicly available synthesis corpus for German. Two different kinds of corpora were recorded: logatome corpora for diphone synthesis and unit selection corpora. This paper only deals with the latter.

The unit selection part consists of 1683 sentences covering all German diphones and a few selected French and English diphones. A subset of the sentences was selected from a Newspaper Corpus (TAZ corpus) with a greedy algorithm. Additionally, semantically unpredictable sentences, provided by the IMS at University of Stuttgart, trade names and proverbs are contained in the set. Four speakers (two female, two male) were recorded with a close-talking microphone, a large membrane microphone and a laryngograph. The sentences were annotated with phonetic and prosodic labels automatically, then corrected by hand.

The corpus is distributed through the European Language Resources Association (ELRA) and can be ordered via the BAS website (`http://www.bas.uni-muenchen.de`).

4. German unit selection voices from the BITS corpora

4.1. Manual vs. automatic annotation

Since the BITS corpora have hand-corrected labels, they capture some phonetic detail, such as coarticulation effects and segmental reductions as they were realised by the speaker (e.g., Schwa elisions, nasal assimilations, or idiosyncratic devoicing). This poses a problem for the MARY system, since the phonemes predicted by MARY do not reflect these effects. As a result, even though a given syllable or word may be in the corpus, it may not be possible to retrieve the corresponding units. For example, one speaker frequently reduced Schwas: For the word “dunkel” (dark), the phonological form /dUNk@l/ was realised phonetically as [dUNkl]. A lookup of candidate units for /dUNk@l/ would need to find Schwa units from a different part of the corpus, even though the original word was available.

A proper solution to this problem would be a trainable postlexical phonological component, to be trained on the speech data from a given speaker in order to capture the speaker’s pronounciation rules. However, such a component is not yet realised in MARY.

In building synthetic voices for MARY from the BITS corpora, we therefore had two choices:

- use the existing manual annotation, knowing that suboptimal candidate units will be retrieved;
- use a fully automatic annotation created by forced alignment of the audio recordings with a phoneme chain created from the text using the MARY phonemisation component.

We decided to explore the trade-offs between both approaches by building two voices from each of the four databases: one with manual (M) and one with automatic (A) labels. We refer to the resulting voices as M1-4 and A1-4, respectively.

The expectation was that the A voices would show some segmental errors introduced by the uncorrected automatic labelling, but that overall the fluidity of the speech would be higher than for the M voices. In particular, it could be expected that the average length of segments joined would be higher for the A than for the M voices. The M voices, on the other hand, would be expected to have more accurate segmental pronounciations.

4.2. Voice creation

For the creation of the voices, the voice creation toolkit described in section 2.3 was used.

For the M (manually labelled) versions of the voices, additional voice import components were implemented which created labels and features based on the given labels. In the process, the phone labels of the annotation had to be mapped to the ones used by MARY, because some of the diacritics were not used by MARY, and some phone symbols were different. Also, in the BITS corpora, vowels followed by “6” (a-schwa) were annotated as diphthongs and had to be split up for MARY. For the computation of the features, first the phones and ToBI tones predicted by MARY were replaced with the actually annotated phones and tones. This modified version was then sent to MARY to compute the unit features needed for computing the target costs.

The automatic labels for the A versions of the voices were created with the components calling SphinxTrain and Sphinx2, using the phoneme chain predicted by MARY from the text. We enriched the MARY pronounciation lexicon to make sure that the text is transcribed as accurately as possible. In the textual form of the BITS corpus, we found 459 unknown words and 123 words interpreted as English words (many of them proper names). Out of these, we manually transcribed 338 of the unknown words, and 40 of the words recognised as English; the remaining words were transcribed properly by the MARY letter-to-sound components. The unit features for the A voices were fully based on MARY predictions from text.

After the labels and features were created, the usual voice building steps were performed for all voices: First, the pitchmarks were calculated from the laryngograph files, using Praat, and with reasonable estimates of the pitch range of each speaker to minimise the risk of octave jumps. Pitch-synchronous mel-frequency cepstral coefficient (MFCC) vectors were computed using the EST tools.

The units, unit features and audio data were converted into a format suitable for the efficient use in the run-time unit selection components. In addition to the purely symbolic unit feature predicted by MARY, the unit F0 and duration were included as acoustic unit features, in view of the computation of acoustic target costs. Join cost features were computed at unit boundaries, comprising 12 MFCCs plus F0, and stored in a file allowing to access them efficiently.

For each voice, regression trees were built to predict phone duration and initial, medial and final log F0 in each syllable, to be used as acoustic targets and potentially for signal post-processing.

For the pre-selection of candidate units, classification trees were built using acoustic similarity as the impurity measure. Acoustic similarity was computed as a combined measure consisting of duration, F0, and linearly time-stretched average Mahalanobis distance between MFCC frames. This tree-building approach is similar to the cluster unit selection algorithm proposed by Black and Taylor [4]; however, our leaves contain between 50 and 100 candidates, for which full target costs are computed at run-time. The classification trees contain halfphone units; for generating diphone candidates, candidates are looked up for both halfphones, and only those that belong to the needed diphone are retained. This method makes it simple to fall back to halfphones: when no instance of a given diphone is found, the two sets of halfphone candidates are retained.

A pruning algorithm was implemented to remove outliers from the leaves of the pre-selection tree. This is particularly useful with fully automatic labelled data, as it can identify some of the most obvious labelling errors. One important kind of outliers are units labelled as silence which are not actually silence; we apply an energy criterion to identify these, based on a silence cutoff value determined from an energy histogram. A second kind of outlier are units that are too long, e.g. because a long portion of silence was labelled to be part of the unit, or because of wrongly predicted phoneme chains, leading to several phonemes to be labelled as a single one. We use a cutoff of 200 ms maximum duration for a halfphone: every non-silence unit that is longer than this threshold is removed. A third kind of outlier are units that have extreme values in the probability ratings generated by wagon during tree training. These are also removed from the pre-selection tree.

We have observed that some of the problems arising from automatic labelling could be filtered out using this pruning step. This is reflected in the amount of data pruned: it lies between 0.9 and 1.2% of the units for the M voices, and between 1.5 and 2.1% of the units for the A voices. However, more subtle pronounciation deviations could not be identified using this approach.

In the runtime system, weights were fine-tuned to reach a balance between linguistic and acoustic target costs on the one hand, and join costs on the other hand. Even though the weights are normalised so that all target cost weights and all join cost weights sum to one, the fact that duration and F0 are currently not normalised makes it necessary to manually adjust the weights for each voice. We did this so as to make sure that target costs are about as high join costs on average, and acoustic target costs (duration + F0) are slightly higher than symbolic target costs (mainly phonetic context).

4.3. Phonetic coverage

One objective measure of the expected quality of a voice is the coverage of diphones as they occur in the language. Therefore, the phonetic coverage of the voices was measured both for the annotated phonemes and the phonemes predicted by MARY. To get an idea of how the coverage of the BITS corpora relates to the German language in general, the results were compared with the coverage of a large German corpus. For this purpose, we collected a textual corpus consisting of 978,269 sentences extracted from German ebooks from Project Gutenberg (`http://www.gutenberg.org`), and transcribed it fully automatically using the MARY phonemisation component.

For a phoneme set of 56 German phonemes, including some English and French xenophones, the phoneme coverage is 100% for the M voices, and 98% for the A voices, where the /T/ (voiceless English "th") is missing.

The diphone coverage varies slightly between the different voices, because for each voice, some of the sentences in the corpus could not be used for building the voice. Overall, the diphone coverage for the A voices, using the automatically predicted phonemes, is slightly worse (around 1690 diphones) than the coverage for the M voices (1770 diphones). Both figure are considerably lower than the number of different diphones found in the Gutenberg corpus (2306 diphones). It strikes the eye that these figures are substantially smaller than the number of $56 * 56 = 3136$ theoretically possible diphones – apparently, only around $2306/3136 = 73\%$ of these actually occur in German. Taking the Gutenberg figure as the reference, rather than the theoretically possible number of diphones, we can thus compute a diphone coverage of $1690/2306 = 73\%$ for the A voices and $1770/2306 = 77\%$ for the M voices.

To get an idea not only of the quantity but also of the quality of the diphone coverage in the BITS voices, we also looked at the distribution of the diphones. Figure 2a shows the distribution of the diphones in the Gutenberg corpus. It can be seen that the distribution follows Zipf's law, according to which the frequency of a word (or in this case, a diphone) is roughly inversely proportional to its rank in the frequency table.

Figures 2b and 2c show the relative frequencies of diphones in the BITS voices A1 and M1, respectively. The distribution curves for the other BITS voices look similar. Whereas the distribution of A1 is highly similar to the distribution of the Gutenberg corpus, M1 has substantially more outliers. Most of these are related to the Schwa elisions annotated in the BITS corpora: For example, the diphone "t_n" (arising by a reduction of /t@n/) occurs far more frequently in M1 than in the Gutenberg corpus, which is transcribed without phonological reduction.

Figure 3 shows a different way of comparing the diphone distribution in A1 and M1 to the Gutenberg corpus. The coverage ratio v shown in the figure is computed for each diphone

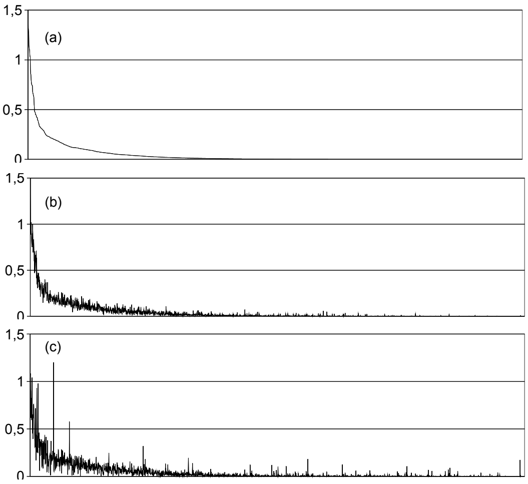


Figure 2: Relative frequency of diphones (a) in the Gutenberg corpus, (b) in voice A1, and (c) in voice M1. In all three, diphones are sorted on the X axis according to their frequency of occurrence in the Gutenberg corpus.

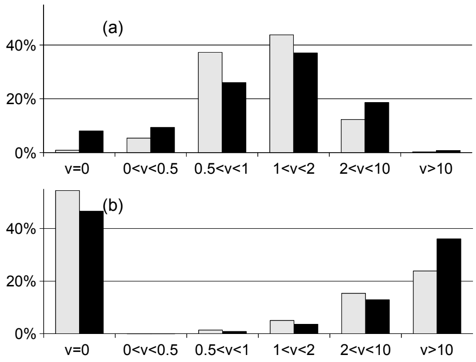


Figure 3: Coverage ratio of Gutenberg diphones for voices A1 (light) and M1 (dark), for (a) the frequent and (b) the rare half of Gutenberg diphones.

as the ratio of the relative frequency of the diphone in the voice and the relative frequency in the Gutenberg corpus. The graph shows the percentage of diphones with a given coverage ratio. Figure 3a represents the most frequent half of the Gutenberg diphones (the left half of Figure 2a), Figure 3b represents the least frequent half of the Gutenberg diphones (the right half of Figure 2a).

It can be seen that for the frequent German diphones, v values around 1 dominate, i.e. the coverage is close to the Gutenberg distribution. This is true for both voices, A1 and M1, with a slight advantage for the automatic labelling method which was also used for transcribing the text corpus. For the rare diphones, on the other hand, we see a clear dichotomy between diphones which are missing and diphones which are over-represented. Over-representation seems generally inevitable when trying to approximate a Zipf distribution with a much smaller corpus: even by occurring only once in the voice, a rare diphone already has a much higher relative frequency than the very low relative frequency in the large corpus.

4.4. Initial assessment of quality

Informal listening tests were performed to compare the quality of the voices, using ten example sentences for each of two text styles. First, news sentences were extracted from the web page of the German newspaper TAZ. Given the fact that the recording script was based on text material from the TAZ newspaper corpus, this can be considered a “within-domain” condition, which can be expected to lead to a relatively good synthesis quality. As a second text style, we used ten sentences from the fairy tale “Däumelieschen” available from the Gutenberg collection (`http://www.gutenberg.org`). This domain being different from the recording script, it can be considered a priori more challenging.

The first author, a trained phonetician, listened to the eight versions of each sentence, generated with the A and the M voice created from each of the four BITS corpora. Labels “+”, “0” and “-” were assigned to each utterance, where “+” indicated that only minor problems could be heard, “0” indicated audible prosodic deviations or minor segmental deviations, and “-” indicated clearly wrong segments. While the individual ratings are certainly subjective, and therefore are not reported in detail, some relatively clear patterns seem to emerge from this preliminary assessment.

Globally, more discontinuities can be heard in the M voices than in the A voices. This is reflected in the average length of consecutive unit stretches selected – 3.5 halfphones for M voices, and 4.0 halfphones for A voices. Furthermore, the M voices tend to sound a bit over-articulated. The A voices generally have a more natural prosody, but occasionally labelling errors are very prominent.

In the preliminary assessment, the A voices received better overall ratings than the M voices, reflecting the fact that prosodic naturalness and continuity were better for many of the sentences, and bad segments occurred only in a few sentences.

The news style sentences received better scores than the fairy tale sentences, lending support to the hypothesis that it is easier to synthesise within-domain material at good quality than material from a different type of text.

“-” labels, indicating segmental errors, occurred mostly for the A voices, but occasionally also for the M voices.

These first impressions provide an indication regarding the trade-off between the M and A voices which motivated the creation of both voices (see Section 4.1). Manual labelling leads to a considerable reduction of wrong segments in the output, and therefore remains a requirement for the professional creation of voice databases which cannot be replaced with filtering methods at the stage of tree pruning; however, when the predicted chain of target units does not reflect the kinds of postlexical phonological effects exhibited by the speaker, the continuity of the generated speech is reduced.

These findings suggest that it is not easy to choose between the M and the A version of a voice. Instead, it seems that the effort to develop a postlexical phonological component which can learn to map lexical-phonemic transcriptions to speaker-dependent surface-phonetic transcriptions would be well justified, because it could be expected to combine the benefits of both methods.

5. Conclusion

We have described the creation of unit selection synthesis voices in the MARY TTS platform, using the German corpora recorded for this purpose in the BITS project. Comparing voices created from the manually corrected labels in the database with voices created from fully automatic forced-alignment, we found systematic differences: higher segmental accuracy for the manually labelled voices, but more natural prosody and higher continuity for automatically labelled voices.

The resulting synthesis voices are work in progress, and can certainly be improved; but they are already quite intelligible German unit selection voices. Given the sparsity of publicly available unit selection systems for German, we will make the resulting voices available for download as soon as possible, under the same research license as the existing German MARY TTS components.

Future work will address various aspects of the current system. In the context of the present paper, the most obviously needed improvement is a trainable postlexical component. In addition, the general voice-building and unit selection methods will be improved, as time permits, along the following lines. Acoustic target and join costs should be computed in a normalised acoustic space, i.e. in z-scores. This will make it easier to set the weights for various target and join cost components. It will also allow us to reuse one speaker's prosody model with another speaker's voice, simply by setting the denormalisation coefficients to the new speaker's mean and standard deviation. Pooling training data from several voices for more robust prosody prediction is another option.

These developments are also in line with our mid-term goals of making progress towards parametrisable expressive speech synthesis. In this context, a major issue in view of high-quality signal modification and efficiency is the representation of the audio signal, e.g. as line spectrum pairs (LSP).

6. Acknowledgements

The work reported here was supported by the EU project HUMAINE (IST-507422), by the DFG project PAVOQUE, and by the ProFIT project IDEAS4Games. The authors would like to thank the BITS team for making their voice databases available, and for allowing us to distribute synthesis voices created from them; Sacha Krstulović for helping with the definition and implementation of file formats and voice import components; Mat Wilson for the design and implementation of a GUI for recording new voices; and Maximilian Kwapil for implementing the pruning algorithm for outliers in the classification tree.

7. References

[1] A. W. Black and N. Campbell, "Optimising selection of units from speech databases for concatenative synthesis," in *Proc. Eurospeech*, vol. 1, Madrid, Spain, 1995, pp. 581–584.

[2] A. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, Atlanta, Georgia, 1996.

[3] A. W. Black, P. Taylor, and R. Caley, "Festival speech synthesis system, edition 1.4," Centre for Speech Technology Research, University of Edinburgh, UK, Tech. Rep., 1999. [Online]. Available: http://www.cstr.ed.ac.uk/projects/festival

[4] A. W. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in *Proc. Eurospeech*, Rhodes/Athens, Greece, 1997.

[5] R. A. J. Clark, K. Richmond, and S. King, "Festival 2 – build your own general purpose unit selection speech synthesiser," in *Proc. 5th ISCA Speech Synthesis Workshop*, Pittsburgh, PA, USA, 2004, pp. 173–178.

[6] "Freetts 1.2," http://freetts.sourceforge.net, 2005.

[7] E. Klabbers, K. Stöber, R. Veldhuis, P. Wagner, and S. Breuer, "Speech synthesis development made easy: The Bonn Open Synthesis System," in *Proc. Eurospeech*, Aalborg, Denmark, 2001, pp. 521–524.

[8] M. Schröder and J. Trouvain, "The German text-to-speech synthesis system MARY: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, pp. 365–377, 2003. [Online]. Available: http://mary.dfki.de

[9] T. Dutoit, V. Pagel, N. Pierret, F. Bataille, and O. van der Vrecken, "The MBROLA project: Towards a set of high quality speech synthesisers free of use for non commercial purposes," in *Proc. ICSLP*, Philadelphia, USA, 1996.

[10] M. Schröder, A. Hunecke, and S. Krstulović, "OpenMary – open source unit selection as the basis for research on expressive synthesis," in *Proc. Blizzard Challenge'06*, 2006.

[11] A. Schweitzer, N. Braunschweiler, T. Klankert, B. Möbius, and B. Säuberlich, "Restricted unlimited domain synthesis," in *Proc. Eurospeech*, Geneva, Switzerland, 2003.

[12] D. Sonntag, R. Engel, G. Herzog, A. Pfalzgraf, N. Pfleger, M. Romanelli, and N. Reithinger, "Smartweb handheld – multimodal interaction with ontological knowledge bases and semantic web services," in *Proc. Intl Workshop on AI for Human Computing (AI4HC) at IJCAI*, Hyderabad, India, 2007.

[13] T. Ellbogen, F. Schiel, and A. Steffen, "The BITS speech synthesis corpus for German," in *Proc. LREC*, Lisbon, Portugal, 2004, pp. 2091–2094.

[14] M. Schröder and S. Breuer, "XML representation languages as a way of interconnecting TTS modules," in *Proc. ICSLP*, Jeju, Korea, 2004.

[15] J. Kominek and A. W. Black, "CMU ARCTIC speech databases," in *Proc. 5th ISCA Speech Synthesis Workshop*, Pittsburgh, PA, USA, 2004, pp. 223–224.

[16] M. R. Walker and A. Hunt, *Speech Synthesis Markup Language Specification*, W3C, 2001. [Online]. Available: http://www.w3.org/TR/speech-synthesis

[17] M. Schröder, "Interpolating expressions in unit selection," in *Proc. 2nd International Conference on Affective Computing and Intelligent Interaction (ACII'2007)*, Lisbon, Portugal, to appear.

[18] O. Turk, M. Schröder, B. Bozkurt, and L. Arslan, "Voice quality interpolation for emotional text-to-speech synthesis," in *Proc. Interspeech*, Lisbon, Portugal, 2005.

[19] A. W. Black and K. Lenzo, "Festvox: Building synthetic voices, edition 1.6," Language Technologies Institute, Carnegie Mellon University, PA, USA, Tech. Rep., 2002. [Online]. Available: http://www.festvox.org

[20] R. Mosur and K. A. Lenzo, *Sphinx-II User Guide*, CMU, http://cmusphinx.sourceforge.net/sphinx2/doc/sphinx2.html.

[21] P. Boersma and D. Weenink, "Praat: doing phonetics by computer." http://www.praat.org, 2007.

[22] S. King, A. W. Black, P. Taylor, R. Caley, and R. Clark, "Edinburgh speech tools library," http://www.cstr.ed.ac.uk/projects/speech_tools, 2003.

Regression Approaches to Voice Quality Control Based on One-to-Many Eigenvoice Conversion

Kumi Ohta, Yamato Ohtani, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano

Graduate School of Information Science, Nara Institute of Science and Technology, Japan
{kumi-o, yamato-o, tomoki, sawatari, shikano}@is.naist.jp

Abstract

This paper proposes techniques for flexibly controlling voice quality of converted speech from a particular source speaker based on one-to-many eigenvoice conversion (EVC). EVC realizes a voice quality control based on the manipulation of a small number of parameters, i.e., weights for eigenvectors, of an eigenvoice Gaussian mixture model (EV-GMM), which is trained with multiple parallel data sets consisting of a single source speaker and many pre-stored target speakers. However, it is difficult to control intuitively the desired voice quality with those parameters because each eigenvector doesn't usually represent a specific physical meaning. In order to cope with this problem, we propose regression approaches to the EVC-based voice quality controller. The tractable voice quality control of the converted speech is achieved with a low-dimensional voice quality control vector capturing specific voice characteristics. We conducted experimental verifications of each of the proposed approaches.

1. Introduction

Voice conversion (VC) is a technique for converting non-linguistic features such as speaker individuality while keeping the linguistic features. One of the most typical VC applications is speaker conversion that converts a certain speaker's voice into another speaker's voice [1]. This technique realizes a voice quality controller which converts one user's voice quality into another voice, which is very useful not only as an amusement device but also as a speech enhancement device for a speaking aid system recovering a disabled person's voice or as a hearing aid system to make speech sounds more intelligible.

Speech morphing [2] [3] is one of the techniques for constructing a voice quality controller. Input speech is usually converted by manipulating acoustic features such as fundamental frequency (F0) and spectral envelope in a simple manner, e.g., linear spectral warping. One advantage of this method is that it is easily used without training for a specific conversion model. On the other hand, this system allows very limited voice quality control of the converted speech. Since, in this case, the resulting voice quality strongly depends on the user's own voice quality, it is indeed difficult to convert any arbitrary voice into any desired speaker's voice.

As a technique for realizing a specific speaker's voice, a statistical approach to VC has been studied [1]. This framework trains a conversion model between a source speaker and a target speaker in advance, using parallel data consisting of utterance pairs of those two speakers [4]. A Gaussian mixture model (GMM) is often used as the conversion model [5]. The resulting model allows the determination of target speech parameters given the source parameters based on minimum mean square error (MMSE) estimation [5] or maximum likelihood estimation (MLE) [6] without any linguistic restrictions. Thus an arbitrary sentence uttered by the source speaker is rendered as an utterance by the target speaker. Because this framework needs training samples of the desired target speaker's voices, it is very difficult to construct a voice quality controller with the flexibility to vary voice quality of the converted speech.

As a novel VC framework, eigenvoice conversion (EVC) has been proposed [7] [8]. The eigenvoice is a popular speaker adaptation technique in the speech recognition area [9] [10]. It has also been applied to HMM-based TTS [11]. EVC realizes the conversion from a particular source speaker's voice into arbitrary speakers' voices (one-to-many EVC) or that from arbitrary speakers' voices into a particular target speaker's voice (many-to-one EVC). In one-to-many EVC, the eigenvoice Gaussian mixture model (EV-GMM) is trained in advance, using multiple parallel data sets consisting of utterance-pairs of the source speaker and multiple pre-stored target speakers. The voice quality of the converted speech is controlled by a small number of free parameters for eigenvectors capturing dominant voice characteristics extracted from pre-stored target speakers, which are called eigenvoices. Therefore, this framework allows us to control manually the voice quality of the converted speech. However, it is difficult to control intuitively the desired voice quality because each eigenvoice doesn't usually represent a specific physical meaning.

Recently, a multiple regression approach has been proposed for intuitively controlling voice quality of synthetic speech in the HMM/HSMM-based TTS [12] [13]. HMM/HSMM parameters are controlled with a low-dimensional vector called the **voice quality control vector**. Each component of the voice quality control vector captures specific characteristics of voice quality described by expression words such as sex, age and brightness. This paper proposes multiple regression approaches to EVC for constructing the voice quality controller that allows us to intuitively control the voice quality of the converted speech. We conducted experimental verifications for showing the advantages and disadvantages of each of them.

This paper is organized as follows. In **Section 2**, the framework of EVC is described. In **Section 3**, the proposed methods for constructing the EVC-based voice quality controller are described. In **Section 4**, experimental verifications are described. Finally, we summarize this paper in **Section 5**.

2. One-to-Many Eigenvoice Conversion (EVC) [7] [8]

The framework of EVC is shown in **Figure 1**.

2.1. Eigenvoice GMM (EV-GMM)

We use $2D$-dimensional acoustic features $\boldsymbol{X}_t = \left[\boldsymbol{x}_t^\top,\ \Delta\boldsymbol{x}_t^\top\right]^\top$ (source speaker's) and $\boldsymbol{Y}_t^{(s)} = \left[\boldsymbol{y}_t{}^{(s)\top},\ \Delta\boldsymbol{y}_t{}^{(s)\top}\right]^\top$ (the

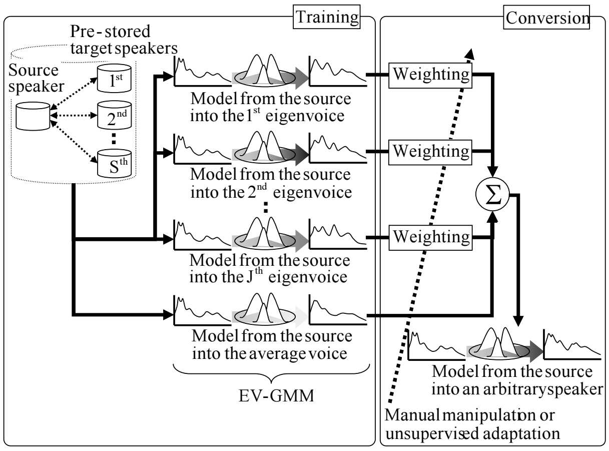

Training
Conversion
Pre-stored target speakers
Source speaker
1st
2nd
Sth
Model from the source into the 1st eigenvoice
Model from the source into the 2nd eigenvoice
Model from the source into the Jth eigenvoice
Model from the source into the average voice
EV-GMM
Weighting
Weighting
Weighting
Σ
Model from the source into an arbitrary speaker
Manual manipulation or unsupervised adaptation

Figure 1: *Framework of EVC.*

s^{th} pre-stored target speaker's) consisting of D-dimensional static and dynamic features at frame t, where $\top$ denotes transposition of the vector. Using a parallel training data set consisting of time-aligned source and target features $\boldsymbol{Z}_t^{(s)} = \left[\boldsymbol{X}_t^\top,\ \boldsymbol{Y}_t{}^{(s)\top}\right]^\top$ determined by Dynamic Time Warping (DTW), the EV-GMM $\boldsymbol{\lambda}^{(EV)}$ on joint probability density $P\left(\boldsymbol{Z}_t^{(s)}|\boldsymbol{\lambda}^{(EV)}\right)$ is trained in advance. The joint probability density is written as

$$P\left(\boldsymbol{Z}_t^{(s)}|\boldsymbol{\lambda}^{(EV)}\right) = \sum_{i=1}^{M} \alpha_i \mathcal{N}\left(\boldsymbol{Z}_t; \boldsymbol{\mu}_i^{(Z)},\ \boldsymbol{\Sigma}_i^{(ZZ)}\right),$$
$$\boldsymbol{\mu}_i^{(Z)} = \begin{bmatrix} \boldsymbol{\mu}_i^{(X)} \\ \boldsymbol{B}_i^{(Y)}\boldsymbol{w} + \boldsymbol{b}_i^{(Y)}(0) \end{bmatrix},$$
$$\boldsymbol{\Sigma}_i^{(ZZ)} = \begin{bmatrix} \boldsymbol{\Sigma}_i^{(XX)} & \boldsymbol{\Sigma}_i^{(XY)} \\ \boldsymbol{\Sigma}_i^{(YX)} & \boldsymbol{\Sigma}_i^{(YY)} \end{bmatrix}, \quad (1)$$

where $\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ shows the normal distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. The i^{th} mixture weight is α_i. The total number of mixtures is M. In the EV-GMM, the target mean vector for the i^{th} mixture is represented as a linear combination of a bias vector $\boldsymbol{b}_i^{(Y)}(0)$ and eigenvectors $\boldsymbol{B}_i^{(Y)} = \left[\boldsymbol{b}_i^{(Y)}(1),\ \boldsymbol{b}_i^{(Y)}(2),\ \cdots,\ \boldsymbol{b}_i^{(Y)}(J)\right]$. The number of eigenvectors is J. The target speaker individuality is controlled with only the J-dimensional weight vector $\boldsymbol{w} = \left[w(1),\ w(2),\ \cdots,\ w(J)\right]^\top$ for eigenvectors. Consequently, the EV-GMM has a parameter set $\boldsymbol{\lambda}^{(EV)}$ consisting of the single weight vector and parameters for individual mixtures such as the mixture weights, the source mean vectors, the bias and eigenvectors, and the covariance matrices. This paper employs diagonal covariance matrices for individual blocks, $\boldsymbol{\Sigma}_i^{(XX)}$, $\boldsymbol{\Sigma}_i^{(XY)}$, $\boldsymbol{\Sigma}_i^{(YX)}$, and $\boldsymbol{\Sigma}_i^{(YY)}$.

2.2. Training of EV-GMM

In order to train the EV-GMM, we use multiple parallel data sets. Each of them consists of utterance-pairs of the source speaker and one of the multiple pre-stored target speakers.

Firstly, we train a target independent GMM $\boldsymbol{\lambda}^{(0)}$ simultaneously, using all of the multiple parallel data sets as follows:

$$\boldsymbol{\lambda}^{(0)} = \arg\max \prod_{s=1}^{S}\prod_{t=1}^{T_s} P\left(\boldsymbol{Z}_t^{(s)}|\boldsymbol{\lambda}\right), \quad (2)$$

The number of feature vectors for the s^{th} speaker is T_s. The number of pre-stored target speakers is S. Secondly, we train each target dependent GMM $\boldsymbol{\lambda}^{(s)}$ by updating only target mean vectors $\boldsymbol{\mu}_i^{(Y)}$ of the target independent GMM $\boldsymbol{\lambda}^{(0)}$ using each of multiple parallel data sets as follows:

$$\boldsymbol{\lambda}^{(s)} = \arg\max \prod_{t=1}^{T_s} P\left(\boldsymbol{Z}_t^{(s)}|\boldsymbol{\lambda}\right). \quad (3)$$

Lastly, we determine the bias vector $\boldsymbol{b}_i^{(Y)}(0)$ and the eigenvectors $\boldsymbol{B}_i^{(Y)}$. We prepare a $(2D \times M)$-dimensional supervector $\boldsymbol{\mu}^{(Y)}(s) = \left[\boldsymbol{\mu}_1^{(Y)}(s)^\top,\ \boldsymbol{\mu}_2^{(Y)}(s)^\top,\ \cdots,\ \boldsymbol{\mu}_M^{(Y)}(s)^\top\right]^\top$ for each pre-stored target speaker by concatenating the target mean vectors $\boldsymbol{\mu}_i^{(Y)}(s)$ of the target dependent GMM $\boldsymbol{\lambda}^{(s)}$. We extract the eigenvectors with principal component analysis (PCA) for the supervectors. Consequently, the supervector is written as

$$\boldsymbol{\mu}^{(Y)}(s) \simeq \boldsymbol{B}^{(Y)}\boldsymbol{w}^{(s)} + \boldsymbol{b}^{(Y)},$$
$$\boldsymbol{B}^{(Y)} = \left[\boldsymbol{B}_1{}^{(Y)\top}, \boldsymbol{B}_2{}^{(Y)\top},\ \cdots,\ \boldsymbol{B}_M{}^{(Y)\top}\right]^\top,$$
$$\boldsymbol{b}^{(Y)} = \left[\boldsymbol{b}_1^{(Y)}(0)^\top, \boldsymbol{b}_2^{(Y)}(0)^\top,\ \cdots,\ \boldsymbol{b}_M^{(Y)}(0)^\top\right]^\top, \quad (4)$$
$$\boldsymbol{b}_i^{(Y)}(0)^\top = \frac{1}{S}\sum_{s=1}^{S}\boldsymbol{\mu}_i^{(Y)}(s), \quad (5)$$

where $\boldsymbol{w}^{(s)}$ consists of the principal components for the s^{th} pre-stored target speaker. We construct the EV-GMM $\boldsymbol{\lambda}^{(EV)}$ from the resulting bias and eigenvectors and the tied parameters, i.e., the mixture weights, the source mean vectors, and the covariance matrices of the target independent GMM. Now, various supervectors, i.e., the target mean vectors are created by varying only J $(< S \ll 2D \times M)$ free parameters of $\boldsymbol{w}$.

2.3. Problems in EV-GMM

The EV-GMM allows the control of voice quality of the converted speech by manually changing the weight vector. However, individual eigenvectors only capture dominant voice characteristics among pre-stored target speakers, which don't represent a specific physical meaning such as a masculine voice, a feminine voice, a hoarse voice, or a clear voice. Therefore, it is difficult to intuitively control the desired voice quality.

3. EVC-Based Voice Quality Controller

We propose regression approaches to the EVC-based voice controller for realizing the control of target mean vectors $\boldsymbol{\mu}_i^{(Y)}$ with the K-dimensional voice quality control vector $\boldsymbol{w}_e$ as follows:

$$\boldsymbol{\mu}_i^{(Y)} = \hat{\boldsymbol{B}}_i^{(Y)}\boldsymbol{w}_e + \hat{\boldsymbol{b}}_i^{(Y)}(0), \quad (6)$$

where

$$\hat{\boldsymbol{B}}_i^{(Y)} = \left[\hat{\boldsymbol{b}}_i^{(Y)}(1),\ \hat{\boldsymbol{b}}_i^{(Y)}(2),\ \cdots,\ \hat{\boldsymbol{b}}_i^{(Y)}(K)\right].$$

First, appropriate components of the voice quality control vector are manually assigned to each pre-stored target speaker. And then, the regression parameters $\hat{\boldsymbol{B}}_i^{(Y)}$ and $\hat{\boldsymbol{b}}_i^{(Y)}(0)$ are estimated by the following three methods: **A**) least squares estimation (LSE) of a regression matrix converting the voice quality control vector into principal components, **B**) LSE of a regression matrix converting the voice quality control vector into target mean vectors, and **C**) MLE of all parameters of EV-GMM under the condition of Eq. (6). **Figures 2**, **3**, and **4** show these proposed methods **A**, **B**, and **C**, respectively.

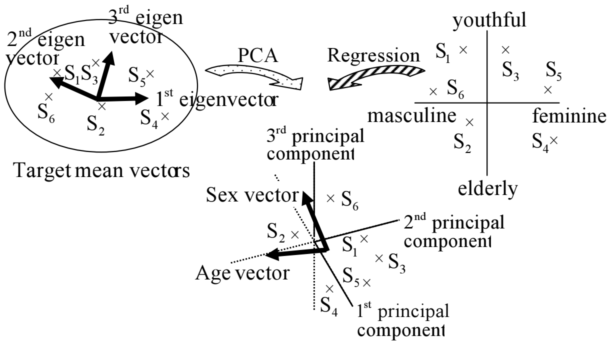


Figure 2: *Proposed method A.*

3.1. Proposed Method A: Regression of Principal Components on Voice Quality Control Vector

The target mean vectors of each pre-stored target speaker are efficiently represented as principal components by using eigenvectors. The proposed method A performs a regression of principal components on the voice quality control vector.

Principal components for the s^{th} target speaker $\boldsymbol{p}^{(s)}$ modeled by the following linear equation,

$$\begin{aligned} \boldsymbol{p}^{(s)} &\simeq \boldsymbol{R}\boldsymbol{w}_e^{(s)} + \boldsymbol{r} \\ &= \left[\ \boldsymbol{r}\ \boldsymbol{R}\ \right] \left[\begin{array}{c} 1 \\ \boldsymbol{w}_e^{(s)} \end{array} \right] \\ &= \boldsymbol{R}'\boldsymbol{w}_e^{(s)\prime}, \end{aligned} \tag{7}$$

where $\boldsymbol{R}$ is a regression matrix and $\boldsymbol{r}$ is a bias vector. $\boldsymbol{w}_e^{(s)}$ is the voice quality control vector for the s^{th} target speaker. In order to estimate the matrix $\boldsymbol{R}'$, we minimize the following error function:

$$\varepsilon_A^2 = \sum_{s=1}^{S} \left(\boldsymbol{p}^{(s)} - \boldsymbol{R}'\boldsymbol{w}_e^{(s)\prime}\right)^{\top} \left(\boldsymbol{p}^{(s)} - \boldsymbol{R}'\boldsymbol{w}_e^{(s)\prime}\right). \tag{8}$$

The LS estimate of $\boldsymbol{R}'$ is given by

$$\hat{\boldsymbol{R}}' = \boldsymbol{P}{\boldsymbol{W}_e}'^{\top} \left(\boldsymbol{W}_e'{\boldsymbol{W}_e}'^{\top}\right)^{-1}, \tag{9}$$

where

$$\begin{aligned} \boldsymbol{P} &= \left[\boldsymbol{p}^{(1)}, \boldsymbol{p}^{(2)}, \ \cdots, \ \boldsymbol{p}^{(S)}\right], \\ \boldsymbol{W}_e' &= \left[{\boldsymbol{w}_e}^{(1)\prime}, {\boldsymbol{w}_e}^{(2)\prime}, \ \cdots, \ {\boldsymbol{w}_e}^{(S)\prime}\right]. \end{aligned}$$

Therefore, using the obtained regression matrix and the bias vector, the regression parameters in Eq. (6) are written as

$$\begin{aligned} \hat{\boldsymbol{B}}_i^{(Y)} &= \boldsymbol{B}_i^{(Y)}\hat{\boldsymbol{R}}, \\ \hat{\boldsymbol{b}}_i^{(Y)}(0) &= \boldsymbol{B}_i^{(Y)}\hat{\boldsymbol{r}} + \boldsymbol{b}_i^{(Y)}(0). \end{aligned} \tag{10}$$

3.2. Proposed Method B: Regression of Target Mean Vectors on Voice Quality Control Vector

Voice characteristics to be controlled might not be properly represented as a linear combination of eigenvectors. If so, it is necessary to change the eigenvectors themselves. The proposed method B performs a regression of the target mean vectors on the voice quality control vector.

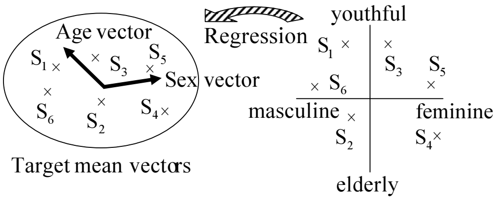


Figure 3: *Proposed method B.*

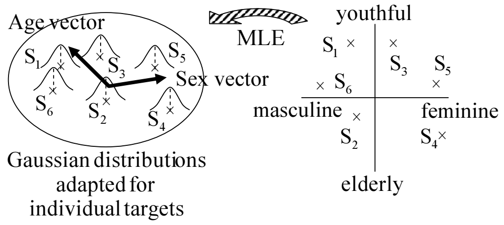


Figure 4: *Proposed method C.*

The target mean vector for the s^{th} target speaker $\boldsymbol{\mu}^{(Y)}(s)$ is modeled by

$$\begin{aligned} \boldsymbol{\mu}^{(Y)}(s) &\simeq \boldsymbol{B}^{(Y)}\boldsymbol{w}_e^{(s)} + \boldsymbol{b}^{(Y)}(0) \\ &= \left[\ \boldsymbol{b}^{(Y)}\ \boldsymbol{B}^{(Y)}\ \right] \left[\begin{array}{c} 1 \\ \boldsymbol{w}_e^{(s)} \end{array} \right] \\ &= \boldsymbol{B}^{(Y)\prime}\boldsymbol{w}_e^{(s)\prime}, \end{aligned} \tag{11}$$

In order to estimate the matrix $\boldsymbol{B}^{(Y)\prime}$, we minimize the following error function:

$$\varepsilon_B^2 = \sum_{s=1}^{S} \left(\boldsymbol{\mu}^{(Y)}(s) - \boldsymbol{B}^{(Y)\prime}\boldsymbol{w}_e^{(s)\prime}\right)^{\top} \left(\boldsymbol{\mu}^{(Y)}(s) - \boldsymbol{B}^{(Y)\prime}\boldsymbol{w}_e^{(s)\prime}\right). \tag{12}$$

The LS estimate of $\boldsymbol{B}^{(Y)\prime}$ is given by

$$\hat{\boldsymbol{B}}^{(Y)\prime} = \boldsymbol{\mu}^{(Y)}{\boldsymbol{W}_e}'^{\top} \left(\boldsymbol{W}_e'{\boldsymbol{W}_e}'^{\top}\right)^{-1}, \tag{13}$$

where

$$\boldsymbol{\mu}^{(Y)} = \left[\boldsymbol{\mu}^{(Y)}(1), \boldsymbol{\mu}^{(Y)}(2), \ \cdots, \ \boldsymbol{\mu}^{(Y)}(S)\right].$$

3.3. Proposed method C: MLE of EV-GMM Parameters

The desired voice quality might not always be realized by the methods mentioned above because voice quality of the converted speech is affected not only by the target mean vectors but also the other EV-GMM parameters. In order to realize more precise voice quality control, the proposed method C optimizes all of the EV-GMM parameters in the sense of ML under the condition that the weight vector is set to $\boldsymbol{w}_e^{(s)}$. This process is considered to be speaker adaptive training (SAT) [14] [15]. Most parameters of the EV-GMM in the previous methods are affected by acoustic variations of the pre-stored target speakers because they are from the target independent GMM. SAT reduces those variations by training the EV-GMM while considering the adaptation process.

The EV-GMM is trained by maximizing the likelihood of the adapted models for individual pre-stored target speakers as follows:

$$\hat{\boldsymbol{\lambda}}^{(EV)} = \arg\max_{\boldsymbol{\lambda}} \prod_{s=1}^{S}\prod_{t=1}^{T_s} P\left(\boldsymbol{Z}_t^{(s)} | \, \boldsymbol{\lambda}^{(EV)}, \, \boldsymbol{w}_e^{(s)}\right), \tag{14}$$

where the voice quality control vector $\boldsymbol{w}_e^{(s)}$ is employed in the adapted model for the $s^{\rm th}$ pre-stored target speaker. In order to estimate the EV-GMM parameters including the regression parameters, we maximize the following auxiliary function with the EM algorithm,

$$\begin{aligned} & Q\left(\boldsymbol{\lambda}^{(EV)},\, \hat{\boldsymbol{\lambda}}^{(EV)}\right) \\ & = \sum_{s=1}^{S}\sum_{i=1}^{M} \bar{\gamma}_i^{(s)} \log P\left(\boldsymbol{Z}_t^{(s)},\, m_i |\, \hat{\boldsymbol{\lambda}}^{(EV)},\, \boldsymbol{w}_e^{(s)}\right), \end{aligned} \tag{15}$$

where

$$\bar{\gamma}_i^{(s)} = \sum_{t=1}^{T_s} P\left(m_i |\, \boldsymbol{Z}_t^{(s)},\, \boldsymbol{\lambda}^{(EV)},\, \boldsymbol{w}_e^{(s)}\right). \tag{16}$$

Because it is difficult to estimate all parameters simultaneously, we estimate them at the following order,

$$\begin{aligned} & Q\left(\boldsymbol{\lambda}^{(EV)},\, \boldsymbol{\lambda}^{(EV)}\right) \\ & \leq Q\left(\boldsymbol{\lambda}^{(EV)},\, (\hat{\boldsymbol{B}}_i^{(Y)},\, \boldsymbol{b}_i^{(Y)}(0),\, \boldsymbol{\mu}_i^{(X)},\, \alpha_i,\, \boldsymbol{\Sigma}_i^{(zz)})\right) \\ & \leq Q\left(\boldsymbol{\lambda}^{(EV)},\, (\hat{\boldsymbol{B}}_i^{(Y)},\, \hat{\boldsymbol{b}}_i^{(Y)}(0),\, \hat{\boldsymbol{\mu}}_i^{(X)},\, \hat{\alpha}_i,\, \hat{\boldsymbol{\Sigma}}_i^{(zz)})\right), \end{aligned}$$

ML estimates of those parameters are written as

$$\begin{aligned} \hat{\boldsymbol{v}}_i &= \left(\sum_{s=1}^{S} \bar{\gamma}_i^{(s)} \boldsymbol{W}_s^\top \boldsymbol{\Sigma}_i^{(ZZ)^{-1}} \boldsymbol{W}_s\right)^{-1} \\ & \quad \times \left(\sum_{s=1}^{S} \boldsymbol{W}_s^\top \boldsymbol{\Sigma}_i^{(ZZ)^{-1}} \bar{\boldsymbol{Z}}_i^{(s)}\right), \end{aligned} \tag{17}$$

$$\hat{\alpha}_i = \frac{\sum_{s=1}^{S} \bar{\gamma}_i^{(s)}}{\sum_{i=1}^{M}\sum_{s=1}^{S} \bar{\gamma}_i^{(s)}}, \tag{18}$$

$$\begin{aligned} \hat{\boldsymbol{\Sigma}}_i^{(ZZ)} &= \frac{1}{\sum_{s=1}^{S} \bar{\gamma}_i^{(s)}} \sum_{s=1}^{S} \Big\{ \bar{\boldsymbol{V}}_i^{(s)} + \bar{\gamma}_i^{(s)} \hat{\boldsymbol{\mu}}_i^{(s)} \hat{\boldsymbol{\mu}}_i^{(s)^\top} \\ & \quad - \left(\hat{\boldsymbol{\mu}}_i^{(s)} \bar{\boldsymbol{Z}}_i^{(s)^\top} + \bar{\boldsymbol{Z}}_i^{(s)} \hat{\boldsymbol{\mu}}_i^{(s)^\top}\right)\Big\}, \end{aligned} \tag{19}$$

where

$$\begin{aligned} \bar{\boldsymbol{Z}}_i^{(s)} &= \begin{bmatrix} \bar{\boldsymbol{X}}_i^{(s)} \\ \bar{\boldsymbol{Y}}_i^{(s)} \end{bmatrix} \\ &= \begin{bmatrix} \sum_{t=1}^{T_s} P\left(m_i |\, \boldsymbol{Z}_t^{(s)},\, \boldsymbol{\lambda}^{(EV)},\, \boldsymbol{w}_e^{(s)}\right) \boldsymbol{X}_t^{(s)} \\ \sum_{t=1}^{T_s} P\left(m_i |\, \boldsymbol{Z}_t^{(s)},\, \boldsymbol{\lambda}^{(EV)},\, \boldsymbol{w}_e^{(s)}\right) \boldsymbol{Y}_t^{(s)} \end{bmatrix}, \end{aligned}$$

$$\boldsymbol{\Sigma}_i^{(ZZ)^{-1}} = \begin{bmatrix} \boldsymbol{P}_i^{(XX)} & \boldsymbol{P}_i^{(XY)} \\ \boldsymbol{P}_i^{(YX)} & \boldsymbol{P}_i^{(YY)} \end{bmatrix},$$

$$\bar{\boldsymbol{V}}_i^{(s)} = \sum_{t=1}^{T_s} P\left(m_i |\, \boldsymbol{Z}_t^{(s)},\, \boldsymbol{\lambda}^{(EV)},\, \boldsymbol{w}_e^{(s)}\right) \boldsymbol{Z}_t^{(s)} \boldsymbol{Z}_t^{(s)^\top},$$

$$\hat{\boldsymbol{\mu}}_i^{(s)} = \boldsymbol{W}_s \hat{\boldsymbol{v}}_i = \begin{bmatrix} \hat{\boldsymbol{\mu}}_i^{(X)} \\ \hat{\boldsymbol{B}}_i^{(Y)} \boldsymbol{w}_e^{(s)} + \hat{\boldsymbol{b}}_i^{(Y)}(0) \end{bmatrix},$$

$$\hat{\boldsymbol{v}}_i = \left[\hat{\boldsymbol{\mu}}_i^{(X)^\top},\, \hat{\boldsymbol{b}}_i^{(Y)}(0)^\top,\, \hat{\boldsymbol{b}}_i^{(Y)}(1)^\top,\, \cdots,\, \hat{\boldsymbol{b}}_i^{(Y)}(K)^\top\right]^\top,$$

$$\boldsymbol{W}_s = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & w_e^{(s)}(1)\boldsymbol{I} & w_e^{(s)}(2)\boldsymbol{I} & \cdots & w_e^{(s)}(K)\boldsymbol{I} \end{bmatrix},$$

and the matrix $\boldsymbol{I}$ is the $D \times D$ unit matrix. This paper employs the target independent GMM $\boldsymbol{\lambda}^{(0)}$ in Eq. (2) for calculating occupancies $\bar{\gamma}_i^{(s)}$ at the first E-step.

4. Experimental Verifications

4.1. Experimental Conditions

We used 30 speakers, 15 male and 15 female, as the pre-stored speakers. These speakers were included in the Japanese Newspaper Article Sentences (JNAS) database [16]. Each of them uttered a set of phonetically balanced 50 sentences. We used a female speaker not included in JNAS as the source speaker, who uttered the same sentence sets as uttered by the pre-stored speakers.

As the voice quality control vector, we used a 7-scaled categorical score (-3: very, -2: quite, -1: somewhat, 0: no preference, 1: somewhat, 2: quite, 3: very) for 5 Japanese word pairs expressing voice quality (masculine/feminine, hoarse/clear, elderly/youthful, thin/deep, and lax/tense), which were in the major expression word pairs extracted by Kido et al. [17, 18]. One Japanese female subject assigned these scores to each of the pre-stored target speakers by listening to natural speech samples of various sentences uttered by each of them. Scores for each word pair were normalized into the Z-score (zero mean and unit variance).

The STRAIGHT analysis method [19] was employed for the spectral extraction. The first through 24th mel-cepstral coefficients into which the extracted STRAIGHT spectrum were converted were used as the spectral parameter. The shift length was set to 5 ms. Sampling frequency was 16 kHz.

First, we trained the EV-GMM as described in **Section 2.2**. And then, its parameters were further updated with each proposed method. In the proposed method A, all 29 eigenvectors were employed with no loss of information. The number of mixtures of the EV-GMM was set to 128.

4.2. Objective Verification

In order to validate whether the resulting EV-GMM appropriately models a correspondence between the voice quality control vector and voice quality of the converted speech, we calculated the Euclidian distance between the manually assigned scores for each of pre-stored target speakers and the estimated ones, so that voice quality of the converted speech was similar to that of each pre-stored target speaker. We also calculated the correlation coefficient between those two kinds of scores. Since it was difficult to determine manually the best score settings, they were approximately determined with maximum likelihood eigen-decomposition (MLED) [9] for the target adaptation data in the same manner as described in [7]. We used 2 sentences for each pre-stored target speaker as the adaptation data in the score determination.

Figure 5 and **Figure 6** shows the Euclidean distances and the correlation coefficients between the manually assigned scores and the estimated scores. Moreover, an example of the assigned and the estimated scores on sex and hoarseness is shown in **Figure 7**. As a reference, each figure also shows the results of the reassigned scores, which were assigned by the same subject a second time on a different day.

We can see that the proposed method A doesn't work at all. These results show that the relationship between the voice

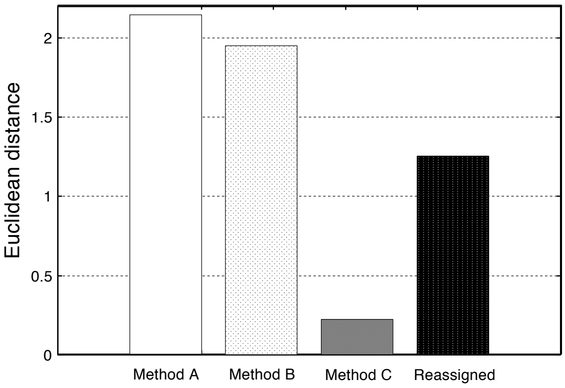


Figure 5: *Euclidean distances between the manually assigned scores and the estimated scores. We show averaged distance over 30 pre-stored target speakers.*

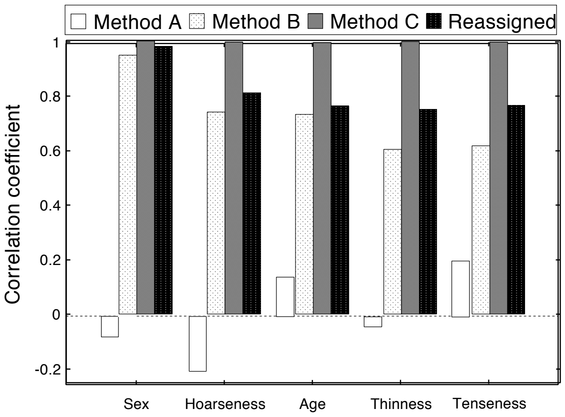


Figure 6: *Correlation coefficients between the manually assigned scores and the estimated scores.*

quality control vector and principal components is difficult to model as a linear conversion.

The proposed method B yields slightly lower distance and much better correlation coefficients than the proposed method A. It is necessary to estimate the regression matrix by directly modeling the relationship between the target mean vectors and the voice quality control vector instead of by using eigenvectors for designing the desired voice quality control. Although the proposed method B works much better than the proposed method A, the score distance is still large. Moreover, the estimated scores are quite different from the assigned ones for several speakers, as shown in **Figure 7**. These degradations are caused by a fact that the training criterion (LS) doesn't correspond to the conversion criterion (ML) and the trained parameters of the EV-GMM are limited to only regression parameters.

The proposed method C causes the best results. This is reasonable because the training criterion corresponds to the conversion criterion and every parameter of the EV-GMM is optimized so that the assigned scores capture the voice quality of the pre-stored target speakers as accurately as possible. However, we can observe from the results of the reassigned scores that even human judgment is not so consistent in scoring. These results imply that it is not always necessary to realize such strict score-consistency as found in the proposed method C.

4.3. Subjective Verification

To compare proposed method B with C, we conducted a preference test on the speech quality of the converted voices. Average voices were used as stimuli. Average voices were converted voices from the source speaker's voices when setting every component of the voice quality control vector to zero. Because the resulting bias vectors were almost the same in those methods, average voices produced by individual EV-GMMs had

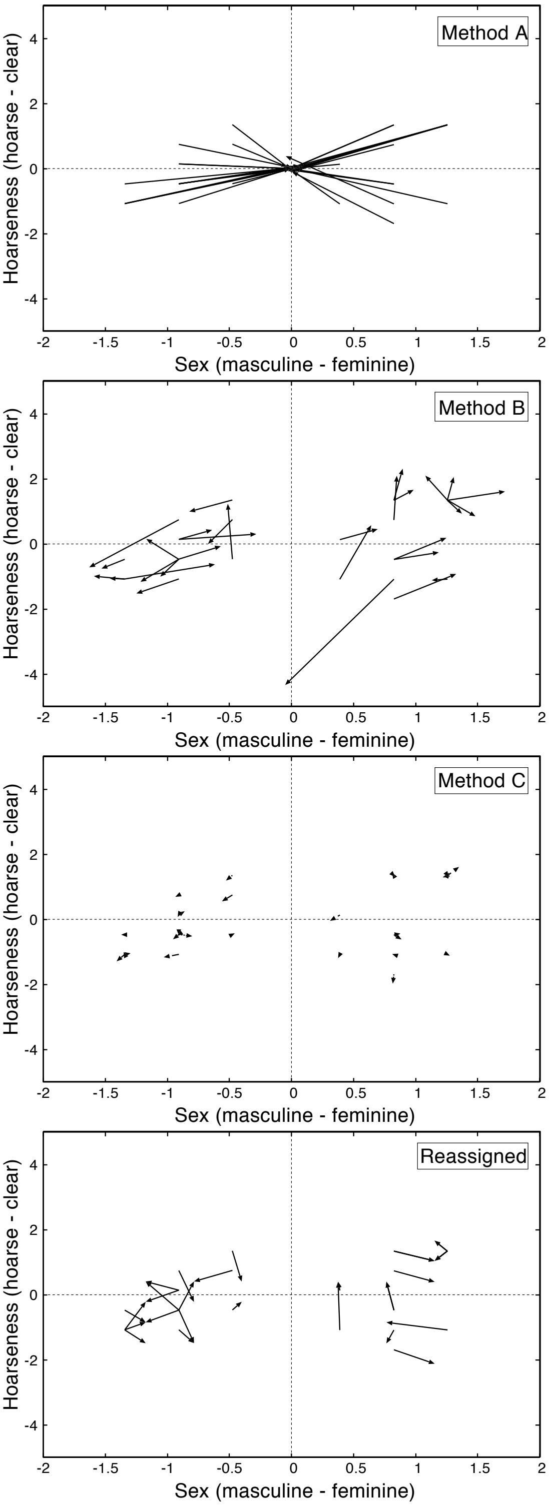


Figure 7: *An example of the manually assigned scores and the estimated scores for sex and hoarseness. The starting point of each arrow shows the manually assigned score, and its ending point shows the estimated score for each of pre-stored target speakers.*

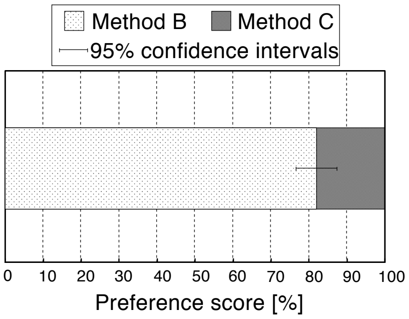


Figure 8: *Result of subjective evaluation.*

very similar speaker individuality. As for F0 conversion, a simple linear conversion based on mean values and standard deviations of log-scaled F0 was employed for converting the source to the average voice. In the preference test in those two proposed methods, we randomly presented a pair of the average voices produced by the EV-GMMs. The subjects were asked which sample sounded more natural. The 50 utterances not included in the training data were evaluated. The number of subjects was 5.

The result of the preference test is shown in **Figure 8**. It is observed that method B outperforms method C. The converted speech in method C sometimes has unstable sound quality. As mentioned above, method C causes the EV-GMM modeling the correspondence between the voice quality control vector and the pre-stored target voice quality as precisely as possible. It is possible that such strict modeling causes large projection errors on the high-dimensional acoustic space, especially if the low-dimensional space represented by the voice quality control vectors covers only a very limited sub-space. Those errors directly affect the estimation of the EV-GMM parameters. We have to cope with this problem in order to realize both high-quality and high-controllability of the EVC-based voice quality controller. It is also possible that the trained parameters in method C converge to local optima due to using inappropriate initial model, i.e., the target independent GMM in this paper.

5. Conclusions

We proposed regression approaches to the voice quality controller based on one-to-many eigenvoice conversion (EVC). First, the voice quality control vector was defined and proper component values of the vector were manually assigned to each of the pre-stored target speakers. And then, the eigenvoice Gaussian mixture model (EV-GMM) was trained so that voice characteristics of the pre-stored target speakers were properly represented by the voice quality control vectors. We conducted experimental verifications for showing advantages and disadvantages of each of the proposed methods.

6. Acknowledgements

This research was supported in part by MEXT's (the Japanese Ministry of Education, Culture, Sports, Science and Technology) e-Society project.

7. References

[1] H. Kuwabara and Y. Sagisaka, "Acoustic characteristics of speaker individuality: control and conversion," *Speech Communication*, Vol. 16, No. 2, pp. 165-173, 1995.

[2] M. Abe, "Speech morphing by gradually changing spectrum parameter and fundamental frequency," in *Proc. ICSLP 96*, Oct. 1996, Vol. 4, pp. 2235-2238.

[3] H. Banno, H. Hata, M. Morise, T. Takahashi, T. Irino, and H. Kawahara, "Implementation of realtime STRAIGHT speech manipulation system: Report on its implementation," *Acoust. Sci. & Tech.*, Vol. 28, No. 3, pp. 140-146, 2007.

[4] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," *J. Acoust. Soc. Jpn. (E)*, Vol. 11, No. 2, pp. 71-76, 1990.

[5] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech and Audio Processing*, Vol. 6, No. 2, pp. 131-142, 1998.

[6] T. Toda, A.W. Black, and K. Tokuda, "Spectral conversion based on maximum likelihood estimation considering global variance of converted parameter," in *Proc. ICASSP 2005*, Mar. 2005, Vol. 1, pp. 9-12.

[7] T. Toda, Y. Ohtani, and K. Shikano, "Eigenvoice conversion based on Gaussian mixture model," in *Proc. INTERSPEECH2006-ICSLP*, Sep. 2006, pp. 2446-2449.

[8] T. Toda, Y. Ohtani, and K. Shikano, "One-to-many and many-to-one voice conversion based on eigenvoices," in *Proc. ICASSP 2007*, Apr. 2007, pp. 1249-1252.

[9] R. Kuhn, J. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Trans. Speech and Audio Processing*, Vol. 8, No. 6, pp. 695-707, 2000.

[10] P. Kenny, G. Boulianne, and P. Dumouchel, "Maximum likelihood estimation of eigenvoices and residual variances for large vocabulary speech recognition tasks," in *Proc. ICSLP 2002*, Sep. 2002, pp. 57-60.

[11] K. Shichiri, A. Sawabe, T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-based speech synthesis," in *Proc. ICSLP 2002*, Sep. 2002, pp. 1269-1272.

[12] K. Miyanaga, T. Masuko, and T. Kobayashi, "A style control technique for HMM-based speech synthesis," in *Proc. INTERSPEECH2004-ICSLP*, Oct. 2004, pp. 1437-1440.

[13] M. Tachibana, T. Nose, J. Yamagishi, and T. Kobayashi, "A technique for controlling voice quality of synthetic speech using multiple regression HSMM," in *Proc. INTERSPEECH2006-ICSLP*, Sep. 2006, pp. 2438-2441.

[14] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. ICSLP 96*, Oct. 1996, Vol. 2, pp. 1137-1140.

[15] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Speaker adaptive training for voice conversion based on eigenvoice," *IEICE Tech. Rep.*, SP2006-40, pp. 31-36, 2006 [in Japanese].

[16] JNAS: Japanese Newspaper Article Sentences. http://www.mibel.cs.tsukuba.ac.jp/jnas/instruct.html

[17] H. Kido, and H. Kasuya, "Extraction of everyday expression associated with voice quality of normal utterance," *J. Acoust. Soc. Jpn.*, Vol. 55, No. 6, pp. 405-411, 1999 [in Japanese].

[18] H. Kido, and H. Kasuya, "Everyday expressions associated with voice quality of normal utterance —Extraction by perceptual evaluation—," *J. Acoust. Soc. Jpn.*, Vol. 57, No. 5, pp. 337-344, 2001 [in Japanese].

[19] H. Kawahara, I. Masuda-Katsuse, and A.de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds," *Speech Communication*, Vol. 27, No. 3-4, pp. 187-207, 1999.

An Evaluation of Many-to-One Voice Conversion Algorithms with Pre-Stored Speaker Data Sets

Daisuke Tani, Yamato Ohtani, Tomoki Toda, Hiroshi Saruwatari and Kiyohiro Shikano

Graduate School of Information Science, Nara Institute of Science and Technology, Japan
{daisuke-t, yamato-o, tomoki, sawatari, shikano}@is.naist.jp

Abstract

This paper describes an evaluation of many-to-one voice conversion (VC) algorithms converting an arbitrary speaker's voice into a particular target speaker's voice. These algorithms effectively generate a conversion model for a new source speaker using multiple parallel data sets of many pre-stored source speakers and the single target speaker. We conducted experimental evaluations for demonstrating the conversion performance of each of the many-to-one VC algorithms, including not only the conventional algorithms based on a speaker independent GMM and on eigenvoice conversion (EVC), but also new algorithms based on speaker selection and on EVC with speaker adaptive training (SAT). As a result, it is shown that an adaptation process of the conversion model improves significantly conversion performance, and the algorithm based on speaker selection works well even when using a very limited amount of adaptation data.

Index Terms: voice conversion, many-to-one VC, EVC, SAT, speaker selection

1. Introduction

Voice conversion (VC) is a technique for converting an input speaker's voice into another speaker's voice while keeping the linguistic information [1]. VC is applied to various applications such as modification of the synthetic speech of Text-to-Speech [2], bandwidth extension of cellular speech [3], and body-transmitted speech enhancement [4]. One of the most useful VC applications is cross-language VC [5,6], which converts speaker individuality across different two languages. This technique realizes a speech translation system or a CALL (Computer Assisted Language Learning) system synthesizing non-native language with user's own voice.

Statistical approaches are often employed in VC, the most popular being the conversion method based on the Gaussian Mixture Model (GMM) [7]. A GMM representing joint probability density of the source and the target speech parameters is trained in advance using parallel data consisting of utterance pairs of the source and the target speakers. The trained GMM allows the determination of the target speech parameters for the given source speech parameters based on minimum mean square error (MMSE) estimation [7] or maximum likelihood estimation (MLE) [8], without any linguistic restrictions. Thus an arbitrary sentence uttered by the source speaker is rendered as a sentence uttered by the target speaker. One essential problem of this approach is the need for parallel data for model training. Moreover, several tens of phoneme-balanced sentences whose total duration is around 3 to 5 minutes are generally required to train the GMM sufficiently for conversion performance. These constraints clearly limit VC applications.

In order to alleviate the problem of parallel training, two main approaches have been proposed. One is parallel data generation from non-parallel data; the other is model adaptation with non-parallel data. The first approach conducts frame alignment between the source and the target voices based on HMM state alignment [9] or unit selection [10]. In that case, conventional model training is employed with the resulting parallel data. This approach requires only the source and the target speakers' voices for the model training. Therefore, the same amount of training data is basically necessary. On the other hand, the latter approach uses other speaker's voices as a prior knowledge for training the model for the desired speaker pair. Mouchtaris et al. [11] proposed a non-parallel training method based on maximum likelihood constrained adaptation. The GMM trained with an existing parallel data set of a certain source and target speaker-pair is adapted for the desired source and target speakers separately. Lee et al. [12] proposed the adaptation method based on maximum a posteriori (MAP). In order to use a more reliable prior knowledge and reduce the amount of adaptation data, Toda et al. [13] proposed the eigenvoice conversion (EVC). EVC trains the eigenvoice GMM (EV-GMM) in advance using multiple parallel data sets consisting of utterance pairs of a single speaker and many pre-stored speakers. Effectively using the feature correlation between those speakers extracted from pre-stored parallel data sets enables unsupervised adaptation of EV-GMM for the desired speaker using only a few arbitrary sentences.

There are two novel VC frameworks to which EVC has been applied, i.e., one-to-many VC and many-to-one VC [14]. One-to-many VC converts the particular source speaker's voice into arbitrary speaker's one. On the other hand, many-to-one VC converts arbitrary speakers' voice into the particular speaker's voice. This paper focuses on many-to-one VC. It enables the conversion of any language uttered by an arbitrary speaker as utterances of the specific target speaker. It has been reported that not only the EV-GMM but also the source-independent GMM (SI-GMM) works in many-to-one VC [14] without any adaptation processes, but simply trained simultaneously using multiple parallel data sets of many pre-stored source speakers and the single target speaker. In this paper, we propose another many-to-one VC method based on speaker selection [15]. And, we introduce speaker adaptive training (SAT) [16] into EVC to further improve conversion performance. These many-to-one VC methods are compared with each other in both objective and subjective evaluations.

This paper is organized as follows. In **Section 2**, we describe the Many-to-One VC algorithms. In **Section 3**, we describe an experimental evaluation. Finally, we summarize this paper in **Section 4**.

2. Many-to-One VC algorithms

We describe four many-to-one VC algorithms based on 1) SI-GMM, 2) speaker selection, 3) EVC, 4) EVC with SAT. The difference among these algorithms is only in the way of constructing the conversion model for a given new source speaker. Every algorithm employs the MLE-based conversion method [8] using the resulting conversion model.

2.1. Many-to-One VC based on SI-GMM [14]

We use $2D$-dimensional acoustic features, $\boldsymbol{X}_t^{(s)} = [\boldsymbol{x}_t^{(s)\top}, \ \Delta\boldsymbol{x}_t^{(s)\top}]^\top$ (the s-th source speaker's), and $\boldsymbol{Y}_t = [\boldsymbol{y}_t^\top, \ \Delta\boldsymbol{y}_t^\top]^\top$ (target speaker's), consisting of D-dimensional static and dynamic features, where $\top$ denotes transposition of the vector. Joint probability density of $\boldsymbol{Z}_t^{(s)} = [\boldsymbol{X}_t^{(s)\top}, \ \boldsymbol{Y}_t^\top]^\top$ consisting of time-aligned source and target features determined by DTW is modeled with a GMM as follows:

$$P(\boldsymbol{Z}_t^{(s)}|\lambda) = \sum_{i=1}^{M} \alpha_i \mathcal{N}(\boldsymbol{Z}_t^{(s)}; \boldsymbol{\mu}_i^{(Z)}, \ \boldsymbol{\Sigma}_i^{(ZZ)}), \tag{1}$$

$$\boldsymbol{\mu}_i^{(Z)} = \begin{bmatrix} \boldsymbol{\mu}_i^{(X)} \\ \boldsymbol{\mu}_i^{(Y)} \end{bmatrix}, \ \boldsymbol{\Sigma}_i^{(ZZ)} = \begin{bmatrix} \boldsymbol{\Sigma}_i^{(XX)} & \boldsymbol{\Sigma}_i^{(XY)} \\ \boldsymbol{\Sigma}_i^{(YX)} & \boldsymbol{\Sigma}_i^{(YY)} \end{bmatrix}, \tag{2}$$

where $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ shows the normal distribuion with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. The i^{th} mixture weight is α_i. The total number of mixtures is M. This paper employs diagonal covariance matrices for individual blocks, $\boldsymbol{\Sigma}_i^{(XX)}$, $\boldsymbol{\Sigma}_i^{(XY)}$, $\boldsymbol{\Sigma}_i^{(YX)}$ and $\boldsymbol{\Sigma}_i^{(YY)}$.

The SI-GMM is trained with all multiple parallel data sets consisting of utterance-pairs of multiple pre-stored source speakers and one target speaker, as follows:

$$\lambda^{(0)} = \arg\max_{\lambda} \prod_{s=1}^{S} \prod_{t=1}^{T_S} P(\boldsymbol{Z}_t^{(s)}|\lambda), \tag{3}$$

where S is the number of pre-stored source speakers. **Figure 1** shows the previous training process.

In the conversion, the SI-GMM is directly used without any adaptation processes.

2.2. Many-to-One VC based on Speaker Selection

It is well known that phonemic spaces of a certain speaker often overlap with those of another speaker. Therefore, the SI-GMM might cause a conversion error especially for source speakers whose voice characteristics are quite different from those of the average voice among the pre-stored source speakers. Therefore, a model adaptation process for each source speaker is useful for alleviating this problem.

Speaker selection is one of the model adaptation techniques. **Figure 2** shows the previous training and the adaptation processes in many-to-one VC based on speaker selection. The conversion model is trained not with all parallel data sets but with only those consisting of the pre-stored source speakers whose voice characteristics are similar to those of the given source speaker. In order to reduce considerably the computational cost for this adaptation, we employ the single EM update of the SI-GMM, using pre-calculated sets of sufficient statistics for individual speaker-pairs. Note that this process allows unsupervised adaptation.

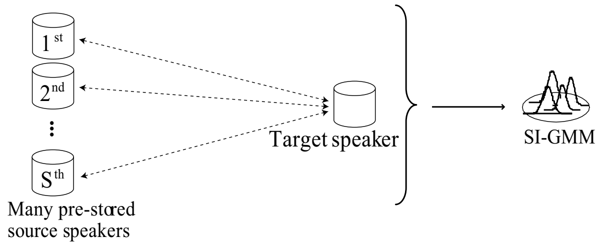


Figure 1: Previous training process in many-to-one VC based on SI-GMM.

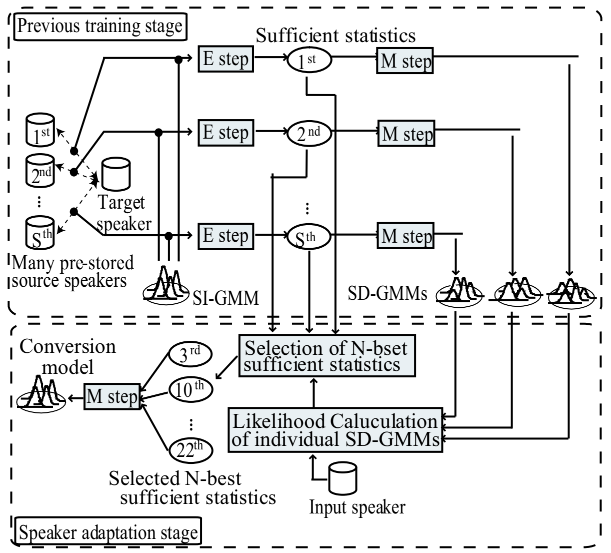


Figure 2: Previous training and adaptation processes in many-to-one VC based on speaker selection.

2.2.1. Previous training

A set of sufficient statistics for each speaker pair is computed with each parallel data set and the SI-GMM as follows:

$$\bar{\gamma}_i^{(s)} = \sum_{t=1}^{T_s} P(m_i|\boldsymbol{Z}_t^{(s)}, \ \lambda^{(0)}), \tag{4}$$

$$\bar{\boldsymbol{Z}}_i^{(s)} = \sum_{t=1}^{T_s} P\big(m_i|\boldsymbol{Z}_t^{(s)}, \lambda^{(0)}\big)\boldsymbol{Z}_t^{(s)}, \tag{5}$$

$$\bar{\boldsymbol{V}}_i^{(s)} = \sum_{t=1}^{T_s} P(m_i|\boldsymbol{Z}_t^{(s)}, \ \lambda^{(0)})\boldsymbol{Z}_t^{(s)}\boldsymbol{Z}_t^{(s)\top}. \tag{6}$$

Moreover, individual source dependent GMMs (SD-GMMs) are trained using each of the calculated sets of sufficient statistics for individual speaker pairs.

2.2.2. Unsupervised adaptation

Firstly, a likelihood of each SD-GMM for given adaptation data of the new source speaker $\boldsymbol{X}^{(org)}$ is calculated as follows:

$$L^{(s)} = \int P(\boldsymbol{X}^{(org)}, \boldsymbol{Y} | \lambda^{(s)}) d\boldsymbol{Y}. \tag{7}$$

Then, the likelihoods of individual SD-GMMs are sorted and the top N speaker-pairs are selected. The conversion model is generated using the N-best sets of sufficient statistics for the selected speaker pairs as follows:

$$\hat{\alpha}_i = \frac{\sum_{s \in S_N} \bar{\gamma}_i^{(s)}}{\sum_m \sum_{s \in S_N} \bar{\gamma}_i^{(s)}}, \tag{8}$$

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum_{s \in S_N} \bar{\boldsymbol{Z}}_i^{(s)}}{\sum_{s \in S_N} \bar{\gamma}_i^{(s)}}, \tag{9}$$

$$\hat{\boldsymbol{\Sigma}}_i = \frac{\sum_{s \in S_N} \bar{\boldsymbol{V}}_i^{(s)}}{\sum_{s \in S_N} \bar{\gamma}_i^{(s)}}, -\hat{\boldsymbol{\mu}}_i \hat{\boldsymbol{\mu}}_i^{\top}, \tag{10}$$

where $\hat{\alpha}_i$, $\hat{\boldsymbol{\mu}}_i$ and $\hat{\boldsymbol{\Sigma}}_i$ are the updated mixture weight, mean vector and covariance matrix, and S_N is a set of the N-best speakers.

2.3. Many-to-One VC based on EVC [14]

2.3.1. Eigenvoice Gaussian Mixture Model (EV-GMM)

The mean vector of the EV-GMM for many-to-one VC is given by

$$\boldsymbol{\mu}_i^{(Z)} = \begin{bmatrix} \boldsymbol{B}_i^{(X)} \boldsymbol{w} + \boldsymbol{b}_i^{(X)}(0) \\ \boldsymbol{\mu}_i^{(Y)} \end{bmatrix}. \tag{11}$$

The source mean vector for the i^{th} mixture is represented as a linear combination of a bias vector $\boldsymbol{b}_i^{(X)}(0)$ and representative vectors $\boldsymbol{B}_i^{(X)} = [\boldsymbol{b}_i^{(X)}(1), \boldsymbol{b}_i^{(X)}(2), \cdots, \boldsymbol{b}_i^{(X)}(J)]$. The number of the representative vectors is J. The source speaker individuality is controlled with only the J-dimensional weight vector $\boldsymbol{w} = [w(1), w(2), \cdots, w(J)]^{\top}$.

2.3.2. Training of EV-GMM

Figure 3 shows the previous training and the adaptation processes of many-to-one VC based on the EVC. Each SD-GMM is trained by updating only source mean vectors of the SI-GMM using each of the multiple parallel data sets. As a source dependent parameter, a supervector for each pre-stored source speaker is constructed by concatenating the source mean vectors of each of the SD-GMMs. The bias and representative vectors, i.e., eigenvectors are determined with principal component analysis (PCA) for all source speakers' supervectors. Finally, the EV-GMM is constructed from the resulting bias and representative vectors and parameters of the SI-GMM.

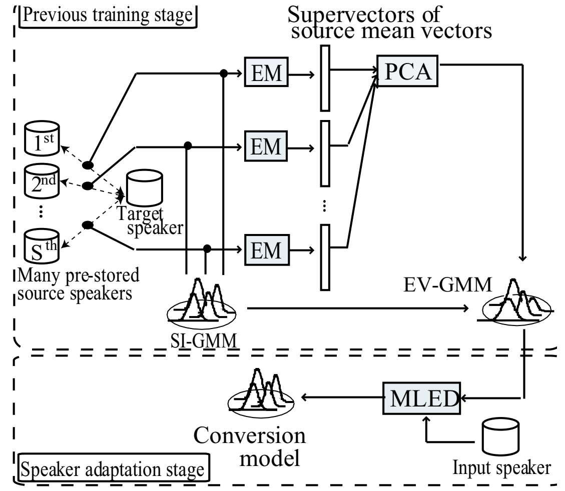


Figure 3: Previous training and adaptation processes in many-to-one VC based on EVC .

2.3.3. Unsupervised Adaptation of EV-GMM

The EV-GMM is adapted for arbitrary speakers by estimating the optimum weight vector for given speech samples without any linguistic information. The weight vector is estimated so that the likelihood of the marginal distribution for a time sequence of the given source features $\boldsymbol{X}^{(org)}$ is maximized as follows:

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \int P(\boldsymbol{X}^{(org)}, \boldsymbol{Y} | \lambda^{(EV)}) d\boldsymbol{Y}. \tag{12}$$

This estimation is performed with EM algorithm.

2.4. Many-to-One VC based on EVC with SAT

The tied parameters of the PCA-based EV-GMM are from the SI-GMM. They are affected by acoustic variations of many pre-stored source speakers. Especially, source covariance values are much larger than those of the SD-GMM. They would cause performance degradation of the adapted EV-GMM.

In order to train an appropriate canonical EV-GMM, we apply speaker adaptive training (SAT) to the EV-GMM training. **Figure 4** shows the previous training and the adaptation processes of many-to-one VC based on the EVC with SAT. The canonical EV-GMM is trained by maximizing the following likelihood of the adapted models for individual pre-stored source speakers,

$$\hat{\lambda}^{(EV)}(\hat{\boldsymbol{w}}_1^S) = \arg\max_{\lambda} \prod_{s=1}^{S} \prod_{t=1}^{T_s} P\Big(\boldsymbol{Z}_t^{(s)} | \lambda^{(EV)}(\boldsymbol{w}_s)\Big), \tag{13}$$

where $\lambda^{(EV)}(\boldsymbol{w}_s)$ denotes the adapted model for the s-th pre-stored source speaker with the weight vector $\boldsymbol{w}_s$. SAT estimates both canonical EV-GMM parameters $\hat{\lambda}^{(EV)}$ and a set of weight vectors for pre-stored source speakers $\hat{\boldsymbol{w}}_1^S = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_S]$. The estimation is performed with EM algorithm by maximizing the following auxiliary function:

$$Q\Big(\lambda^{(EV)}(\boldsymbol{w}_1^S), \hat{\lambda}^{(EV)}(\hat{\boldsymbol{w}}_1^S)\Big) = \sum_{s=1}^{S} \sum_{i=1}^{M} \hat{\gamma}_i^{(s)} \log P(m_i, \boldsymbol{Z}^{(s)} | \hat{\lambda}^{(EV)}(\hat{\boldsymbol{w}}_s)), \tag{14}$$

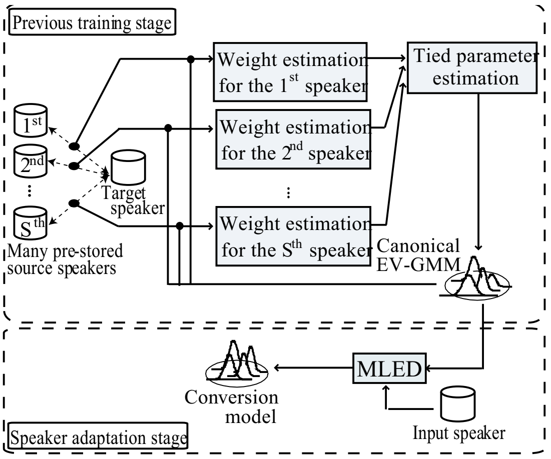

Previous training stage
1st
2nd
⋮
Sth
Many pre-stored source speakers
Target speaker
Weight estimation for the 1st speaker
Weight estimation for the 2nd speaker
⋮
Weight estimation for the Sth speaker
Tied parameter estimation
Canonical EV-GMM
MLED
Conversion model
Input speaker
Speaker adaptation stage

Figure 4: Previous training and adaptation processes of many-to-one VC based on EV-GMM with SAT .

where

$$\hat{\gamma}_i^{(s)} = \sum_{t=1}^{T_s} P(m_i|\boldsymbol{Z}_t^{(s)}, \lambda^{(EV)}(\boldsymbol{w}_s)).$$

It is difficult to update all parameters simultaneously because some depend on others. Therefore, each parameter of EV-GMM is updated as follows:

$$\begin{aligned}
&Q\Big(\lambda^{(EV)}(\boldsymbol{w}_1^S), \lambda^{(EV)}(\boldsymbol{w}_1^S)\Big)\\
\le &Q\Big(\lambda^{(EV)}(\boldsymbol{w}_1^S), (\hat{\boldsymbol{w}}_1^S, \boldsymbol{B}_i^{(X)}, \boldsymbol{b}_i^{(X)}(0), \boldsymbol{\mu}_i^{(Y)}, \alpha_i, \boldsymbol{\Sigma}_i^{(zz)})\Big)\\
\le &Q\Big(\lambda^{(EV)}(\boldsymbol{w}_1^S), (\hat{\boldsymbol{w}}_1^S, \hat{\boldsymbol{B}}_i^{(X)}, \hat{\boldsymbol{b}}_i^{(X)}(0), \hat{\boldsymbol{\mu}}_i^{(Y)}, \hat{\alpha}_i, \boldsymbol{\Sigma}_i^{(zz)})\Big)\\
\le &Q\Big(\lambda^{(EV)}(\boldsymbol{w}_1^S), (\hat{\boldsymbol{w}}_1^S, \hat{\boldsymbol{B}}_i^{(X)}, \hat{\boldsymbol{b}}_i^{(X)}(0), \hat{\boldsymbol{\mu}}_i^{(Y)}, \hat{\alpha}_i, \hat{\boldsymbol{\Sigma}}_i^{(zz)})\Big).
\end{aligned}$$

The ML estimate of the weight vector for the s-th pre-stored source speaker is written as

$$\begin{aligned}
\hat{\boldsymbol{w}}_s = &\left(\sum_{i=1}^{M} \bar{\gamma}_i^{(s)} \boldsymbol{B}_i^{(X)^\top} \boldsymbol{P}_i^{(XX)} \boldsymbol{B}_i^{(X)}\right)\\
&\times \left[\sum_{i=1}^{M} \Big\{ \boldsymbol{B}_i^{(X)^\top} \boldsymbol{P}_i^{(XY)} \big(\bar{\boldsymbol{Y}}_i^{(s)} - \bar{\gamma}_i^{(s)} \boldsymbol{\mu}_i^{(Y)}\big)\right.\\
&\left. + \boldsymbol{B}_i^{(X)^\top} \boldsymbol{P}_i^{(XX)} \big(\bar{\boldsymbol{X}}_i^{(s)} - \bar{\gamma}_i^{(s)} \boldsymbol{b}_i^{(X)}(0)\big)\big)\Big\}\right],
\end{aligned} \tag{15}$$

where

$$\bar{\boldsymbol{Z}}_i^{(s)} = \begin{bmatrix} \bar{\boldsymbol{X}}_i^{(s)} \\ \bar{\boldsymbol{Y}}_i^{(s)} \end{bmatrix} = \begin{bmatrix} \sum_{t=1}^{T_s} P\big(m_i|\boldsymbol{Z}_t^{(s)}, \lambda^{(EV)}(w_s)\big)\boldsymbol{X}_t^{(s)} \\ \sum_{t=1}^{T_s} P\big(m_i|\boldsymbol{Z}_t^{(s)}, \lambda^{(EV)}(w_s)\big)\boldsymbol{Y}_t^{(s)} \end{bmatrix},$$

$$\boldsymbol{\Sigma}_i^{(ZZ)^{-1}} = \begin{bmatrix} \boldsymbol{P}_i^{(XX)} & \boldsymbol{P}_i^{(XY)} \\ \boldsymbol{P}_i^{(YX)} & \boldsymbol{P}_i^{(YY)} \end{bmatrix}.$$

ML estimates of the tied parameters are written as

$$\hat{\alpha}_i = \frac{\displaystyle\sum_{s=1}^{S} \gamma_i^{(s)}}{\displaystyle\sum_{i=1}^{M}\sum_{s=1}^{S} \gamma_i^{(s)}}, \tag{16}$$

$$\hat{\boldsymbol{v}}_i = \left(\sum_{s=1}^{S} \bar{\gamma}_i^{(s)} \hat{\boldsymbol{W}}_s^\top \boldsymbol{\Sigma}_i^{(ZZ)^{-1}} \hat{\boldsymbol{W}}_s\right)^{-1} \left(\sum_{s=1}^{S} \hat{\boldsymbol{W}}_s^\top \boldsymbol{\Sigma}_i^{(ZZ)^{-1}} \bar{\boldsymbol{Z}}_i^{(s)}\right), \tag{17}$$

$$\begin{aligned}
\hat{\boldsymbol{\Sigma}}_i^{(ZZ)} = &\frac{1}{\sum_{s=1}^{S} \bar{\gamma}_i^{(s)}} \sum_{s=1}^{S} \Big\{ \bar{\boldsymbol{V}}_i^{(s)} + \bar{\gamma}_i^{(s)} \hat{\boldsymbol{\mu}}_i^{(s)} \hat{\boldsymbol{\mu}}_i^{(s)^\top}\\
&- \Big(\hat{\boldsymbol{\mu}}_i^{(s)} \bar{\boldsymbol{Z}}_i^{(s)^\top} + \bar{\boldsymbol{Z}}_i^{(s)} \hat{\boldsymbol{V}}_i^{(s)^\top}\Big)\Big\},
\end{aligned} \tag{18}$$

where

$$\bar{\boldsymbol{V}}_i^{(s)} = \sum_{t=1}^{T_s} P\big(m_i|\boldsymbol{Z}_t^{(s)}, \lambda^{(EV)}(\boldsymbol{w}_s)\big)\boldsymbol{Z}_t^{(s)} \boldsymbol{Z}_t^{(s)^\top},$$

$$\hat{\boldsymbol{\mu}}_i^{(s)} = \hat{\boldsymbol{W}}_s \hat{v}_i = \begin{bmatrix} \hat{\boldsymbol{B}}_i^{(X)} \hat{\boldsymbol{w}}_s + \hat{\boldsymbol{b}}_i^{(X)}(0) \\ \hat{\boldsymbol{\mu}}_i^{(Y)} \end{bmatrix},$$

$$\hat{\boldsymbol{v}}_i = \Big[\hat{\boldsymbol{\mu}}_i^{(X)^\top}, \{\hat{\boldsymbol{b}}_i^{(X)}(0)\}^\top, \{\hat{\boldsymbol{b}}_i^{(X)}(1)\}^\top, \cdots, \{\hat{\boldsymbol{b}}_i^{(X)}(J)\}^\top, \Big]^\top,$$

$$\hat{\boldsymbol{W}}_s = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \hat{w}_1^{(s)}\boldsymbol{I} & \hat{w}_2^{(s)}\boldsymbol{I} & \ldots & \hat{w}_J^{(s)}\boldsymbol{I} \end{bmatrix},$$

and the matrix $\boldsymbol{I}$ is $D \times D$ unit matrix.

3. Experimental Evaluations

3.1. Experimental conditions

We used 160 speakers, 80 male and 80 female, included in the Japanese Newspaper Article Sentences (JNAS) database [17] as the pre-stored source speakers. Each of them uttered a set of phonetically balanced 50 sentences. We used a male speaker not included in JNAS as the target speaker in many-to-one VC. He uttered the same sentence sets as uttered by the pre-stored speakers. We used 10 test speakers, 5 male and 5 female, who were not included among the pre-stored speakers. They uttered 53 sentences that were also not included in the pre-stored data sets. The number of adaptation sentences was varied from 1 to 32. The remaining 21 sentences were used for evaluations. More detailed conditions are described in [13,14].

We used the 1st through 24th mel-cepstral coefficients obtained from the smoothed spectrum analyzed by STRAIGHT [18] as a spectral parameter. The frame shift was set to 5 ms. A simple linear conversion with means and standard deviations of log-scaled F_0 of the source and the target speakers was employed in the F_0 conversion.

In many-to-one VC based on EVC with/without SAT, the number of representative vectors of the EV-GMM was set to 159. In the MLED-based unsupervised adaptation, the first E-step was conducted with the SI-GMM. And then, the next E-steps were conducted with the adapted EV-GMM. The number of times of EM iteration was set to 10. Note that only a few times of EM iteration also works well because rapid convergence is usually obtained.

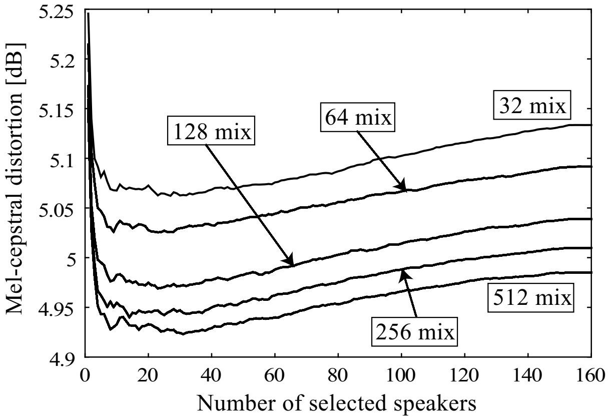


Figure 5: Mel-cepstral distortion as a function of the number of selected pre-stored speakers in many-to-one VC based on speaker selection. The number of mixtures is set to 32, 64, 128, 256 and 512. Only one sentence is used for the adaptation.

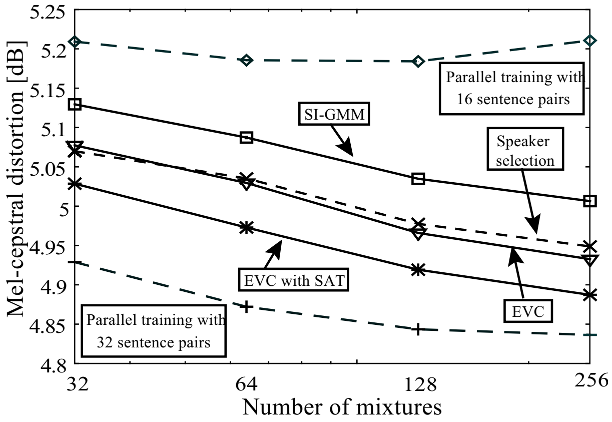


Figure 6: Mel-cepstral distortion as a function of the number of mixtures. The number of adaptation sentences is set to 2. Results of the conventional parallel training [14] are also shown as references

3.2. Objective evaluations

We conducted objective evaluations using the mel-cepstral distortion between the converted and the target mel-cepstra as an evaluation measure. The averaged distortion over all test speakers was 8.11 [dB] before the conversion.

Figure 5 shows the mel-cepstral distortion as a function of the number of selected pre-stored speakers in many-to-one VC based on speaker selection. Note that results when selecting 160 pre-stored speakers are the same as those in many-to-one VC based on SI-GMM. The adaptation method based on speaker selection improves conversion accuracy, compared with the method based on the SI-GMM, because the adapted GMM models an acoustic space of the given source speaker more properly than does the SI-GMM. We can see that the best conversion accuracy is achieved when selecting around 20 to 40 pre-stored speakers. Rapid degradation of conversion accuracy is observed when setting the number of the selected speakers much smaller. Using directly the conversion model for another speaker does not obviously work even if, among the pre-stored speakers, his voice characteristics are the most similar to those of the given source speaker. Therefore, it is important to construct the conversion model properly, covering the acoustic space of the given source speaker by mixing multiple speakers' data sets. The number of selected speakers is set to 27 in many-to-one VC based on speaker selection in the following evaluations.

Figure 6 shows mel-cepstral distortion as a function of the number of mixtures. Each many-to-one VC algorithm with the adaptation process outperforms that based on the SI-GMM. Many-to-one VC based on speaker selection allows the adaptation of every parameter of the conversion model. However, its adaptation mechanism is rougher than that of EVC, in terms of using a constant rate of mixing data sets of the selected pre-stored speakers. On the other hand, EVC estimates the best mixing rate, i.e., weights for eigenvectors, in the sense of ML, although it allows only the adaptation of source mean vectors. Consequently, the conversion performance in speaker selection is comparable to that in EVC. SAT optimizes tied parameters of the EV-GMM considering the adaptation process. Therefore, the performance of EVC is obviously improved by applying SAT into the EV-GMM training.

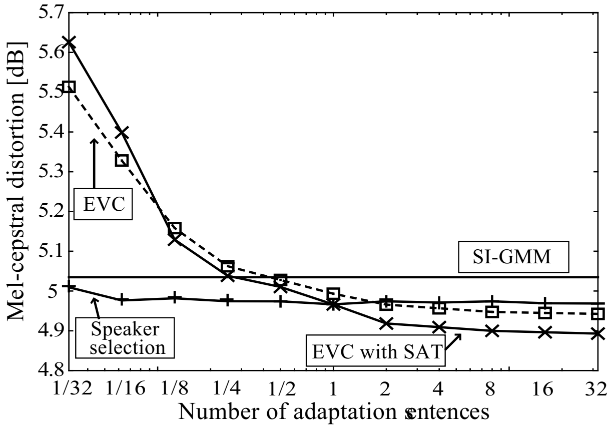


Figure 7: Mel-cepstral distortion as a function of the number of adaptation sentences. The number of mixtures is set to 128.

Figure 7 shows the mel-cepstral distortion as a function of the number of adaptation sentences varying from 1/32 to 32 sentences. In the 1/32 sentence, only one of 32 blocks into which one sentence is divided is used as adaptation data. It includes only around 32 frames, whose total duration is around 0.16 seconds. EVC obviously improves performance compared with the SI-GMM, when using one or more adaptation sentences. Introducing SAT into EVC improves further performance. However, when decreasing the adaptation data less than one sentence, conversion accuracy starts to degrade rapidly due to over-adaptation. Although reducing the number of representative vectors alleviates the over-adaptation problem, similar degradation tendencies were still observed in another experiment which in not shown here. On the other hand, speaker selection still keeps the performance improvements compared with the SI-GMM even when decreasing the adaptation data less than one sentence. Compared with EVC, speaker selection is more robust against the amount of adaptation data. These results show that the best adaptation method differs according to the amount of adaptation data.

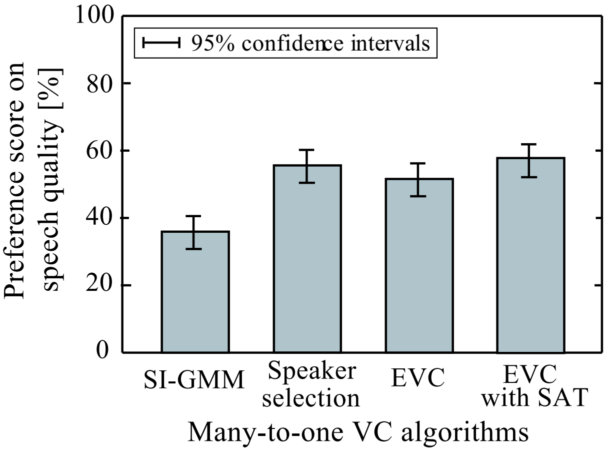


Figure 8: Result of subjective evaluation in many-to-one VC.

3.3. Subjective evaluation

We conducted subjective evaluation of speech quality of the converted voices. Four many-to-one VC methods were evaluated in the preference test. The number of mixtures was set to 128. The number of adaptation sentences was set to 2. The number of subjects was 6. We randomly presented a pair of the converted voices from two different methods. The subjects were asked which sample sounded more natural. Each subject evaluated 120 sample-pairs including every pair-combination of the four methods.

Figure 8 shows the result of the preference test. Three adaptation methods significantly improve the converted speech-quality compared with the method based on the SI-GMM. Each adaptation process alleviates unstable sounds of the converted speech sometimes caused by the SI-GMM. This result is very similar as shown in the objective evaluations.

4. Conclusions

This paper described many-to-one voice conversion (VC) algorithms that convert an arbitrary speaker's voice into a particular target speaker's voice. We conducted an experimental evaluation of many-to-one VC algorithms, using not only the conventional methods based on the source independent GMM (SI-GMM) and on EVC, but also two new methods based on speaker selection and EVC with speaker adaptive training (SAT). Results of objective and subjective evaluations showed that, in many-to-one VC, the adaptation process results in a better conversion model than the SI-GMM. Moreover, an algorithm based on speaker selection worked well with very little amount of adaptation data.

5. Acknowledgements

This research was supported in part by MEXT's (the Japanese Ministry of Education, Culture, Sports, Science and Technology) e-Society project.

6. References

[1] H. Kuwabara, and Y. Sagisaka. Acoustic characteristics of speaker individuality control and conversion. *Speech Communication*, Vol. 16, No. 2, pp. 165-173, 1995.

[2] A. Kain and M.W. Macon. Spectral voice conversion for text-to-speech synthesis. *Proc. ICASSP*, pp. 285-288, Seattle, USA, May 1998.

[3] K-Y Park and H. S. Kim. Narrowband to wideband conversion of speech using GMM based transformation.*Proc. ICASSP*, pp. 1843-1846, 2000.

[4] M. Nakagiri, T. Toda, H. Kashioka and K. Shikano. Improving body Transmitted Unvoiced Speech with Statistical Voice Conversion. *Proc. INTERSPEECH2006-ICSLP*, pp. 2270-2273, Pittsburgh, USA, Sep. 2006.

[5] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara. Voice conversion through vector quantization. *J. Acoust. Soc. Jpn. (E)*, Vol. 11, No. 2, pp. 71-76, 1990.

[6] M. Mashimo, T. Toda, H. Kawanami, K. Shikano and N Campbell. Cross-language voice conversion using bilingual database. *IPSJ Journal*, Vol.43, No.7, pp.2177-2185, July 2002.

[7] Y. Stylianou, O. Cappe and E.Moulines. Continuous probabilistic transform for voice conversion. *IEEE Trans. on Speech and Audio Processing*, Vol. 6, no. 2, pp. 131-142, Mar. 1998.

[8] T. Toda, A.W. Black and K. Tokuda. Spectral conversion based on maximum likelihood estimation considering global variance of converted parameter. *Proc. ICASSP*, pp. 9-12, 2005.

[9] H. Ye and S. Young. Quality-enhanced voice morphing using maximum likelihood transformations. *IEEE Trans. ASLP*, Vol. 14, No. 4, pp. 1301-1312, 2006.

[10] D. Sundermann, H. Hoge, A. Bonafonte, H. Ney, A. Black and S. Narayanan. Text-independent voice conversion based on unit selection. *Proc. ICASSP*, pp. 81-84, 2006.

[11] A. Mouchtaris, J. Spiegel, and P. Mueller. Non-parallel training for voice conversion by maximum likelihood constrained adaptation. *Proc. ICASSP*, pp. 1-4, May. 2004

[12] C.H. Lee and C.H. Wu. Map-based adaptation for speech conversion using adaptation data selection and non-parallel training. *Proc. ICSLP*, pp. 1164-1167, Sept. 2006.

[13] T. Toda, Y. Ohtani and K. Shikano. Eigenvoice Conversion Based on Gaussian Mixture Model. *Proc. ICSLP*, pp. 2446-2449, Sept. 2006.

[14] T. Toda, Y. Ohtani and K. Shikano. One-to-Many and Many-to-One Voice Conversion Based on Eigenvoices. *Proc. ICASSP*, Vol. 4, pp. 1249-1252, Apr. 2007.

[15] S. Yoshizawa, A. Baba, K. Matsunami, Y. Mera, M. Yamada and K. Shikano. Unsupervised speaker adaptation based on sufficient HMM statistics of selected speakers. *Proc.ICASSP2001*, pp. 341-344, May 2001.

[16] T. Anastasakos, J. McDonough, R. Schwartz and J. Makhoul. A compact model for speaker-adaptive training. *Proc. ICSLP*, Vol. 2, 1996.

[17] JNAS: Japanese Newspaper Article Sentences *http://www.milab.is.tsukuba.ac.jp/jnas/instruct.html*

[18] H. Kawahara, I. Masuda-Katsuse, and A.de Cheveigue. Restructuring speech representations using a pitch adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. *Speech Communivation*, Vol. 27, No. 3-4, pp. 187-207, 1999.

Towards an Improved Modeling of the Glottal Source in Statistical Parametric Speech Synthesis

João P. Cabral, Steve Renals, Korin Richmond and Junichi Yamagishi

The Centre for Speech Technology Research
University of Edinburgh,UK

jscabral@inf.ed.ac.uk, s.renals@ed.ac.uk, korin@cstr.ed.ac.uk, jyamagis@inf.ed.ac.uk

Abstract

This paper proposes the use of the Liljencrants-Fant model (LF-model) to represent the glottal source signal in HMM-based speech synthesis systems. These systems generally use a pulse train to model the periodicity of the excitation signal of voiced speech. However, this model produces a strong and uniform harmonic structure throughout the spectrum of the excitation which makes the synthetic speech sound buzzy. The use of a mixed band excitation and phase manipulation reduces this effect but it can result in degradation of the speech quality if the noise component is not weighted carefully. In turn, the LF-waveform has a decaying spectrum at higher frequencies, which is more similar to the real glottal source excitation signal.

We conducted a perceptual experiment to test the hypothesis that the LF-model can perform as well as or better than the pulse train in a HMM-based speech synthesizer. In the synthesis, we used the mean values of the LF-parameters, calculated by measurements of the recorded speech. The result of this study is important not only regarding the improvement in speech quality of these type of systems, but also because the LF-model can be used to model many characteristics of the glottal source, such as voice quality, which are important for voice transformation and generation of expressive speech.

Index Terms: LF-model, Statistical parametric speech synthesis, HMM-based speech synthesis

1. Introduction

Glottal source modeling has been commonly used in rule-based speech synthesizers, since they are fully parametric. For example, the formant synthesizer proposed by Klatt and Klatt uses the KLGLOTT88 [1] model, which permits the control of several glottal parameters such as the open quotient, breathiness and spectral tilt. Concatenative synthesizers model the glottal source by inverse filtering, but unit-selection synthesizers typically aim to avoid signal processing and simply concatenate the speech units in order to obtain better speech quality. Thus, this type of system does not permit flexibility to control any glottal parameters besides the fundamental frequency.

Methods for speech transformation usually use inverse filtering to separate the speech signal into vocal tract and excitation components. Typically, voice quality transformations are performed on the spectrum of the vocal tract but a model of the glottal source can also be used to simulate different aspects of voice quality by controlling the glottal parameters, such as in [2].

Emerging applications, such as dialogue systems or virtual characters, demand expressive speech which is difficult to obtain with unit-selection synthesis. To generate expressive speech unit-selection requires larger speech databases, e.g. a speech corpus recorded with different emotions. However, the recordings are costly and demanding to conduct.

Statistical speech synthesis generates high-quality speech and is fully parametric, e.g. [3] and [4]. The high degree of parametric flexibility can overcome the limitations of concatenation synthesizers to generate variable speech. Compared with formant speech synthesizers, one great advantage of HMM-based synthesizers is that the parameters are automatically obtained from training data. Typically, the features used are the spectrum and F_0, which is controlled with a binary pulse. However, a drawback of this approach is that the synthetic speech is characterized by a buzzy quality. This is explained by the strong harmonic structure of the pulse signal at higher frequencies when compared with the true glottal source signal. To reduce this effect, more recent versions of this approach, such as [5], use the high-quality STRAIGHT [6] method for analysis and synthesis, and a multi-band mixed excitation with phase manipulation of the periodic pulse component.

In the work described here, we employ a more parametric model of the excitation (described in Section 2) than the traditional pulse train used in the HMM-based speech synthesis. The goal of doing this is to improve the naturalness of the synthetic speech and to enhance the parametrization of the glottal source. The glottal parameters may be used to better model and transform effects related to voice quality and the speech characteristics of the speaker. For example, in [7], the authors showed that F_0 is strongly correlated with the glottal parameters. Thus, the control and modeling of these parameters could also improve the naturalness of the synthetic speech.

2. Glottal source model

We have used the LF-model [8] because, in general, it gives a good approximation of the differentiated glottal volume velocity (DGVV) which we intend to model. It has also been extensively studied and used in speech research so we could find and compare different techniques to extract the glottal features.

2.1. LF-model

The model we use is divided into three parts and is given by Equation 1. Figure 1 shows the LF-waveform and the glottal parameters. The first part of the model is described by an exponentially increasing sine wave that starts at the opening instant of the vocal folds, t_o, and ends at the instant of maximum negative amplitude, t_e. The second branch is given by a decaying exponential function that models the closure after the abrupt flow termination. The time constant t_a is the duration from t_e to the point where a tangent to the exponential at $t = t_e$ hits

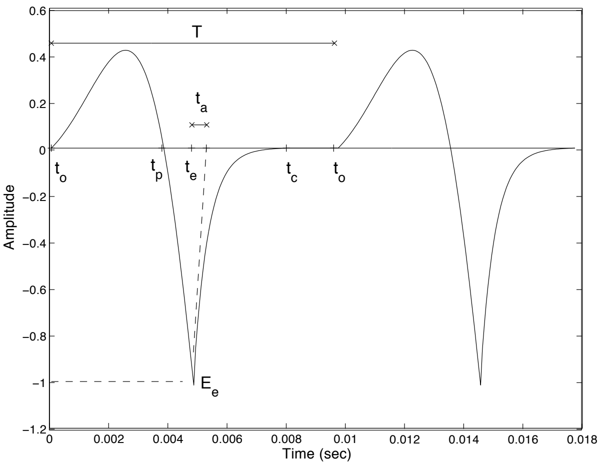


Figure 1: Segment of the LF-model waveform with the representation of the glottal parameters of Equation 1 during one period.

the time axis and measures the abruptness of the closure. The exponential part ends in the zero t_c. For simplification, it is usually assumed that $t_c - t_0 = T$, i.e., the fundamental period. Instead, we consider the glottal folds can be totally closed for a longer duration, from t_c until the end of the period. The other two parameters are the instant of maximum airflow t_p and the excitation amplitude E_e.

$$e(t) = \begin{cases} E_0 e^{\alpha t} \sin(w_g t), & 0 \leq t \leq t_e \\ -\frac{E_e}{\epsilon t_a}[e^{-\epsilon(t-t_e)} - e^{-\epsilon(t_c - t_e)}], & t_e < t \leq t_c \\ 0, & t_c < t \leq T \end{cases} \tag{1}$$

where $w_g = \frac{\pi}{t_p}$.

The parameters ϵ and α can be calculated from Equation 1 by imposing $e(t_e) = E_e$ and the energy balance $\int_0^T e(t) = 0$.

The LF-model can also be described by other parameters which are related with properties of the glottal flow in the frequency domain. The most relevant parameters are the open quotient (OQ), speed quotient (SQ), and the return quotient (RQ), which can be calculated from the basic time domain parameters as follows [9]:

$$OQ = \frac{t_e + t_a}{T} \tag{2}$$

$$SQ = \frac{t_p}{t_e - t_p} \tag{3}$$

$$RQ = \frac{t_a}{T} \tag{4}$$

In the spectral domain, the LF-model can be stylized by three asymptotic lines with +6dB/oct, -6dB/oct and -12db/oct slopes [10]. Figure 2 shows this spectral representation. The crossing point of the first two lines corresponds to a peak (called glottal spectral peak) at the frequency F_g. The last line is due to the spectral tilt which contributes with an additional -6dB/oct above the frequency F_c. The frequency F_g can be calculated as in [11]:

$$F_g = \frac{1}{2\pi O_q T} \sqrt{\frac{e(\alpha_m)}{j(\alpha_m)}} \tag{5}$$

where $j(\alpha_m)$ and $e(\alpha_m)$ are functions of the asymmetry coefficient $\alpha_m = SQ/(1 + SQ)$. F_c depends on several glottal parameters and it can be computed as described in [12]. However, it mostly depends on the glottal parameter t_a and it can be approximated by a simpler expression given in [8]:

$$F_c = \frac{1}{t_a 2\pi} \tag{6}$$

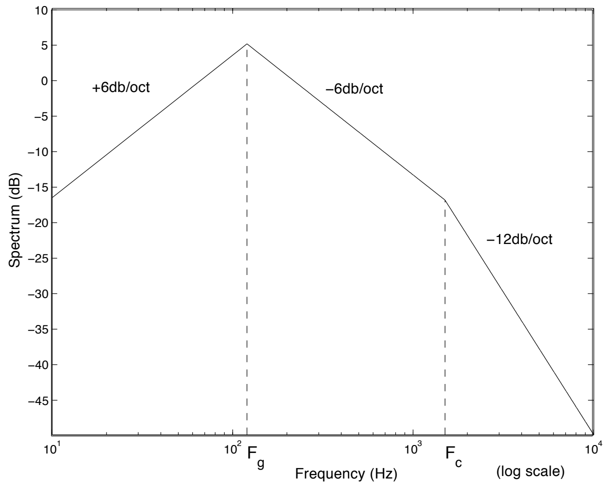


Figure 2: Linear stylization of the LF-Model spectrum.

2.2. Feature extraction

We measured the LF-parameters in ten utterances of the male speaker, which were selected from the speech corpus that was used to train the statistical models of the speech synthesizer. We calculated the mean values of the glottal parameters to generate the excitation signal in the speech synthesis.

Each utterance, sampled at 16 kHz, was analyzed pitch-synchronously using the epochs (instants of maximum excitation) calculated with the Entropic Signal Processing System (ESPS) tools. The algorithm to calculate the epochs is described in [13] and [14]. We obtained an estimate of the DGVV waveform by inverse filtering the speech signal. The resulting signal was high-pass filtered with a pre-emphasis filter ($\alpha = 0.97$) to eliminate the effect of the lip radiation. The LPC coefficients were calculated for each frame using a Hanning window, centered at the glottal epochs and with duration of 20 ms. Then, the residual was low-pass filtered at 4 kHz to reduce the high-frequency rumble effect on the energy envelope of the residual and permit a more accurate estimation of the glottal parameters.

The parameter t_e was estimated from the pitch-marks and E_e was the value of the waveform at that time instant. The other LF-parameters were estimated for each pitch cycle in the voiced regions.

The time instants t_c, t_o, and t_p can be extracted from the estimated glottal flow waveform. For example, t_p and t_o were calculated from the the electroglottographic (EGG) signal in [15]. We obtained an estimation of the glottal flow waveform by

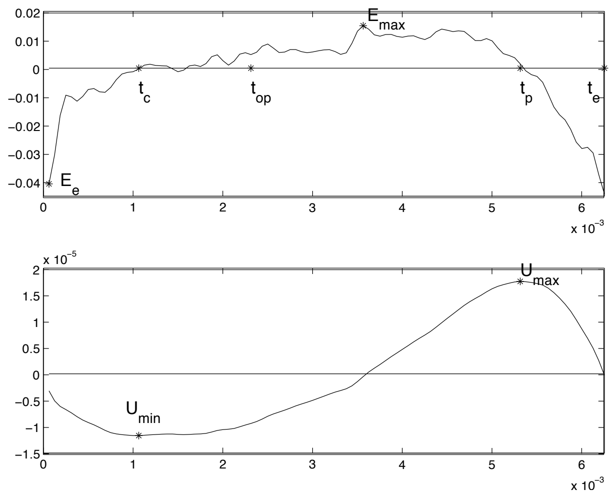


Figure 3: Estimation of t_c, t_o, and t_p. Top: a pitch cycle of the LPC residual; Bottom: integration of the residual signal (estimation of the glottal flow).

taking the integration of the LPC residual signal. The residual was first high-pass filtered by a linear phase FIR filter with cut-off frequency of 80 Hz to reduce the low frequency amplitude fluctuation that results from the integration operation. Figure 3 helps to explain the method to estimate these parameters. In the figure, the point of maximal flow amplitude U_{max} gives the instant t_p and the point of minimum flow amplitude U_{min} is the estimation of t_c. From [16], the point t_o can be approximated by:

$$t_o = \frac{2(U_{max} - U_{min})}{\pi E_{max}} \tag{7}$$

where E_{max} is the maximal value of the residual in the period. There are methods that measure these parameters using amplitude thresholds or zero crossings, e.g. [17], but they do not necessarily give precise results because they are sensitive to rumble noise and it is difficult to set the appropriate thresholds.

The estimation of t_a is typically more difficult. It is usually obtained by fitting a model to the inverse filtered signal, which requires the use of an optimization algorithm and more complex calculations. We use a simple and effective method which consists of calculating the derivative of each pitch cycle of the residual and then detecting the peak of maximal amplitude in the return phase (starts at t_e and has duration equal to t_a). This peak is represented by M in Figure 4. Figure 1 shows that the tangent to the exponential decaying curve in the LF-model has the maximum slope at the instant t_e. Thus, we calculate $t_a = E_e/(MF_s)$, where F_s is the sampling rate, by assuming that the amplitude of the peak, M, is equal to the slope of the tangent at $t = t_e$.

Figure 5 a) shows the contours of the measured parameters for two voiced regions of an utterance spoken by a male speaker. In general, the parameters appear to increase linearly with T, with exception of t_a which is approximately constant. In [7], the measurements of the parameters for 3 vowels spoken with different pitch show similar relations, except for deviations at high values of T. In that study, each vowel followed its own raising path and the parameters were influenced by the preceding phone. Our results also show variation of the glottal parameters trajectories between different phonetic segments of the utterances.

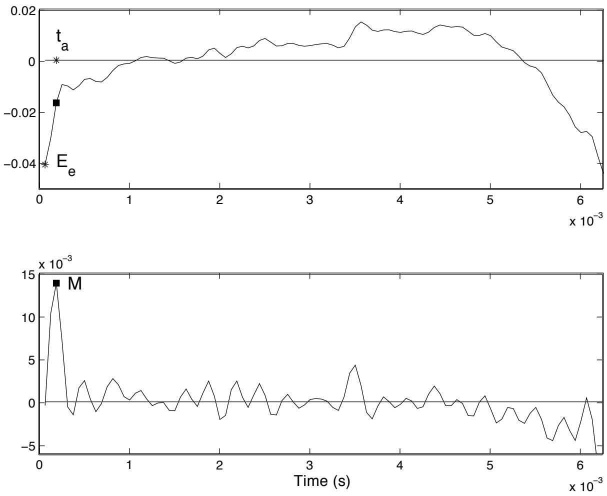


Figure 4: Estimation of t_a. Top: a pitch cycle of the LPC residual; Bottom: derivative of the residual signal.

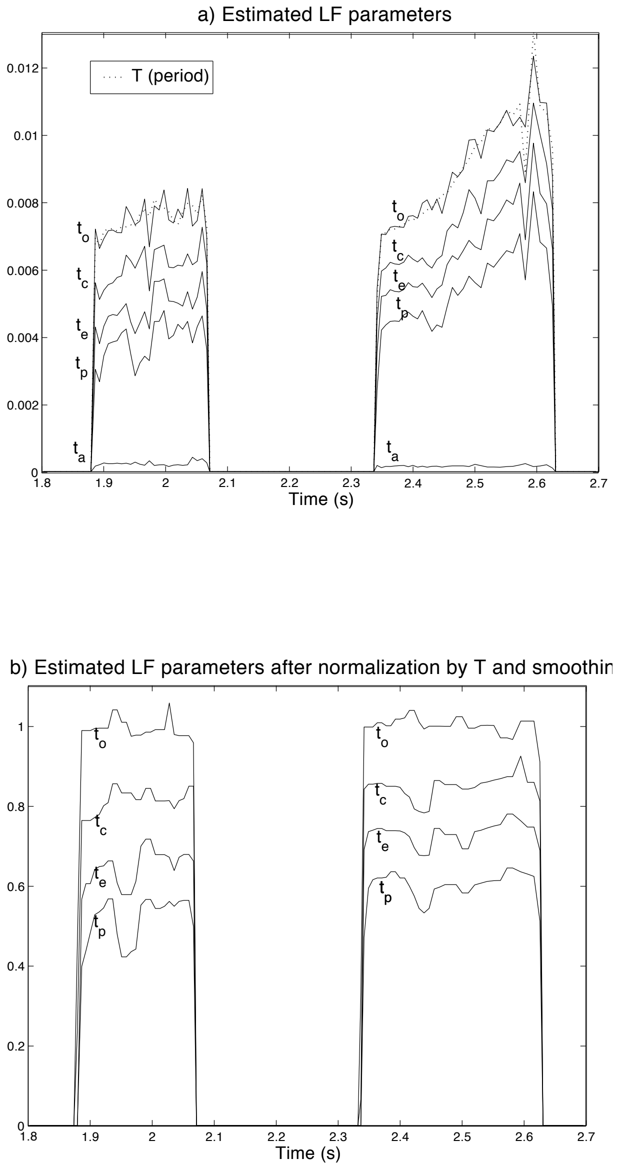
The values of the parameters, with the exception of t_a and E_e, are normalized by the pitch period. We use the median function to obtain smoother variations in the curves. Figure 5 b) shows the curves of the LF-parameters after the normalization and smoothing operations. Finally, the mean values of the glottal parameters are calculated.

3. System

3.1. General description

We integrated the LF-model into the speaker-dependent HMM-based speech synthesizer called Nitech-HTS 2005 [5]. This system uses the high-quality STRAIGHT method [6] to extract F_0, to compute the mel-cepstrum and to estimate spectral aperiodicity. The Nitech-HTS 2005 system uses a mixed multi-band excitation signal with phase manipulation, but it also has the option to use only the pulse train. The F_0 and aperiodicity parameters are used to generate the mixed-excitation signal. Speech is synthesized from the mixed-excitation and the mel-cepstral coefficients using an Mel Log Spectrum Approximation (MLSA) filter.

The system generates the excitation signals of voiced speech using a pulse (centered within a 512 sample length frame) which is processed to obtain phase randomization at the higher frequencies and summed with white Gaussian noise, in the spectral domain. The noise component is estimated on five frequency bands: 0-1, 1-2, 2-4, 4-6, and 6-8 kHz. For unvoiced speech, the glottal source component is modeled only by white Gaussian noise. The resulting short-time signals are multiplied by asymmetric windows and added using the Pitch-Synchronously Overlap-and-Add (PSOLA) algorithm [18]. The weighting windows are centered in the pulse and are composed of two half-hanning windows: the first half of a hanning window lasts the duration of the previous period and the second (decaying half) lasts the duration of the current period. In case of the unvoiced frames the durations of the windows are set to a constant.


a) Estimated LF parameters

b) Estimated LF parameters after normalization by T and smoothin

Figure 5: Curves of the estimated glottal parameters.

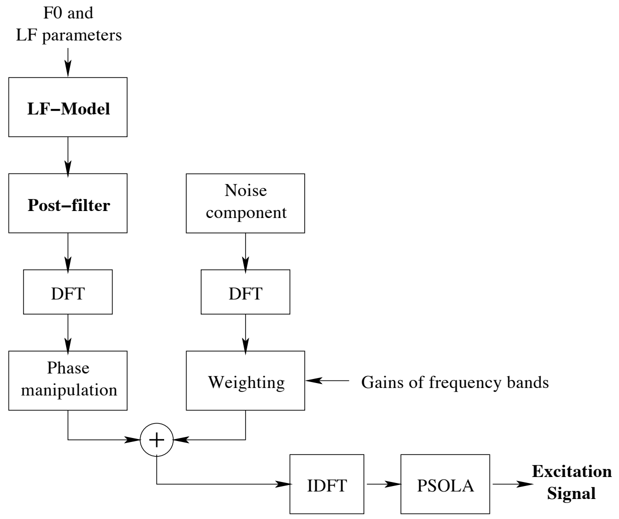


Figure 6: Block diagram of the excitation generation part using the LF-model.

3.2. Integration of the LF-model

We modified the synthesis part of the Nitech-HTS 2005 system to give the option to use the LF-model instead of the pulse train. By using the same approach based on STRAIGHT for the analysis and synthesis we can compare the effect on the speech quality of the two excitation source models. Figure 6 shows the schematic diagram of the excitation generation using the LF-model. When the system uses the new option, each voiced frame contains two pitch cycles of the LF-waveform, centered at the instant of maximum excitation, t_e. The LF-parameters, with the exception of t_a and E_e, are obtained by multiplying the normalized mean values of the glottal parameters by the synthesis period $T = 1/F_0$. In these calculations we assume that the variation of these parameters with T is approximately linear and constant. The parameters T_a and E_e are set to the constant mean value (not normalized) because they showed to have no correlation with T.

The STRAIGHT method estimates the spectrum envelope of the speech signal which is not a good estimation of the vocal tract because it only eliminates the F_0 effects of the glottal source from the speech. Thus, all the other aspects of the glottal source, such as the spectral tilt and the differences of amplitude of the first harmonics in the excitation spectrum, are described by the mel-cepstrum.

The pulse signal is used to model the periodicity of the excitation in the STRAIGHT speech synthesis method. The advantage of using this signal is that is spectrally flat. However, a drawback of using this model is that it has a strong harmonic structure at the higher frequencies when compared with the excitation of real speech, which has the effect of making the synthetic speech sound buzzy. Figure 7 shows the spectrum of a segment of the pulse train.

Glottal source models fit well in the source-filter theory which separates the speech signal in three independent processes: glottal excitation, vocal tract filter, and lip radiation. In this case, speech can be generated by feeding the glottal excitation through the vocal tract filter and performing a simple differentiation operation to model the lip radiation effect. However, source models are not appropriate for the synthesis with

STRAIGHT because this method uses a MLSA filter obtained from the mel-centrum instead of a vocal tract filter. We adapt the LF-model to the STRAIGHT synthesis method by using a post-filter that transforms the spectrum of the LF-signal into an approximately flat spectrum. This is equivalent to remove the spectral properties of the glottal spectral peak and the spectral tilt from the LF-waveform since they are described by the mel-spectrum. The post-filter is a linear phase FIR filter described by three linear segments which are symmetric to the slopes of the LF-model spectrum represented in Figure 2: -6dB/oct, +6dB/oct and +12dB/oct, respectively. We calculated the frequencies F_g and F_c from the equations 5 and 6, respectively, and using the mean values of the glottal parameters. Figure 8 shows the spectrum of a segment of the LF-model and the same segment after post-filtering.

If the HMM-based speech synthesizer was used to model the glottal parameters and generate them as it does with F_0, it would be necessary to use a time-varying post-filter which could be a limitation of approach.

The advantages of using the LF-model within this system are that it produces a less harmonic structure at the high-frequencies of the spectrum than the pulse train and permits flexibility to transform voice quality by modifying the glottal parameters.

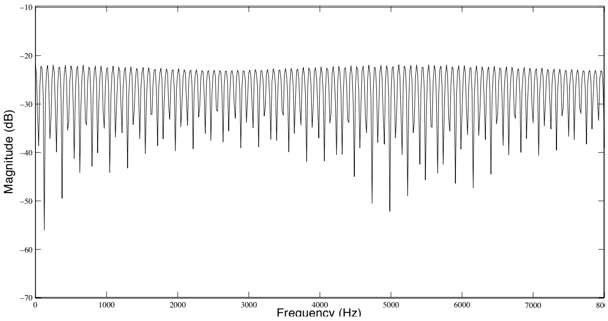


Figure 7: Spectrum of a segment of the pulse train (with duration 25 ms).

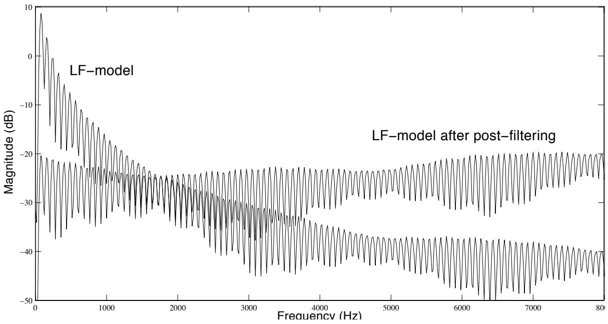


Figure 8: Spectrums of a segment (with duration 25 ms) of the LF-model signal and this signal after the post-filtering to obtain an approximately flat spectrum.

4. Perceptual evaluation

A forced-choice perceptual test was conducted to evaluate the performance of the LF-model when compared with the traditional pulse train used to model the excitation signal in the HMM-based synthesizer described in the previous section.

4.1. Stimuli

Speech was synthesized using the simple excitation (without multi-band noise or phase manipulation). The aperiodic components could also be used with the LF-model but they would have the same effect on the synthetic speech as when using the pulse train because the periodic and noise components are assumed to be independent. The US-English voice EM001 (male speaker) was built from the speech database released for the Blizzard Challenge 2007 (a total of approximately 8 hours of speech data). In the training part, the HMMs were modeled with the 39 order mel-cepstral coefficients obtained by the STRAIGHT analysis, the logF_0 and the aperiodicity measurements.

The mean values of the LF-parameters were calculated from the measures of the parameters obtained for eight speech utterances of the speech database of the speaker EM001.

The stimuli consisted of ten different utterances. For each utterance two speech signals were synthesized, using the LF-model and the pulse model for the excitation. The duration of the speech signals varied from 2.6 to 7.2 sec.

4.2. Experiment

The instructions presented to the subjects were simply to listen the two synthetic speech samples for each utterance and select the one that sounded most natural. At the end, they had to indicate if they used headphones or speakers, and if they were native speakers of English (U.K./U.S.) or not.

4.3. Listeners

Students and staff of Edinburgh University were asked to perform the test which was presented via a web interface browser. Eighteen listeners participated in the test, from which seven were native speakers of English.

4.4. Results

The results of the perceptual experiment are presented in Table 1. In general, subjects preferred the speech generated with the LF-model than with the pulse train. This result was expected because the source model presents a less harmonic structure at the higher frequencies when compared to the pulse signal, which reduces the buzzy effect of the synthetic speech. Although there is a clear improvement with the LF-model, the rate of 64% indicates that this model does not overcome completely the limitations of the pulse train. Thus, additional properties of the glottal source need to be modeled, such as the noise, to obtain more natural speech.

5. Conclusions and future work

The LF-model of the glottal source was implemented in a HMM-based speech synthesis system which originally used the pulse signal to model the excitation. The glottal source model increases the parametric flexibility of the system and permits to transform voice characteristics of the speech by modifying the glottal parameters.

A perceptual experiment was conducted to evaluate the per-

	Excitation	
	LF-Model	Pulse Train
Non-native speakers	61%	39%
Native speakers	68,6%	31,4%
Total scores and 95% CI	64% ± 6.7%	36% ± 6.7%

Table 1: Scores, in percentage, obtained by each excitation model in the evaluation of the naturalness of the synthetic speech.

formance of the LF-model when compared with the pulse train in the quality of the synthetic speech. The results indicate that the speech synthesized with the LF-model sounds more natural. Although the difference in speech quality in comparison with the pulse model is not large, the LF-model can be used with the multi-band mixed excitation to obtain further improvements.

In this work, the statistical parametric synthesizer used the STRAIGHT for the analysis and synthesis. This method uses the pulse train to model the periodicity of the voiced excitation. The LF-model is not compatible with STRAIGHT for the excitation generation because it models more characteristics of the source besides the period and presents a decaying spectrum in contrast to the spectrally flat spectrum of the pulse. Thus, a post-filter was used to adapt the glottal source model to the STRAIGHT spectrum. The LF-model was used with the mean values of the glottal parameters, which were estimated from recorded utterances of the speech database.

In the near future, we will implement the statistical parametric approach with the glottal source model and a good method to estimate the vocal tract filter. Another interesting topic for future work is to model the glottal parameters with the HMMs.

6. References

[1] Klatt, D.H. and Klatt, L.C., ”Analysis, Synthesis, and Perception of Voice Quality Variations among Female and Male Talkers”, J. Acoust. Soc. Amer., Vol. 87(2):820–857, 1990.

[2] Childers, D. G., ”Glottal Source Modelling for Voice Conversion”, Speech Communication, 7(6):697–708, 1995.

[3] Tokuda, K., Zen, H. and Black, A.W., “An HMM-based Speech Synthesis System Applied to English.”, Proc. of the 2002 IEEE SSW, pp.227230, USA, 2002.

[4] Black, A.W., Zen, H. and Toda, T., “Statistical Parametric Speech Synthesis”, Proc. of the IEEE ICASSP, pp.1229–1232, Hawaii, 2007.

[5] Zen, H., Toda, T., Nakamura, M. and Tokuda, K., ”Details of Nitech HMM-based Speech Synthesis System for the Blizzard Challenge 2005”, IEICE Trans. Inf. and Syst., Vol.E90-D, No.1, pp. 325–333, 2007.

[6] Kawahara, H., Masuda-Katsuse, I. and Cheveigné, A., ”Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f_0 extraction: Possible role of a repetitive structure in sounds”, Speech Communication, Vol. 27, pp. 187–207, 1999.

[7] Tooher, M. and McKenna, J.G., “Variation of the glottal LF parameters across F0, vowels, and phonetic environment”, Proc. of the ITRW (VOQUAL’03), Geneva, Switzerland, August 2003.

[8] Fant, G., ”The voice source in connected speech”, Speech Communication, Vol. 22, pp. 125–139, 1997.

[9] Fant, G. and Lin, Q., “Frequency domain interpretation and derivation of glottal flow parameters”, STL-QPSR, 29(2-3), pp. 1–21, 1988.

[10] Doval, B. and d’Alessandro, C., ”The spectrum of glottal flow models.” Notes et document LIMSI, num. 99–07, 1999.

[11] d’Alessandro, C. and Doval, B., “Voice quality modification for emotional speech synthesis”, Proc. of the Eurospeech 2003, pp. 1653-1656, Geneva, Switzerland, 2003.

[12] Doval, B. and d’Alessandro, C., “Spectral Correlates of Glottal Waveform Models: An Analytical Study”, Proc. of the Int. Conf. on Acoust., Speech and Signal Proc., Germany, Vol. 2, pp. 1295–1298, 1997.

[13] Talkin, D., ”Voicing epoch determination with dynamic programming”, J. Acoust. Soc. Amer., 85, Supplement 1, 1989.

[14] Talkin, D. and Rowley, J., “Pitch-Synchronous analysis and synthesis for TTS systems”, Proc. of the ESCA Workshop on Speech Synthesis, C. Benoit, Ed., Imprimerie des Ecureuils, Gieres, France, 1990.

[15] Krishnamurthy, A.K. and Childers, D.G., “Two-channel speech analysis”, IEEE Trans. Signal Process., Vol. 34, no. 4, pp. 730-743, 1986.

[16] Gobl, C. and Ní Chasaide, A., “Amplitude-based source parameters for measuring voice quality”, Proc. of the ITRW (VOQUAL’03), pp. 151–156, Geneva, Switzerland, August 2003.

[17] Arroabarren, I. and Carlosena, A., “Glottal source parameterization: a comparative study”, Proc. of the ITRW (VOQUAL’03), pp. 29–34, Geneva, Switzerland, August 2003.

[18] Moulines, E. and Charpentier, F., ”Pitch-synchronous waveform processing techniques for text to speech synthesis using diphones”, Speech Communications, Vol. 9, pp. 453–476, December 1990.

GMM-Based Speech Transformation Systems under Data Reduction

Larbi Mesbahi, Vincent Barreaud, Olivier Boeffard

IRISA / University of Rennes 1 - ENSSAT
6 rue de Kerampont, B.P. 80518, F-22305 Lannion Cedex
France
{lmesbahi,vincent.barreaud,olivier.boeffard}@irisa.fr

Abstract

The purpose of this paper is to study the behavior of voice conversion systems based on gaussian mixture model (GMM) when reducing the size of the training data corpus. Our first objective is to locate the threshold of degradation on the training corpus from which the error of conversion becomes too important. Secondly, we seek to observe the behavior of these conversion systems with regard to this threshold, in order to establish a relation between the size of training data corpus and the complexity of each method of transformation. We observed that the threshold is beyond 50 sentences (ARCTIC corpus), whatever the conversion system. For this corpus, the conversion error of the best approach increases only by 1.77 % compared to the complete training corpus which contains 210 utterances.

Index Terms: voice conversion , GMM, learning data reduction.

1. Introduction

During these last years, many applications in speech processing as text-to-speech synthesis or biometric identification by voice called upon speech transformation techniques. A voice conversion system tries to modify the vocal characteristics of a source speaker so that it is perceived as a target speaker. This technological issue is important: a man/machine interaction service can be more acceptable by offering various TTS voices [1][2].

Seminal approaches carried out a mapping-codebook conversion [3]. The main drawback of these approaches lies in the introduction of spectral discontinuities on the transformed signal. Several solutions were proposed in order to improve quality and precision, among them neuronal approaches [4], or segmental codebooks (STASC) [5]. Gaussian Mixture Model classifiers (GMM) make it possible to improve the mapping-codebook approaches, [6] [7]. Recently, the latter have been generalized using Hidden Markov Models, HMM, in order to treat the temporal dynamic aspect of the conversion function [8]. During our study, we were interested in the transformation techniques based on GMM acoustic segmentation, seen its many advantages such as the robustness, the continuity and the precision of the conversion function. However, these techniques have some drawbacks as over-fitting and the over-smoothing [1]. For many applications, in particular in the biometric field, it is necessary to carry out a voice conversion with very few data for the target speaker. The recording duration is usually short and the kind of voice differs from a speaker to another. In such a situation of scarce data, voice conversion methods must be adapted to ensure a good conversion quality. Our objective is to study the behavior of different conversion techniques based on GMM models [6] [7] [2] [1] under few learning data conditions. For this purpose, we gradually reduced the learning corpus and we evaluated the transformation quality on a reference test corpus. We carried out successive reductions, respectively of 75%, 50%, 25%, 10% and 5% on the initial learning set. Our objective is not to search for an optimization of the content of the training corpus under some imposed speech duration constraints, but to see whether the performance of the studied approaches remain stable when reducing the training corpus size. A complementary objective is to estimate the number of necessary sentences in order to maintain a good quality of conversion. Our main goal is to establish a compromise between the size of stored data and the conversion precision. This paper is organized as follows. In section 2 the studied GMM-based voice conversion approaches are presented. In the section 3 we treat and analyze the effect of data reduction on the quality of the conversion. Section 4 describes the experimental methodology and the obtained results. The conclusion will draw some prospects of this study.

2. GMM-based voice conversion

In the following, we consider two sequences of N q-dimensional acoustical vectors. The sequence corresponding to source speaker is represented by $X = [x_1, \ldots, x_N]^T$ and the target speaker by $Y = [y_1, \ldots, y_N]^T$. Given a GMM-based partitioning of the speakers' acoustic spaces, we need to estimate a piecewise function $\mathcal{F}(.)$ such that, $\forall n \in [1, \ldots, N]$, $\mathcal{F}(x_n)$ will be close to y_n. The GMM partitioning is commonly a joint source/target estimation realized after a Dynamic Time Warping (DTW) alignment. GMM are frequently used to model a speaker's acoustic space offering a continuous mapping of the acoustic vector space. With such a model, the probability for a vector to be in a class is given by the weighted sum of probabilities for this vector to belong to each gaussian component [6].

The probability distribution of x_n is modeled by a M-component GMM as in the following equation:

$$P(x_n) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(x_n, \mu_m, \Sigma_m)$$

with $\sum_{m=1}^{M} \alpha_m = 1, \quad \forall m \in [1, \ldots, M], \alpha_m \geq 0$, where $\mathcal{N}(., \mu_m, \Sigma_m)$ is the normal distribution of mean μ_m, and covariance matrix Σ_m. The α_m scalars represent prior probabilities of component m. The GMM parameters are estimated by EM (*Expectation-Maximisation*) on a learning set. The obtained GMM is a source model (see 2.1) or a joint model (see 2.2).

Once the GMM partitioning is done, the source/target conversion function can be derived as a weighted linear regression drawn from the conditional distribution of y_n with respect to

x_n (analogous to a bayesian regression). To present the studied techniques uniformly, this piecewise linear transform can be expressed with parameters A_m (matrix) and B_m (vector) as follows:

$$\mathcal{F}(x_n) = \sum_{m=1}^{M} P_m(x_n)[B_m + A_m(x_n - \mu_m)] \qquad (1)$$

with $P_m(x_n)$ the posterior probability of the m-th component given x_n.

In the following section, we present the solution proposed by Stylianou, [6]. Section 2.2 describes Kain's approach[7]. Finally, in sections 2.4 and 2.4, we present two approaches which takes into account the risk of *over-smoothing*[1] and *over-fitting*.

2.1. GMM on source only

The conversion method proposed in [9] uses a GMM source model. The conversion function that produces the linear regression is given by:

$$\mathcal{F}(x_n) \quad = \textstyle\sum_{m=1}^{M} P_m(x_n)[\nu_m + \Gamma_m \Sigma_m^{-1}(x_n - \mu_m)]$$

The ν_m and Γ_m parameters are estimated by a least squares minimization [6]. The covariance matrix Σ_m of the GMM model can be full or diagonal (called **stylianou-diag** in the following).

2.2. GMM on joint source-target

In this approach, [7] suggests to jointly model the target and the source by a GMM. Thus, $\forall n \in [1, \ldots, N]$ a joint vector is build, $z_n = [x_n y_n]$. We obtain the following density:

$$P(z_n) = P(x_n, y_n) = \quad \textstyle\sum_{m=1}^{M} \alpha_m \mathcal{N}(z_n, \mu_m, \Sigma_m)$$

$$\Sigma_m = \begin{bmatrix} \Sigma_{(m,XX)} & \Sigma_{(m,YX)} \\ \Sigma_{(m,XY)} & \Sigma_{(m,YY)} \end{bmatrix} \qquad \mu_m = \begin{bmatrix} \mu_{(m,X)} \\ \mu_{(m,Y)} \end{bmatrix}$$

The conversion function becomes:

$$\mathcal{F}(x_n) = \qquad (2)$$
$$\textstyle\sum_{m=1}^{M} P_m(x_n)[\mu_{(m,Y)} + \Sigma_{(m,YX)}\Sigma_{(m,XX)}^{-1}(x_n - \mu_{(m,X)})]$$

In the following, this technique will be referred to as **kain**.

2.3. Conversion and Over-smoothing risk

The major flaw of these GMM-based techniques is clearly presented in [1]. The spectral characteristics of the converted voices are excessively smoothed, referred to as *over-smoothing*; the consequence is an unclear speech signal. In [2], *Chen et al.* have demonstrated that 90% of the elements of the matrix product $\Sigma_{(m,YX)}\Sigma_{(m,XX)}^{-1}$ are ≤ 0.1 and 40% are ≤ 0.01, the correlation between the source and target speakers being weak. The effect of this statistical smoothing is to reduce the influence of the second term in equation 1, that is to say the term which contains the variability of X. Toda et al., [1], preserves the quality of a GMM-based conversion while simultaneously reducing the *over-smoothing* phenomenon. The solution is to impose a minimum level on the variance of the converted speech vectors. The maximum likelihood model (ML) is proposed to overcome the *over-smoothing* effect. The conversion function correspond to the following form:

$$\mathcal{F}(x) = (W^T D_m^{-1} W)^{-1} W^T D_m^{-1} E_m \qquad (3)$$

with

$$E_m = [E_1(m_{i1}), E_2(m_{i2}), \ldots, E_N(m_{iN})]$$

$$D_m^{-1} = diag[Dm_{i1}^{-1}, Dm_{i2}^{-1}, \ldots, Dm_{iN}^{-1}]$$

$$E_n(m_i) = \mu_{(i,Y)} + \Sigma_{(i,YX)}\Sigma_{(i,XX)}{}^{-1}(x_n - \mu_{(i,X)})$$

$$Dm_i = \Sigma_{(i,YY)} - \Sigma_{(i,YX)}\Sigma_{(i,XX)}{}^{-1}\Sigma_{(i,XY)}$$

n: takes the values from 1 to N, N: is the number of vectors, M: is the total number of GMM components, W: transformation matrix [10]. In the following, this solution will be called **toda**.

2.4. Conversion and Over-fitting risk

This phenomenon was already described by Stylianou in [6]: the problem is principally linked to the choice of a model that is too complex compared to the size of the learning set. *Over-fitting* is characterized by the fact that performances on the learning set increase and performances on a validation corpus decrease when the number of parameters of the model rises. The resulting model loses its generalization capability. In order to limit the *over-smoothing* issue while still obtaining a minimal distortion between the transformed and target vectors, we slackened the equality constraint on covariances introduced in [2] by directly binding these covariances to a diagonal matrix A_m (see equation 1). A diagonal matrix prohibits the cross-correlation between coordinates of the acoustic vectors. A_m is replaced by a global diagonal matrix, noted Γ. The coordinates of Γ, γ^j for $1 \leq j \leq q$, are estimated by a least squares estimation:

$$\gamma^j = \frac{\sum_{n=1}^{N}\left(y_n^j - \sum_{m=1}^{M} P_m(x_n)\mu_{(m,Y)}^j\right)\left(x_n^j - \sum_{m=1}^{M} P_m(x_n)\mu_{(m,X)}^j\right)}{\sum_{n=1}^{N}\left(x_n^j - \sum_{m=1}^{M} P_m(x_n)\mu_{(m,X)}^j\right)^2}$$

$B_m = \mu_{(m,Y)}$. We note this transformation by **gamma-vector**.

3. Data reduction effect

Within a general framework of automatic learning, the techniques of data reduction aim to reduce the computing time necessary to the transformation operations. The reduction is reached by selecting optimal databases, classically, either by techniques like K-means, or vectorial quantification [11]. Other approaches estimate a reduction in order to minimize the complexity of certain optimization problems [12].

In our study, we do not impose a specific acoustic criterion on the reduction mechanism. The adopted heuristics consist in reducing the initial database uniformly. Various progressive reduction are applied to the original training corpus in order to assess the influence on the quality of the transformation. Moreover, we try to establish a link between the size of the learning corpus and the parameters of the voice transformation model (number of training sentences, number of GMM components, dimension of the acoustic vector, etc.).

4. Experimental methodology

4.1. Experimental methodology

The comparative study is carried on an english database, noted *bdl-jmk*. This corpus corresponds to the speakers *bdl* and *jmk* of the ARCTIC speech database [13]. The methodology applied is as follows: 70% of the sentences in the corpus define the learning set. The remaining 30% define the test set. The sentences are chosen randomly. Based on this first learning and test partition, we defined various reduced learning corpora. The learning corpus corresponding to x% reduction of the full learning corpus will be noted as x%-(*bdl-jmk*). To summarize, the methodological conditions are as follows:

1. $\forall x, y \in \{100, 75, 50, 25, 10, 5\}$ such as $x < y$, we have $x\%$-(*bdl-jmk*) $\subset$ $y\%$-(*bdl-jmk*).
2. The *bdl-jmk* test corpus is the same for all reduction models and contain 90 utterances.

For each learning corpus, the number of speech utterances are as follows: 210 utterances for 100%-(*bdl-jmk*), 157 for 75%-(*bdl-jmk*), 105 for 50%-(*bdl-jmk*),52 for 25%-(*bdl-jmk*), 21 for 10%-(*bdl-jmk*) and 10 utterances for 5%-(*bdl-jmk*).

For each corpus, we respect the following methodology:

1. MFCC vectors computing (sampling frequency is 16 Khz, a 30 ms Hamming window is applied, the analyzing step is 10ms). The order of the MFCC vector is set to 13 (including energy) except for the **toda** transformation, where a vector is of dimension 26 (MFCC with their deltas).
2. Dynamic time warping between the *source* and *target* sequences using an euclidian norm on the MFCC vectors.
3. Parameter estimation of the GMM models (means, covariances and weights). We estimate joint source/target models. The source or target models are obtained by marginalizing a joint model. The learning process is carried out with a relative convergence threshold on the likelihood set to $1e^{-5}$. GMM models with 8, 32 and 64 components have been calculated. According to the studied conversion techniques, covariance matrices are full or diagonal.
4. Conversion of the source MFCC vectors by applying one of the conversion techniques described previously.

In this paper, we use a distortion score to measure the performance of the studied conversion functions with respect to various reduction ratios.

This distortion is defined as the mean distance between target and converted speech and normalized by the distance between source and target (Normalized Cepstral Distance). For this purpose, we used the following normalized cepstral distance for our objective tests:

$$e(\hat{c}^s, c^t) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{P}(\hat{c}_{ij}^s - c_{ij}^t)^2}{\sum_{i=1}^{N}\sum_{j=1}^{P}(c_{ij}^s - c_{ij}^t)^2}$$

such as:$\hat{c}^s$ is the transformed source vector, c^t is the target vector and c^s the source vector.

In order to consider reliable confidence intervals on these average scores, experiments are conducted 16 times (the complete process from the definition of a training and test sets). The scores are estimated both on test and learning corpora so as to appreciate the *over-fitting* effect. From the set of utterances issued in the ARCTIC corpus for two speakers *bdl* and *jmk*, 16 initial learning and test corpus were drawn randomly according to a 70/30 proportion. From each one of these 16 training corpora, 6 reduced corpora are drawn: from 100%-(*bdl-jmk*) to 5%-(*bdl-jmk*). Consequently, conversion techniques have to be tested on 96 corpora. For each one of these experiments, 3 models of acoustic space representation are calculated: GMM with 8, 32 and 64 components. Finally 6 transformation systems are tested, 2 of them required the training of parameters in addition to those of the GMM.

4.2. Results and discussion

The tables 1 and 2 present the Normalized Cepstral Distance between source and target for respectively the learning and test corpus. By column, the various reduction ratios applied to the learning corpora: 75%, 50%, 25%, 10% and 5%. By raw, various models of transformation for all GMM : 8, 32 and 64 components. An average score with a 95% confidence interval is estimated on 16 different experiments.

On reading these two tables, we can first note that each studied conversion system reacts in accordance with the following relations:

1. Learning[x%-(*bdl-jmk*)] $\leq$ Test[x%-(*bdl-jmk*)], $\forall x$.
2. Test[x%-(*bdl-jmk*)]$\leq$ Test[y%-(*bdl-jmk*)], $\forall x \leq y$.
3. Learning[x%-(*bdl-jmk*)] $\leq$ Learning[y%-(*bdl-jmk*)], ∀ x ≤ y.

Based on our discussion in section 3, we try to establish a relationship between the reduction ratio of the training corpus and parameters of transformation systems:

1. When the number of training sentences decreases, degradation increases. It will be seen that this degradation is not proportional to reduction ratio.
2. In certain extreme situations, it is not possible to compute a GMM model for lack of a sufficient number of data. For instance, the 64 components GMM cannot be computed on the 5%-(*bdl-jmk*) corpus. For the 10%-(*bdl-jmk*) corpus, the **stylianou-diag** with 64 components GMM cannot be carried out because the transformation matrices estimated by least square methods become singular.
3. The **toda** transformation system, which uses a source/target joint GMM of 52 dimensional acoustical vectors (MFCC coefficients source and target plus theirs derived), gives an error higher than any other method whatever the corpus reduction and the number of GMM component.

These observations lead to the following comments. The GMM parameters used by the studied transformations do not model efficiently the test data if the reduction of learning corpus is to important. Among those parameters, we note the probability distribution of a vector x_n for the m^{th} component of a GMM, noted as $P_m(x_n)$. This parameter influences largely the quality of the transformation. Moreover, the dimension of the acoustic vector should be counted as a parameter that influences the conversion quality. Indeed, in a similar framework, [14] shows that the classification error decreases as the dimension of observed vector increases (for a constant number of acoustic samples). What ever the learning method of the transformation is, a reduction of the learning set entails an increase of the distortion on the test set. One can search for a reduction threshold that keeps the distortion in an acceptable range. Yet, the studied conversion methods do not use the same number of parameters nor the same type to describe their transformations. Thus, they do not have the same behavior with regard to the reduction of the learning set. Consequently, a common reduction threshold, for all the conversion technique, is quite unimaginable. We rather propose to establish a range of reduction threshold that would establish a compromise between a light learning set and a high conversion precision. A parallel can be established between this last remark and [11] where ”safety regions” are established in machine learning.

		Reduced training corpora					
Transformation		100%-(*bdl-jmk*)	75%-(*bdl-jmk*)	50%-(*bdl-jmk*)	25%-(*bdl-jmk*)	10%-(*bdl-jmk*)	5%-(*bdl-jmk*)
kain Learning	GMM 8	0.391 ± 0.001	0.390 ± 0.001	0.389 ± 0.002	0.383 ± 0.003	0.371 ± 0.007	0.351 ± 0.009
	GMM 32	0.375 ± 0.001	0.373 ± 0.001	0.370 ± 0.002	0.360 ± 0.003	0.328 ± 0.007	0.311 ± 0.010
	GMM 64	0.365 ± 0.001	0.363 ± 0.001	0.358 ± 0.002	0.342 ± 0.002	0.318 ± 0.007	–
gamma-vector Learning	GMM 8	0.423 ± 0.003	0.423 ± 0.003	0.423 ± 0.003	0.419 ± 0.003	0.414 ± 0.007	0.408 ± 0.007
	GMM 32	0.398 ± 0.001	0.397 ± 0.001	0.396 ± 0.002	0.391 ± 0.003	0.376 ± 0.007	0.370 ± 0.008
	GMM 64	0.385 ± 0.001	0.384 ± 0.001	0.382 ± 0.002	0.374 ± 0.002	0.363 ± 0.007	–
stylianou-diag Learning	GMM 8	0.422 ± 0.003	0.422 ± 0.003	0.422 ± 0.003	0.419 ± 0.003	0.415 ± 0.007	0.413 ± 0.006
	GMM 32	0.396 ± 0.001	0.395 ± 0.001	0.395 ± 0.002	0.392 ± 0.003	0.384 ± 0.007	–
	GMM 64	0.386 ± 0.001	0.385 ± 0.001	0.385 ± 0.002	0.380 ± 0.003	–	–
toda Learning	GMM 8	0.497 ± 0.001	0.497 ± 0.002	0.497 ± 0.002	0.496 ± 0.004	0.490 ± 0.008	0.482 ± 0.006
	GMM 32	0.456 ± 0.001	0.455 ± 0.001	0.454 ± 0.002	0.449 ± 0.004	0.437 ± 0.009	0.418 ± 0.010
	GMM 64	0.445 ± 0.001	0.444 ± 0.002	0.443 ± 0.002	0.436 ± 0.003	0.414 ± 0.009	–

Table 1: This table presents the Normalized Cepstral Distance between source and target for learning corpus. By column, the various reduction ratios applied to the training corpora are 75%, 50%, 25%, 10% and 5%. By raw, various models of transformation for all GMM : 8, 32 and 64 components. An average score is estimated on 16 different experiments associate with a 95% confidence interval.

		Reduced training corpora					
Transformation		100%-(*bdl-jmk*)	75%-(*bdl-jmk*)	50%-(*bdl-jmk*)	25%-(*bdl-jmk*)	10%-(*bdl-jmk*)	5%-(*bdl-jmk*)
kain Test	GMM 8	0.395 ± 0.002	0.395 ± 0.002	0.397 ± 0.002	0.402 ± 0.002	0.420 ± 0.004	0.446 ± 0.009
	GMM 32	0.387 ± 0.002	0.390 ± 0.002	0.396 ± 0.002	0.416 ± 0.002	0.472 ± 0.006	0.548 ± 0.016
	GMM 64	0.389 ± 0.002	0.395 ± 0.002	0.407 ± 0.002	0.440 ± 0.003	0.534 ± 0.009	–
gamma-vector Test	GMM 8	0.423 ± 0.003	0.423 ± 0.003	0.424 ± 0.004	0.425 ± 0.003	0.432 ± 0.003	0.435 ± 0.005
	GMM 32	0.402 ± 0.002	0.404 ± 0.002	0.406 ± 0.002	0.414 ± 0.002	0.434 ± 0.003	0.460 ± 0.007
	GMM 64	0.395 ± 0.002	0.398 ± 0.002	0.404 ± 0.002	0.419 ± 0.002	0.459 ± 0.006	–
stylianou-diag Test	GMM 8	0.422 ± 0.003	0.422 ± 0.003	0.424 ± 0.004	0.424 ± 0.003	0.430 ± 0.003	0.437 ± 0.004
	GMM 32	0.398 ± 0.002	0.399 ± 0.002	0.401 ± 0.002	0.406 ± 0.002	0.420 ± 0.003	–
	GMM 64	0.391 ± 0.002	0.393 ± 0.002	0.396 ± 0.002	0.405 ± 0.002	–	–
toda Test	GMM 8	0.496 ± 0.002	0.496 ± 0.003	0.497 ± 0.003	0.501 ± 0.004	0.505 ± 0.005	0.508 ± 0.004
	GMM 32	0.458 ± 0.002	0.459 ± 0.002	0.461 ± 0.002	0.463 ± 0.003	0.467 ± 0.004	0.498 ± 0.008
	GMM 64	0.451 ± 0.002	0.452 ± 0.002	0.456 ± 0.002	0.465 ± 0.003	0.488 ± 0.005	–

Table 2: This table presents the Normalized Cepstral Distance between source and target on the test corpus. By column, the various reduction ratios applied to the training corpora are 75%, 50%, 25%, 10% and 5%. By raw, various models of transformation for all GMM : 8, 32 and 64 components. An average score is estimated on 16 different experiments associate with a 95% confidence interval.

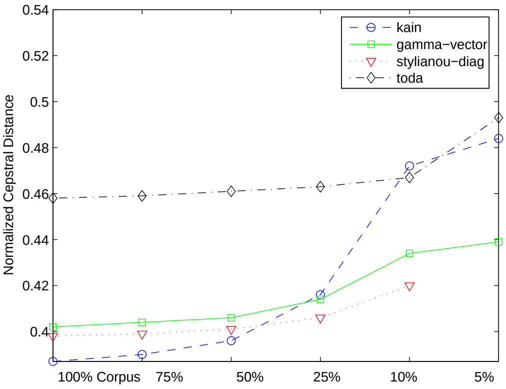


Figure 1: Evolution of normalized cepstral distance scores between transformed and target voices according to the data reduction with 75%, 50%, 25%, 10% and 5%, for the **kain**, **stylianou-diag**, **toda** et **gamma-vector** approaches. These results are obtained on the test corpus with 32 GMM components.

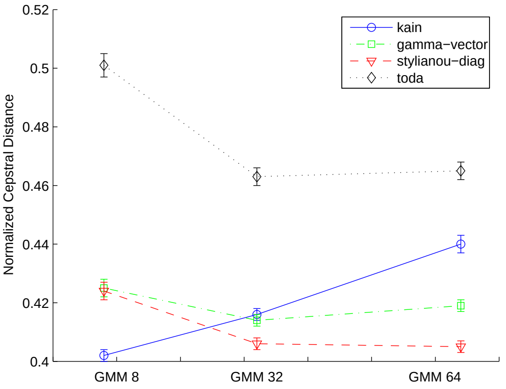


Figure 2: Evolution of normalized cepstral distance scores between transformed and target voices according to the number of GMM components for all approaches **kain**, **stylianou-diag**, **toda** and **gamma-vector**. We have fixed the threshold ratio at 25%. These results are obtained on the test corpus.

Figure 1 represents mean distortion scores between the transformed speaker and the target speaker, for a 32 component GMM. The scores are given for all the tested transformations for reduced learning sets. It can be observed that the performance of **kain** decreases for more than 25% reduction. Note that this transformation remains the more precise if it uses a less complex GMM (for instance a 8 component GMM: see table 2). **toda** is stable up to 10% reduction and crosses **kain** at this point. **stylianou-diag** is beter than **gamma-vector** up to 10%. **gamma-vector** is quite stable up to 5% since this technique uses far less parameters than the others.

We can observe, on this same figure, that the upper bound of the interval containing the optimal reduction thresholds for all the transformations is 25% (that is, a 52 sentences per corpus). Observe that the distortion of all the transformations lies in a stability zone that can go to above 25% reduction. For **gamma-vector**, the lower bound of this stability zone reaches 5% reduction (that is a 10 sentences per corpus). In that case, the conversion distortion is still acceptable (a 2.8% increase relative to the original learning set). Unfortunately, this technique suffers from over-fitting (for a fixed learning set, distortion rises as the number of components rises). For **stylianou-diag**, the threshold is before 10% reduction (21 sentences). It does not suffer from over-fitting. For **toda**, the threshold is before 5% reduction. The precision of this method is lower than any other transformation and is submitted to over-fitting. **kain** always give the best precision, its threshold is about 10%. To conclude, the general observation is all methods using less parameters are more stable.

Figure 2 shows the opposite situation of figure 1. Here, the distortion scores are presented for each transformation and for 25% reduction, since we consider this ratio as an optimal threshold. For each transformation, we used successively 8, 32 and 64 components GMM. We can notice that **kain** suffers from over-fitting even if its precision still overcomes **stylianou-diag** and **gamma-vector** for 8 components GMM. For 32 and 64 components GMM, **stylianou-diag** presents a better score than **gamma-vector**.

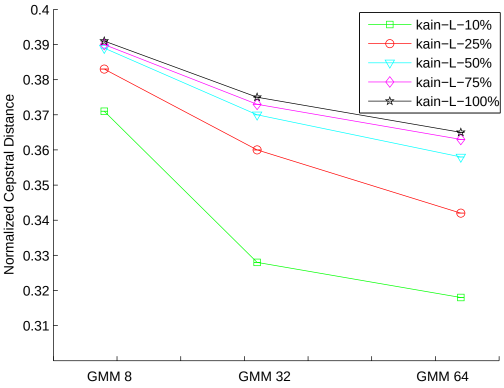


Figure 3: Evolution of normalized cepstral distance scores between transformed and target voices for **kain** approach, with the corresponding reductions 75%, 50%, 25% and 10%, according to the number of GMM components. These results are obtained on the learning corpus.

Figures 3 and 4 show the variations of the distortion scores

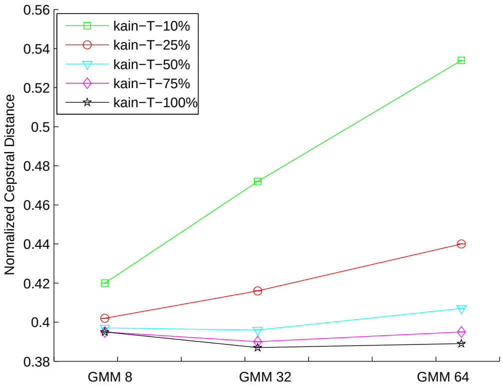


Figure 4: Evolution of normalized cepstral distance scores between transformed and target voices for **kain** approach, with the corresponding reductions 75%, 50%, 25% and 10%, according to the number of GMM components. These results are obtained on the test corpus.

of **kain** on the learning and test sets, for all reductions. It can be observed that, as the number of gaussian components rise, the score on the learning set improves while the score on the test set worsen. On the test set, the results can be divided in two classes. One of them contains the 100%, 75%, 50% and 25% reductions. For 10% reduction, scores can no longer be compared. This observation drove us to choose 25% as the reduction threshold. The obtained learning set regroups 52 sentences and is the best compromise between the corpus size and the conversion precision.

5. Conclusion

This work presents an experimental evaluation of various voice transformation techniques based on GMM models relative to the learning set's size. We observed that, in order to keep a good conversion score when this size is reduced, the number of parameter describing the transformations (number of gaussian component, the covariance type) must be reduced as well. For instance, for the transformation proposed by Kain with 32 components GMM, the normalized cepstral distance when using 5% of the original learning set, has a 41.6% variation relative to the score obtained with 100% of the learning set. This variation is only of 15.24% for a 8 components GMM. Furthermore, for the same transformation with 8 components GMM when using 25% of the original learning set, the variation of the distortion is only of 1.77% relative to the score obtained with 100% of the learning set. For a smaller learning set, the distortion rises unlinearly. We have observed that, on the Artic database, studied systems give fair conversion scores even if only 52 training sentences are available. Future work will evaluate the presented techniques when using various acoustic parameterizations. By varying the nature and dimension of the acoustic parameters, we seek to study the influence of reducing the learning set on conversion's precisions on an other level.

6. References

[1] T. Toda, A. W. Black, and K. Tokuda, "Spectral conversion based on maximum likelihood estimation considering global variance of converted parameter," in *International Conference on Acoustics, Speech, and Signal Processing*, 2005, pp. I9–I12.

[2] Y. Chen, M. Chu, E. Chang, and J. Liu, "Voice conversion with smoothed gmm and map adaptation," in *EUROSPEECH*, Geneva, Switzerland, September 2003, pp. 2413 – 2416.

[3] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, New York, April 1988, pp. 655–658.

[4] M. Narendranath, H. A. Murthy, S. Rajendran, and B. Yegnanarayana, "Transformation of formants for voice conversion using artificial neural networks," Elsevier Science B. V., pp. 207–216, 1995.

[5] L. M. Arslan, "Speaker transformation algorithm using segmental codebooks (stasc)," *Speech Communication*, vol. 28, pp. 211 – 226, 1999.

[6] Y. Stylianou, O. Capp, and E. Moulines, "Continuous probabilistic transform for voice conversion," in *IEEE Transactions on Speech and Audio Processing*, vol. 6, 1998, pp. 131–142.

[7] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1998, pp. 285–288.

[8] D. Helenca, B. Antonio, A. Kain, and J. Van Santen, "Including dynamic and phonetic information in voice conversion systems," in *Proceedings of the International Conference on Spoken Language Processing*, 2004, pp. 1193–1196.

[9] Y. Stylianou, O. Capp, and E. Moulines, "Statistical methods for voice quality transformation," in *EUROSPEECH*, Madrid, Espagne, 1995, pp. 447–450.

[10] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *ICASSP - International Conference on Acoustics, Speech, and Signal Processing*.

[11] K. Ravindra and H. Saman, "Reducing the number of training samples for fast support vector machine classification," *Neural Information Processing-Letters nd Reviews*, vol. 2, March 2004.

[12] M. Sebban, R. Nock, and S. Lallich, "Stopping criterion for boosting-based data reduction techniques: from binary to multiclass problems," *Journal of machine learning research*, vol. 3, pp. 863–885, 2002.

[13] J. Kominek and A. Black, "The cmu arctic speech databases for speech synthesis research," *Tech. Rep. CMU-LTI-03-177*, 2003.

[14] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEETrans. Information Theory*, vol. 14(1), pp. 55–63, 1968.

Improved Average-Voice-based Speech Synthesis Using Gender-Mixed Modeling and a Parameter Generation Algorithm Considering GV

Junichi Yamagishi[1], *Takao Kobayashi*[2], *Steve Renals*[1],
Simon King[1], *Heiga Zen*[3], *Tomoki Toda*[4], *Keiichi Tokuda*[3]

[1]University of Edinburgh, [2]Tokyo Institute of Technology,
[3]Nagoya Institute of Technology, [4]Nara Institute of Science and Technology,
jyamagis@inf.ed.ac.uk, takao.kobayashi@ip.titech.ac.jp, s.renals@ed.ac.uk,
simon.king@ed.ac.uk, zen@sp.nitech.ac.jp, tomoki@is.naist.jp, tokuda@nitech.ac.jp

Abstract

For constructing a speech synthesis system which can achieve diverse voices, we have been developing a speaker independent approach of HMM-based speech synthesis in which statistical average voice models are adapted to a target speaker using a small amount of speech data. In this paper, we incorporate a high-quality speech vocoding method STRAIGHT and a parameter generation algorithm with global variance into the system for improving quality of synthetic speech. Furthermore, we introduce a feature-space speaker adaptive training algorithm and a gender mixed modeling technique for conducting further normalization of the average voice model. We build an English text-to-speech system using these techniques and show the performance of the system.

1. Introduction

Recent concatenative speech synthesis approaches give us high quality synthetic speech. However, as is well known, these approaches always require large-scale speech corpora for generating natural sounding speech and as a consequence, become an inefficient choice and a major bottleneck when we need to quickly add new speakers' voices and construct a speech synthesizer which can simultaneously deal with many speakers' voices. To eliminate this bottleneck would lead to both cost reduction for building a new voices and many new applications for human-computer interfaces using speech input/output. In order to make such speech synthesis realistically feasible, we need to develop an approach in which synthetic speech comparable to that of a speaker-dependent system built using a large amount of speech data can be generated from a small amount of the speech data.

For this purpose, we have been developing speaker independent HMM-based speech synthesis in which “average voice models” are created using hidden semi-Markov models (HSMMs) and adapted with a small amount of speech data from the target speaker (e.g. [1, 2]). This speech synthesis method (Fig. 1) is referred to as “average-voice-based speech synthesis (AVSS).” By using this framework, we can obtain synthetic speech for a target speaker from even 100 utterances (about 6 minutes). Interestingly, we have shown that synthetic speech using this approach is perceived as being more natural sounding than that of the speaker-dependent (SD) system by many listeners because of the data-rich average voice model [3].

However, this system has similar drawbacks to the SD system: the synthetic speech has a “buzzy” quality, because the mel-cepstral vocoder with simple pulse or noise excitation of this system is identical to that of the speaker-dependent system. In order to alleviate the problem, Zen et al. [4] incorporated a high-quality speech vocoding method, STRAIGHT with mixed excitation [5], and a parameter generation algorithm considering global variance (GV) [6] into the speaker dependent HMM system and drastically improved the quality of synthetic speech. These improvements made a great contribution to the system in an open evaluation of corpus-based text-to-speech (TTS) synthesis system, named Blizzard Challenge 2005 [7].

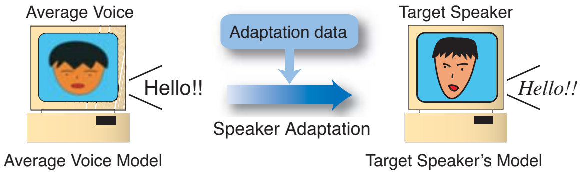


Figure 1: Average-voice-based speech synthesis.

It is important to remember that the amount of speech data available from the target speaker is very limited in the AVSS system. To add several new parameters required for a new technique results in increase of the number of parameters to be estimated from the small amount of speech data. Therefore it would be, strictly speaking, a trade-off problem to additionally use the mixed excitation system and the parameter generation algorithm considering GV in the AVSS system. However, fortunately, the number of additional parameters for the mixed excitation system is relatively small, and that for the parameter generation algorithm considering GV is small enough to directly estimate from the adaptation data.

Therefore, we have incorporated these promising techniques into the AVSS system to improve the quality of synthetic speech. We have investigated that these techniques are effective even under condition of limited amount of speech data, based on the results of subjective evaluations. In addition to these techniques, we propose a feature-space speaker adaptive training (SAT) technique using HSMM and a gender mixed modeling technique for conducting further speaker normalization of the average voice model. Although we utilized an HSMM-based model-space SAT algorithm in our conventional system, an HSMM-based feature-space SAT algorithm is alternatively used in order to efficiently utilize both mean vectors and covariance matrices of Gaussian probability density functions (pdfs) for the normalization of the average voice model. Then, in order to reflect gender information of training speakers as a prior information in the training and adaptation stages, we develop a gender mixed modeling technique. In these experiments, we apply the AVSS system using those techniques to U.S. English, build a new system named “AVSS 2006 ” and compare the system with our conventional system. We furthermore compare the system with the speaker dependent system “Nitech-HTS 2005,” which was the best system in the Blizzard Challenge 2005, in order to assess the performance of the AVSS system in the state-of-the-art TTS systems.

2. Details of the AVSS 2006 system

2.1. Speech Analysis using STRAIGHT

We use the STRAIGHT mel-cepstrum [4], log F_0, and aperiodicity measures as acoustic features in the same manner as the speaker dependent system Nitech-HTS 2005. The mel-cepstral coefficients are obtained by STRAIGHT spectral analysis [5] in which F_0-adaptive spectral smoothing is carried out in the time-frequency region. The F_0 values are estimated using the following three-stage extraction to reduce error of F_0 extraction such as halving and doubling and to suppress voiced/unvoiced error. First, using IFAS-based method [8], the system extracted F_0 values for all speech data of each speaker within a common search range. Then, the F_0 range of each speaker was roughly determined based on a histogram of the extracted F_0 values. F_0 values were re-extracted in the speaker-specific range using the IFAS algorithm, fixed-point analysis [9], and ESPS get-F_0 [10]. Finally, a median value of the extracted F_0 values at each frame was utilized as an eventual F_0 value. The aperiodicity measures for mixed excitation are based on a ratio between the lower and upper smoothed spectral envelopes, and averaged on five frequency sub-bands. In addition to these static features, dynamic and acceleration features of each static feature are used.

2.2. Acoustic Models and Labels

As in the case of our conventional Japanese AVSS system, we utilize context-dependent multi-stream left-to-right MSD-HMM/HSMMs [11] in order to simultaneously model the above acoustic features and duration. Details of the phonetic and linguistic contexts for U.S. English are identical to [12]. In addition to this phonetic and linguistic information, we added gender information of speakers into the context labels for conducting the gender-mixed modeling technique in the training procedures described in the next section.

2.3. Speaker Adaptive Training

Using the above HMM/HSMMs, we trained average voice models from training data consisting of several speakers’ speech. Training of the average voice model uses the SAT algorithm. Although we utilized a model-space SAT algorithms [13] using linear transformations of mean vectors of Gaussian pdfs in our conventional systems [1, 2], a feature-space SAT algorithm [14] is used as an alternative algorithm in the AVSS 2006 system to efficiently utilize both mean vectors and covariance matrices of the Gaussian pdfs for the speaker normalization of the average voice model. We can derive the feature-space SAT in the framework of HSMM in a similar way to [1]. Here we assume that each state of the HSMM has the following an output pdf $b_i(\boldsymbol{o})$ and a duration pdf $p_i(d)$:

$$b_i(\boldsymbol{o}) = \mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{1}$$

$$p_i(d) = \mathcal{N}(d; m_i, \sigma_i^2). \tag{2}$$

where $\boldsymbol{o}$ and d is an observation vector and a duration at state i, respectively. The feature-space SAT of the HSMM estimates the parameters of the Gaussian pdfs as follows:

$$\overline{\boldsymbol{\mu}}_i = \frac{\sum_{f=1}^{F} \sum_{t=1}^{T_f} \sum_{d=1}^{t} \gamma_t^d(i) \sum_{s=t-d+1}^{t} \overline{\boldsymbol{o}}_s^{(f)}}{\sum_{f=1}^{F} \sum_{t=1}^{T_f} \sum_{d=1}^{t} d \cdot \gamma_t^d(i)} \tag{3}$$

$$\overline{\boldsymbol{\Sigma}}_i = \sum_{f=1}^{F} \sum_{t=1}^{T_f} \sum_{d=1}^{t} \gamma_t^d(i) \frac{\sum_{s=t-d+1}^{t} (\overline{\boldsymbol{o}}_s^{(f)} - \overline{\boldsymbol{\mu}}_i)(\overline{\boldsymbol{o}}_s^{(f)} - \overline{\boldsymbol{\mu}}_i)^\top}{\sum_{f=1}^{F} \sum_{t=1}^{T_f} \sum_{d=1}^{t} d \cdot \gamma_t^d(i)} \tag{4}$$

$$\overline{m}_i = \frac{\sum_{f=1}^{F} \sum_{t=1}^{T_f} \sum_{d=1}^{t} \gamma_t^d(i) \cdot \overline{d}^{(f)}}{\sum_{f=1}^{F} \sum_{t=1}^{T_f} \sum_{d=1}^{t} \gamma_t^d(i)} \tag{5}$$

$$\overline{\sigma}_i^2 = \frac{\sum_{f=1}^{F} \sum_{t=1}^{T_f} \sum_{d=1}^{t} \gamma_t^d(i) \cdot (\overline{d}^{(f)} - \overline{m}_i)^2}{\sum_{f=1}^{F} \sum_{t=1}^{T_f} \sum_{d=1}^{t} \gamma_t^d(i)} \tag{6}$$

where F is number of the training speakers, T_f is total number of frames of a speaker f, and $\gamma_t^d(i)$ is the state occupancy probability at state i of the HSMM. Note that $\overline{\boldsymbol{o}}_s = \boldsymbol{\zeta} \boldsymbol{o}_s + \boldsymbol{\epsilon}$ and $\overline{d} = \chi d + \nu$ are linearly transformed observation vector and duration, respectively. These transformation matrices ($\boldsymbol{W} = [\boldsymbol{\zeta}, \boldsymbol{\epsilon}]$ and $\boldsymbol{X} = [\chi, \nu]$) are simultaneously estimated using the HSMM-based CMLLR algorithm [15]. This technique can be viewed as a generalized version of several normalization techniques such as CMN, CVN, VTLN, and bias removal of F_0 and duration. Since this HSMM-based feature-space SAT algorithm requires a lot of computation, we basically train the acoustic models using the HMM-based feature-space SAT algorithm and apply the HSMM-based SAT algorithm in the final embedded training procedures (see Fig. 2).

Another advantage of this feature-space SAT is feasibility. As reported in [14], in the the model-space SAT algorithms, it is necessary to store a full matrix for each Gaussian pdf, or store statistics for each Gaussian component for every speaker. In our *speaker-independent* HMM-based speech synthesis system, the number of the Gaussian pdfs reaches $\mathcal{O}(10^7)$ or more, and it partly makes the parameter estimation impractical. In particular, the embedded training procedures in which we could use the model-space SAT were restricted to the training procedures in which the parameters of the Gaussian pdfs were tied among several pdfs. On the other hand, we can apply the feature-space SAT algorithm to all the embedded training procedures and conduct further normalization in the training of the average voice model.

2.4. Gender-Mixed Modeling

In general, speech data weaves speaker-dependent characteristics with gender-dependent characteristics in addition to phonetic and prosodic features. We must reproduce both the gender-dependent characteristics as well as the speaker-dependent characteristics of the target speaker in our system. If large amounts of training data for both genders are available, it would be the most efficient choice to use gender-dependent average voice models using enough training data as an initial model of the speaker adaptation. However, in practice, we encounter common problems from the amount of the training data available from either gender or both genders being limited. In such cases, it would not be the best choice to use gender-dependent average voice models. In addition to this, it is not straightforward to clarify that how many training sentences and speakers are enough for constructing the appropriate gender-dependent average voice models in any condition.

Another practical approach is to use a gender-independent average voice model (or the opposite gender-dependent model

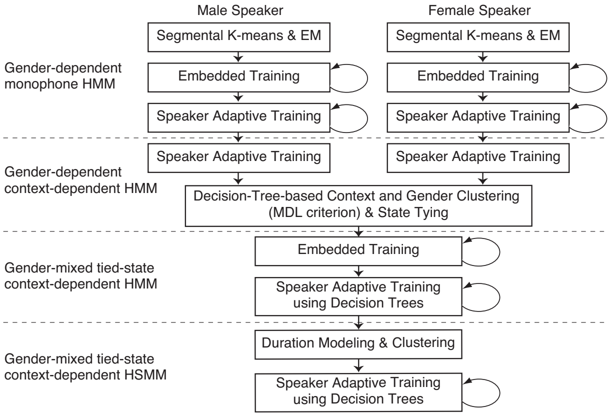

Male Speaker
Female Speaker
Segmental K-means & EM
Segmental K-means & EM
Gender-dependent monophone HMM
Embedded Training
Embedded Training
Speaker Adaptive Training
Speaker Adaptive Training
Gender-dependent context-dependent HMM
Speaker Adaptive Training
Speaker Adaptive Training
Decision-Tree-based Context and Gender Clustering (MDL criterion) & State Tying
Gender-mixed tied-state context-dependent HMM
Embedded Training
Speaker Adaptive Training using Decision Trees
Gender-mixed tied-state context-dependent HSMM
Duration Modeling & Clustering
Speaker Adaptive Training using Decision Trees

Figure 2: Details of gender-mixed modeling. This modeling technique consists of the speaker adaptive training and the decision-tree-based context and gender clustering.

using enough training data) as an initial model, instead of the correct gender-dependent average voice model. However, we have shown that naturalness and similarity of the synthetic speech using those average voice models becomes significantly worse than that of the synthetic speech using the correct gender-dependent average voice model [16]. This is a logical conclusion because we have to adapt not only speaker-dependent characteristics but also gender-dependent characteristics of the average voice model based on a small amount of the adaptation data. An alternative approach is to simultaneously use the gender-dependent average voice models to complement one another and to perform soft decisions in the speaker adaptation [16]. However, there was no significant improvements between the results of the simultaneous use of the gender-dependent average voice models and those of the single gender-dependent average voice model. Although the simultaneous use of the gender-dependent average voice models could complement one another, it required twice as many parameters for the adaptation as the gender-dependent average voice model, and it seemed to suffer from “curse of dimensionality.” In summary, we are required to develop an approach which satisfies the following three conditions: 1) it reflects the gender-dependent characteristics as a prior information, 2) it makes the best possible use of the training data from both genders and complements one other if necessary, and 3) it does not increase the number of parameters required for the speaker adaptation.

To achieve this, we propose a *gender-mixed modeling* technique. The key idea of this gender-mixed modeling is similar to *style-mixed modeling* proposed in [17]. The gender-mixed modeling technically includes the speaker adaptive training and a decision-tree-based context and gender clustering technique. The actual training procedures for the modeling were conducted as follows (see Fig. 2). In order to conduct both normalization of the speaker-dependent characteristics and conservation of the gender-dependent characteristics, we first train gender-dependent monophone HMMs using the SAT algorithm. Then we convert them into gender-dependent context-dependent HMMs, and re-estimate the model parameters using the SAT algorithm again. Then, using the state occupancy probabilities obtained in the SAT framework, the decision-tree-based context clustering technique using minimum description length (MDL) criterion is applied to the HMMs, and the model parameters of the HMMs at each leaf node of the decision trees are tied. In the clustering, gender information of each speaker is treated as one of contexts for the clustering, and the clustering technique is applied to both the gender-dependent models at the same time. As a result, the gender information is included in a single acoustic model. Note that the decision trees were separately constructed for each state of mel-cepstrum, $\log F_0$, aperiodicity measures, and duration parts. Hence, when the target feature is generally gender-specific, such as $\log F_0$, the gender would be automatically split at around a root node of the tree by using gender-related questions, and the pdfs of the feature can keep the gender-dependent characteristics if required. Then, when dependency on gender of the target feature locally occurs such as duration, the gender information are automatically split as well as other contexts during the construction of a decision tree, and thereby we can make use of the training data from both genders laconically. We refer to the resulting model as a gender-mixed average voice model. Figure 3 shows a part of the constructed decision tree for the mel-cepstral part in the fifth state of the HMMs.

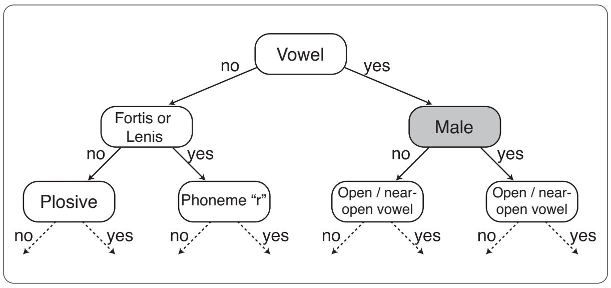


Figure 3: Part of a constructed decision tree in the gender-mixed modeling. Genders of training speakers are split by using gender-related questions as well as other contexts.

We re-estimate the clustered HMMs using SAT algorithm with piecewise linear regression functions. To determine regression classes for the piecewise linear regression, the decision trees constructed for the gender-mixed model are used, since use of the decision tree automatically reflects both differences of gender information and phonetic and linguistic information, and it is expected that more appropriate normalization for the average voice model is conducted. We then calculate initial duration pdfs from trellises of the HMMs [18], and conduct the decision-tree-based context and gender clustering for the duration pdfs. Using the tied duration pdfs, we perform the HSMM-based SAT algorithm with piecewise linear regression functions in order to normalize speaker characteristics included in the duration pdfs as well as other acoustic features. In each iteration of these SAT stages, we first estimated transformation matrices three times, and then updated mean vectors of both output and duration pdfs, their covariance matrices, weight for MSD, and transition matrices five times. Then we repeated the iterations three times in each SAT stage.

In the speaker adaptation stage, we adapt the gender-mixed average voice model to that of the target speaker by using a small amount of speech data with gender information of the target speaker. We utilize a combined algorithm of HSMM-based constrained structural maximum a posteriori linear regression (CSMAPLR) [19] and maximum a posteriori (MAP) adaptation [3]. In the CSMAPLR adaptation, the decision trees for the gender-mixed average voice model are used for the same reason as the above SAT algorithm with piecewise linear regression functions.

2.5. Parameter Generation Considering Global Variance

In the synthesis stage, input text is first transformed into a sequence of context-dependent phoneme labels with the gender information of the target speaker. Based on the label sequence, a sentence HSMM is constructed by concatenating context-dependent HSMMs. From the sentence HSMM, mel-cepstrum, $\log F_0$, and aperiodicity-measure sequences are obtained using the parameter generation algorithm considering GV [6], in which phoneme durations are determined using the duration pdfs. The parameter generation algorithm is a penalized maximum likelihood method in which the GV pdf (a Gaussian pdf for the variance of the trajectory at utterance level) acts as a penalty for the likelihood function. The algorithm tries to keep the global variance of the generated trajectory as wide as that of the target speaker, while maintaining an appropriate parameter sequence in the sense of maximum likelihood. It is possible to adapt the GV pdf from a speaker-independent model to that of a target speaker using MAP adaptation. However, the number of parameters of a GV pdf is very small. Specifically, it is equal to the dimensionality of the static features. Hence we directly estimate the GV pdf from the adaptation data. The generation method for speech waveforms is identical to that of Nitech-HTS 2005. A one-pitch waveform is synthesized from STRAIGHT mel-cepstral coefficients and the mixed excitation with the MLSA filter, and then a synthesized waveform was generated with PSOLA.

3. Experiments

3.1. Experimental conditions

We carried out several subjective and objective evaluation tests to assess the performance of the AVSS 2006 system. We used the CMU-ARCTIC speech database, which contains a set of about a thousand phonetically balanced sentences uttered by 4 male speakers (AWB, BDL, JMK, RMS) and 2 female speakers (CLB, SLT), and a speech database, which was released from ATR for the purpose of the Blizzard Challenge 2007 and contains the same sentences as that of CMU-ARCTIC speech database and additional sentences uttered by a male speaker EM001. To model the synthesis units, we used the "radio" phone set of the Festival speech synthesis system, and took the phonetic and linguistic contexts included in the utterance files of the Festival speech synthesis system into account.

Speech signals were sampled at a rate of 16 kHz and windowed by an F_0-adaptive Gaussian window with a 5-ms shift. The feature vectors consisted of 25 STRAIGHT mel-cepstral coefficients (including the zeroth coefficient), $\log F_0$, aperiodicity measures, and their dynamic and acceleration coefficients. We used 5-state left-to-right context-dependent multi-stream MSD-HSMMs without skip paths. Each state had a single Gaussian pdf with a diagonal covariance matrix. In the speaker adaptation, the transformation matrices were triblock diagonal corresponding to the static, dynamic, and acceleration coefficients.

3.2. Evaluation of the AVSS 2006 system

First, we evaluated naturalness and similarity of the synthetic speech generated from the adapted model. We chose a male speaker AWB as a target speaker of the speaker adaptation and used 3 male speakers (BDL, JMK, RMS) and 2 female speakers (CLB, SLT) of CMU-ARCTIC database as training speakers for the average voice model. The number of training data from each speaker was about 1000 sentences and the number of the adaptation sentences from the target speaker was 100 sentences selected from the corpus randomly. Then, ten test sentences which were not included in either the training or the adaptation data were used for the subjective evaluations. We constructed our conventional system (AVSS 2005 system) [2] and the AVSS 2006 system using the above training data and adapted the resulting average voice models of each system to the target speaker using the above adaptation data. Note that the shared-decision-tree-based context clustering algorithm was not used in both systems, since the algorithm is a directly-opposed idea from that of gender mixed modeling.

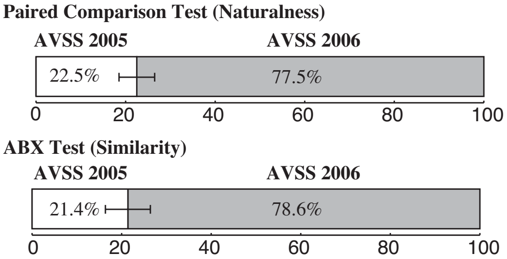


Figure 4: The average preference scores of the paired comparison test and the ABX test using our conventional system (AVSS 2005 system) and the proposed system (the AVSS 2006 system).

We then conducted a paired comparison test to investigate that these techniques are effective even under condition of limited amount of speech data. We compared the synthesized speech generated from the adapted models using the AVSS 2005 or 2006 systems. The subjective evaluations were conducted via the Internet. 28 subjects were presented a pair of synthetic speech utterances generated from the adapted models in random order, and asked which speech sounded more natural. At the same moment, we conducted an ABX comparison test to assess adaptation performance of the average voice models of both systems. In the ABX test, the subjects were presented a reference speech in addition to the above pair of synthesized speech, and asked to select the first or second synthetic speech as being similar to the reference speech. The reference speech was the recorded original speech. The same test sentences as the paired comparison test were used.

Figure 4 shows the average preference scores with 95% confidence interval of the paired comparison test and the ABX test. From this figure, we can see that naturalness and similarity of the synthetic speech generated from the adapted model using the AVSS 2006 system are drastically improved compared to our conventional system. In order to analyze which technique brings this good result, we separately investigated effects of STRAIGHT, feature-space SAT, gender mixed modeling, and parameter generation algorithm considering GV using preliminary evaluations. From the preliminary evaluations, we confirmed that each method had some effect, and above all the parameter generation algorithm considering GV made a huge contribution to the improvements in these subjective evaluations. However, it is interesting to note that objective measures such as mel-cepstral distance or RMSE of $\log F_0$ between synthetic speech using GV and real speech became worse than those between synthetic speech without GV and real speech. Since the experimental results for the STRAIGHT and the parameter generation algorithm considering GV were similar to the results of speaker-dependent system [4], we report the effect of the feature-space SAT and gender mixed modeling in the next subsections.

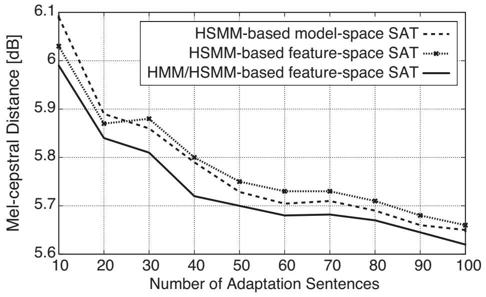


Figure 5: Objective evaluation of the SAT algorithm: Average mel-cepstral distance.

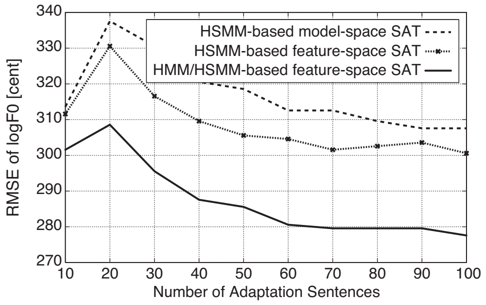


Figure 6: Objective evaluation of the SAT algorithm: RMSE of $\log F_0$.

3.3. Evaluation of the Feature-Space SAT

We evaluated the feature-space SAT algorithm using two types of objective evaluations based on the average mel-cepstral distance and RMSE of $\log F_0$. In these evaluations, we chose a male speaker EM001 as a target speaker of the speaker adaptation and used 4 male speakers (AWB, BDL, JMK, RMS) and 2 female speakers (CLB, SLT) of CMU-ARCTIC database as training speakers for the average voice model. We constructed three kinds of the gender-independent average voice model using HSMM-based model-space SAT and HMM/HSMM-based feature-space SAT, and adapted the resulting average voice models of each system to the target speaker. The amount of training data from each speaker was about 1100 sentences. The adaptation data was from 10 sentences to 100 sentences. 1000 test sentences were used for the evaluations, and these were included in neither the training nor the adaptation data. For the calculation of the average mel-cepstral distance and the RMSE of $\log F_0$, the state duration of each HSMM was adjusted after Viterbi alignment with the target speakers' real utterance.

Figure 5 shows the average mel-cepstral distance between spectra generated from the adapted model and spectra obtained by analyzing target speakers' real utterance. Figure 6 shows the RMSE of $\log F_0$ between F_0 patterns of synthetic and real speech. Silence, pause, and consonant regions were eliminated from the mel-cepstral distance calculation. Since F_0 is not observed in the unvoiced region, the RMSE of $\log F_0$ was calculated in the region where both the generated and the real F_0 were voiced. Comparing HSMM-based model-space and feature-space SAT only, one sees that the feature-space SAT gives slightly better results in the adaptation of the F_0 parameter, whereas the error of the feature-space SAT partly becomes slightly worse in the adaptation of the spectral parameters. However, we can also see that when we consistently apply the feature-space SAT to all the embedded training procedures for HMMs and HSMMs, both the mel-cepstral distance and RMSE of $\log F_0$ significantly decrease.

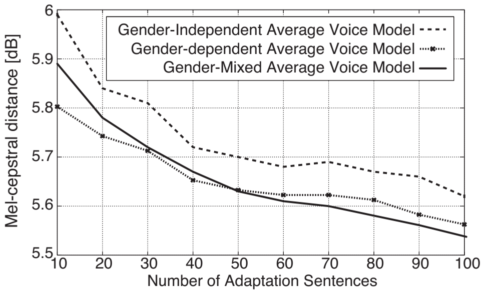


Figure 7: Objective evaluation of the gender-mixed modeling: Average mel-cepstral distance.

3.4. Evaluation of the Gender-Mixed Modeling

Then, we evaluated the gender-mixed modeling using the mel-cepstral distance. We constructed the gender-independent, gender-dependent, and gender-mixed average voice models and adapted these average voice models to the target speaker using the same adaptation data. The experimental condition on the speech data in this subsection is the same as 3.3.

Figure 7 shows the average mel-cepstral distance between spectra generated from the adapted model and spectra obtained by analyzing target speakers' real utterance. Silence, pause, and consonant regions were eliminated from the mel-cepstral distance calculation. Comparing the gender-dependent and gender-mixed average voice models, we can see that from 10 to 50 adaptation sentences, the gender-dependent modeling is generally better, whereas the gender-mixed modeling becomes better from the 50 to 100 adaptation sentences. We believe that this is because the gender-mixed average voice model has many more pdfs than the gender-dependent model, although we need to perform further experiments to investigate it.

3.5. Comparison with Nitech-HTS 2005

Finally, we conducted a comparison category rating (CCR) test and assessed the performance of the AVSS system with the state-of-the-art TTS systems. For this purpose, we compared the synthesized speech generated from the AVSS 2006 system with that of the speaker-dependent system Nitech-HTS 2005. The only difference between this Nitech-HTS 2005 system and a system reported in [4] is dimension of mel-cepstral coefficients. In [4], 39 mel-cepstral coefficients were used. However, this increases the number of parameters of the matrix for linear transformation. Hence we consistently utilize 24 mel-cepstral coefficients for both systems. The experimental condition on the training data in this subsection is the same as 3.3. We constructed the AVSS 2006 system using the training data and adapted the resulting average voice model to the target speaker using 100 sentences of the target speaker EM001. The speaker-dependent system Nitech-HTS 2005 was built using 1000 sentences of the target speaker EM001. For reference, we compared synthesized speech generated from an adapted model using the same 1000 sentences of the target speaker EM001 as adaptation data. 25 subjects were first presented with synthetic speech of Nitech-HTS 2005 as a reference speech and then with synthesized speech from the adapted models using 100 sentences or 1000 sentences in random order. Then the subjects were asked to comprehensively evaluate the synthetic speech generated from the adapted models compared with the reference speech. The evaluation was done on a 5-point scale, that is, 2 for better, 1 for slightly better, 0 for almost the same, -1 for slightly worse, and 2 for worse than the reference speech.

The average values and their 95% confidence interval of each adapted model in the CCR tests were **0.140±0.145** for 100 sentences and **0.424±0.08** for 1000 sentences, respectively. The values indicate that the AVSS 2006 system can synthesize speech of almost the same quality as the Nitech-HTS 2005 system from just 100 sentences, that is, 10% of the training data for the speaker-dependent systems. This is a very meaningful result since the Nitech-HTS 2005 system was evaluated as a best system in the Blizzard Challenge 2005, and we can say that the synthetic speech using the AVSS 2006 system bears comparison with other state-of-the-art TTS systems. Furthermore, we can see that the synthetic speech generated from the AVSS 2006 system using 1000 sentences is judged to be slightly better than those using 100 sentences and Nitech-HTS 2005 system. This result implies that this average voice approach is no longer just a speaker conversion system and it has the potential to surpass the common speaker-dependent approach.

4. Conclusions

In this paper, we incorporated a high-quality speech vocoding method STRAIGHT and a parameter generation algorithm with GV into the AVSS system for improving quality of synthetic speech. In addition to these techniques, we also proposed a feature-space SAT algorithm using the HSMM and a gender mixed modeling technique for conducting further speaker normalization of the average voice model. We applied the AVSS system using these techniques to U.S. English and built a new system named AVSS 2006 system. From the subjective evaluations, we shown that naturalness and similarity of the synthetic speech of the AVSS 2006 system were drastically improved compared to our conventional system, and then the AVSS 2006 can synthesize speech of the almost the same quality as the Nitech-HTS 2005 system from just 100 sentences.

Our future work is to develop a modeling technique for dealing with several dialects of English in the framework of the average voice model. We will also focus on developing an unsupervised speaker adaptation algorithm for speech synthesis.

5. Acknowledgments

This research was conducted for the purpose of the Blizzard Challenge 2006 and 2007. The authors would like to thank Dr. Yasser Hifny Abdel-Haleem of IBM T.J. Watson research center for his original idea on gender-mixed modeling.

6. References

[1] J. Yamagishi and T. Kobayashi, "Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 2, pp. 533–543, Feb. 2007.

[2] J. Yamagishi, T. Kobayashi, M. Tachibana, K. Ogata, and Y. Nakano, "Model adaptation approach to speech synthesis with diverse voices and styles," in *Proc. ICASSP 2007*, Apr. 2007, pp. 1233–1236.

[3] K. Ogata, M. Tachibana, J. Yamagishi, and T. Kobayashi, "Acoustic model training based on linear transformation and MAP modification for HSMM-based speech synthesis," in *Proc. ICSLP 2006*, Sept. 2006, pp. 1328–1331.

[4] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 1, pp. 325–333, Jan. 2007.

[5] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, pp. 187–207, 1999.

[6] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 5, pp. 816–824, May 2007.

[7] A.W. Black and K. Tokuda, "The Blizzard Challenge – 2005: Evaluating corpus-based speech synthesis on common datasets," in *Proc. EUROSPEECH 2005*, Sept. 2005, pp. 77–80, http://festvox.org/blizzard/blizzard2005.html.

[8] D. Arifianto, T. Tanaka, T. Masuko, and T. Kobayashi, "Robust F0 estimation of speech signal using harmonicity measure based on instantaneous frequency," *IEICE Trans. Information and Systems*, vol. E87-D, no. 12, pp. 2812–2820, Dec. 2004.

[9] H. Kawahara, H. Katayose, A. Cheveigné, and R. Patterson, "Fixed point analysis of frequency to instantaneous frequency mapping for accurate estimation of F0 and periodicity," in *Proc. EUROSPEECH 1999*, Sept. 1999, pp. 2781–2784.

[10] Entropic Research Laboratory Inc, *ESPS Programs Version 5.0*, 1993.

[11] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "A hidden semi-Markov model-based speech synthesis system," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 5, pp. 825–834, May 2007.

[12] K. Tokuda, H. Zen, and A.W. Black, "An HMM-based speech synthesis system applied to English," in *Proc. of IEEE Speech Synthesis Workshop*, Sept. 2002.

[13] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. ICSLP-96*, Oct. 1996, pp. 1137–1140.

[14] M.J.F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[15] J. Yamagishi, K. Ogata, Y. Nakano, J. Isogai, and T. Kobayashi, "HSMM-based model adaptation algorithms for average-voice-based speech synthesis," in *Proc. ICASSP 2006*, May 2006, pp. 77–80.

[16] J. Isogai, J. Yamagishi, and T. Kobayashi, "Model adaptation and adaptive training using ESAT algorithm for HMM-based speech synthesis," in *Proc. EUROSPEECH 2005*, Sept. 2005, pp. 2597–2600.

[17] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi, "Acoustic modelingof speaking styles and emotional expressions in HMM-based speech synthesis," *IEICE Trans. Inf. & Syst.*, vol. E88-D, no. 3, pp. 503–509, Mar. 2005.

[18] H. Zen, K. Tokuda, T. Masuko, T. Yoshimura, T. Kobayashi, and T. Kitamura, "State duration modeling for hmm-based speech synthesis," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 3, pp. 692–693, Mar. 2007.

[19] Y. Nakano, M. Tachibana, J. Yamagishi, and T. Kobayashi, "Constrained structural maximum a posteriori linear regression for average-voice-based speech synthesis," in *Proc. ICSLP 2006*, Sept. 2006, pp. 2286–2289.

An Excitation Model for HMM-Based Speech Synthesis Based on Residual Modeling

Ranniery Maia[†,‡], *Tomoki Toda*[†,††], *Heiga Zen*[†‡], *Yoshihiko Nankaku*[†‡], *Keiichi Tokuda*[†,†‡]

[†]National Inst. of Inform. and Comm. Technology (NiCT), Japan
[‡]ATR Spoken Language Comm. Labs, Japan
[††]Nara Institute of Science and Technology, Japan
[†‡]Nagoya Institute of Technology, Japan

ranniery.maia@atr.jp, tomoki@is.naist.jp, {zen,nankaku,tokuda}@sp.nitech.ac.jp

Abstract

This paper describes a trainable excitation approach to eliminate the unnaturalness of HMM-based speech synthesizers. During the waveform generation part, mixed excitation is constructed by state-dependent filtering of pulse trains and white noise sequences. In the training part, filters and pulse trains are jointly optimized through a procedure which resembles analysis-by-synthesis speech coding algorithms, where likelihood maximization of residual signals (derived from the same database which is used to train the HMM-based synthesizer) is pursued. Preliminary results show that the novel excitation model in question eliminates the unnaturalness of synthesized speech, being comparable in quality to the the best approaches thus far reported to eradicate the *buzziness* of HMM-based synthesizers.

1. Introduction

Speech synthesis based on Hidden Markov Models (HMMs) [1] represents a good choice for Text-to-Speech (TTS) with flexibility concerning the synthesis of voices with several styles [2] as well as portability [3]. Nevertheless, unnaturalness of the synthesized speech owing to the parametric way in which the final speech waveform is produced still represents a challenging issue, and attempts at solving this problem have become a research topic with growing interest.

The first approach to improve the quality of HMM-based synthesizers through the modification of the excitation model was reported by Yoshimura et al. [4]. It basically consisted in the modeling of the parameters encoded by the Mixed Excitation Linear Prediction (MELP) algorithm [5] by HMMs, jointly with mel-cepstral coefficients and $F0$. During the synthesis, these parameters were generated and used to construct mixed excitation in the same way as the MELP algorithm. Later, using the same philosophy Zen et al. proposed the utilization of the STRAIGHT vocoding method [6] for HMM-based speech synthesis. It consisted in the modeling of aperiodicity parameters by HMMs in order to enable the construction of the STRAIGHT parametric mixed excitation during the synthesis stage. Details of their implementation are reported in [7]. Aside from these two attempts to solve the problem in question, other approaches have also been recently reported [8, 9]. Although these methods improve the quality of the final waveform, minimization of the distortion between natural and synthesized speech has not been performed so far.

Considering the evolving steps of speech coders which make use of the source-filter model for speech production, significant improvement in the quality of the decoded speech can be achieved by analysis-by-synthesis (AbS) speech coders when compared with vocoders which attempt to heuristically generate the excitation source, such as linear predictive (LP) vocoding and MELP. As an illustration of the success of AbS coding schemes, the Code-Excited Linear Prediction (CELP) algorithm represents an important advance for speech coding with high-quality at low bit rates and has been standardized by many institutes and companies for mobile communications [10].

Concerning the TTS research field, Akamine and Kagoshima applied the philosophy of AbS speech coding to speech synthesis in a method so-defined Closed-Loop Training (CLT) [11]. It consisted in the derivation of speech units for concatenation by minimizing the distortion between natural and synthesized speech, after being modified by the PSOLA algorithm [12]. It was reported that speech synthesized through the concatenation of units selected from inventories designed by CLT achieves a high degree of smoothness (a traditional issue of concatenation-based systems) even for small corpora.

This paper describes a novel excitation approach for HMM-based speech synthesis based on the CLT procedure [13]. The excitation model consists of a set of state-dependent filters and pulse trains, which are iteratively optimized as the maximization of the likelihood of residual signals (which must be derived from the same database which is used to train the HMM-based synthesizer) is pursued. In the synthesis part the trained excitation model is employed to generate mixed excitation by inputting pulse train and white noise into the filters. The states in which the filters vary can be represented, for example, by leaves of decision-trees for mel-cepstral coefficients.

The rest of this paper is organized as follows: Section 2 outlines the proposed excitation method; Section 3 explains how the excitation model is trained, namely state-dependent filter determination and pulse train optimization; Section 4 concerns the waveform generation part; Section 5 shows some experiments; and the conclusions are in Section 6.

2. Proposed excitation model

The excitation scheme in question is illustrated in Figure 1. During the synthesis, the input pulse train, $t(n)$, and white noise sequence, $w(n)$, are filtered through $H_v(z)$ and $H_u(z)$, respectively, and added together to result in the excitation signal $e(n)$. The voiced and unvoiced filters, $H_v(z)$ and $H_u(z)$, respectively, are associated with each HMM state $s = \{1, \ldots, S'\}$,

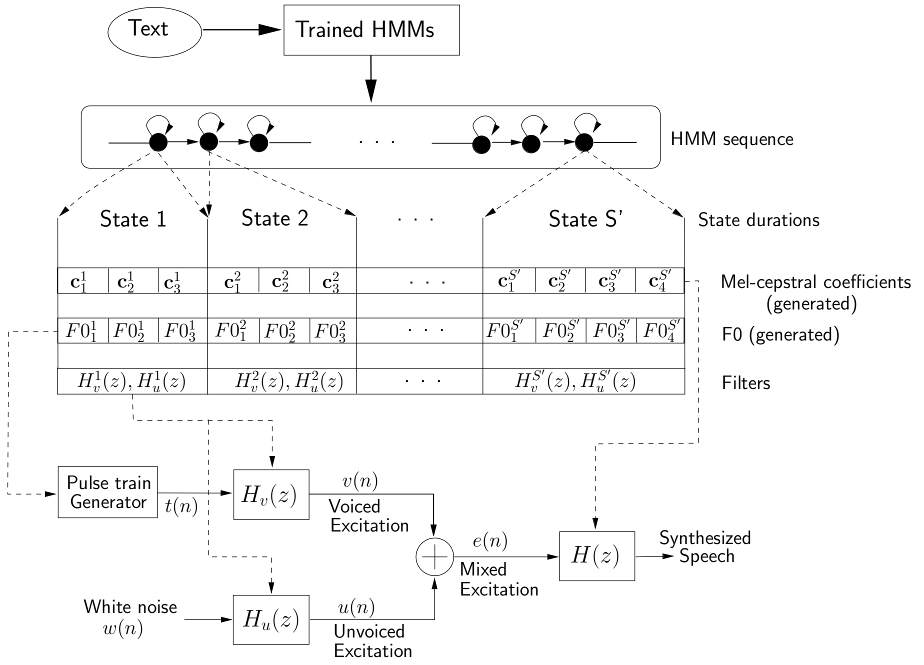

Text
Trained HMMs
HMM sequence
State 1
State 2
· · ·
State S'
State durations
$\mathbf{c}_1^1$ $\mathbf{c}_2^1$ $\mathbf{c}_3^1$ $\mathbf{c}_1^2$ $\mathbf{c}_2^2$ $\mathbf{c}_3^2$ · · · $\mathbf{c}_1^{S'}$ $\mathbf{c}_2^{S'}$ $\mathbf{c}_3^{S'}$ $\mathbf{c}_4^{S'}$
Mel-cepstral coefficients (generated)
$F0_1^1$ $F0_2^1$ $F0_3^1$ $F0_1^2$ $F0_2^2$ $F0_3^2$ · · · $F0_1^{S'}$ $F0_2^{S'}$ $F0_3^{S'}$ $F0_4^{S'}$
F0 (generated)
$H_v^1(z), H_u^1(z)$ $H_v^2(z), H_u^2(z)$ · · · $H_v^{S'}(z), H_u^{S'}(z)$
Filters
Pulse train Generator
$t(n)$
$H_v(z)$
$v(n)$
Voiced Excitation
$e(n)$
Mixed Excitation
$H(z)$
Synthesized Speech
White noise $w(n)$
$H_u(z)$
$u(n)$
Unvoiced Excitation

Figure 1: *Proposed excitation scheme for HMM-based speech synthesis: filters $H_v(z)$ and $H_u(z)$ are associated with each HMM state.*

as depicted in Figure 1, and their transfer functions are given by

$$H_v(z) = \sum_{l=-M/2}^{M/2} h(l) z^{-l}, \qquad (1)$$

$$H_u(z) = \frac{K}{1 - \sum_{l=1}^{L} g(l) z^{-l}}, \qquad (2)$$

where M and L are the respective orders.

2.1. Effect of $H_v(z)$ and $H_u(z)$

The function of the voiced filter $H_v(z)$ is to transform the input pulse train $t(n)$, yielding the signal $v(n)$ whose waveform is similar to the sequence $e(n)$, used as target during the excitation training. Because pulses are mostly considered in voiced regions, $v(n)$ is referred to as voiced excitation. The property of having finite impulse response leads to stability and phase information retention. Further, since the final waveform is synthesized off-line, a non-causal structure appears to be more appropriate.

Since white noise is assumed to be the input of the unvoiced filter, the function of $H_u(z)$ is thus to weight the noise - in terms of spectral shape and power - resulting in the unvoiced excitation component $u(n)$ which is eventually added to the voiced excitation $v(n)$ to form the mixed signal $e(n)$.

3. Excitation training

In order to visualize how the proposed model can be trained, the excitation generation part of Figure 1 is modified into the diagram of Figure 2, by considering $t(n)$ and $e(n)$ as input of the excitation construction block. In this case it can be seen that white noise is the output which results from filtering $u(n)$ through the inverse unvoiced filter $G(z)$.

By observing the system shown in Figure 2, an analogy with analysis-by-synthesis speech coders [10] can be made as follows. The target signal is represented by the residual $e(n)$, the error of the system is $w(n)$, and the terms whose incremental modification can minimize $w(n)$ in some sense are the filters $H_v(z)$ and $H_u(z)$, and pulse train $t(n)$.

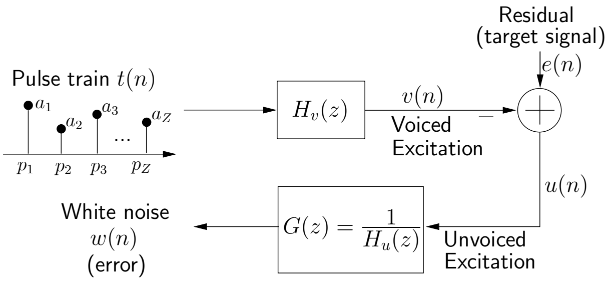


Figure 2: *Modification of the excitation scheme: pulse train and residual are the input while white noise is the output.*

Concerning the utilization of AbS to speech synthesis, the diagram of Figure 2 shows some similarities with the approach proposed by Akamine and Kagoshima [11]. However, aside from the fact that Akamine's scheme was intended to be applied to unit concatenation-based systems, other major differences between the proposed method and his scheme are:

- target signals correspond to residual sequences (not natural speech);
- the PSOLA modification part is replaced by the convolution between voiced filter coefficients and pulse trains;
- the error signal $w(n)$ is taken into account to derive the unvoiced component during the synthesis.

In the next two sections the AbS procedure which must be conducted, namely determination of the state-dependent filters and pulse train optimization are described.

3.1. Filter determination

The filters are determined in a way to maximize the likelihood of $e(n)$ given the excitation model (which comprises the voiced filter $H_v(z)$, unvoiced filter $H_u(v)$, and pulse train $t(n)$).

3.1.1. Likelihood of $e(n)$ given the excitation model

The likelihood of the residual vector $\mathbf{e} = [e(0) \cdots e(N-1)]^T$, with $[\cdot]^T$ meaning transposition and N being the whole database length in number of samples[1], given the voiced excitation vector $\mathbf{v} = [v(0) \cdots v(N-1)]$ and $\mathbf{G}$, is

$$P[\mathbf{e}|\mathbf{v}, \mathbf{G}] = \frac{1}{\sqrt{(2\pi)^N |\mathbf{G}^T\mathbf{G}|^{-1}}} e^{-\frac{1}{2}[\mathbf{e}-\mathbf{v}]^T\mathbf{G}^T\mathbf{G}[\mathbf{e}-\mathbf{v}]}, \quad (3)$$

with $\mathbf{G} = [\bar{\mathbf{g}}_0 \cdots \bar{\mathbf{g}}_{N-1}]$ being the $N \times (N+L)$ inverse unvoiced filter impulse response matrix, where each column

$$\bar{\mathbf{g}}_j = \begin{bmatrix} 0 \cdots 0 & 1/K_s & g_s(1)/K_s \cdots g_s(L)/K_s & 0 \cdots 0 \end{bmatrix}^T, \quad (4)$$

has respectively j and $(N + L - j)$ zeros before and after the coefficients of the inverse unvoiced filter, $\{1/K_s, g_s(1)/K_s, \ldots, g_s(L)/K_s\}$. The index $s = \{1, \ldots, S\}$ indicates the state in which the j-th database sample belongs to, and S is the total number of states considering the entire database. Therefore, considering this state-dependency, $\mathbf{v}$ can be written as

$$\mathbf{v} = \sum_{s=1}^{S} \mathbf{A}_s\mathbf{h}_s = \mathbf{A}_1\mathbf{h}_1 + \ldots + \mathbf{A}_S\mathbf{h}_S, \quad (5)$$

where $\mathbf{h}_s = [h_s(-M/2) \cdots h_s(M/2)]^T$ is the impulse response vector of the voiced filter for state s, and the term $\mathbf{A}_s$ is the overall pulse train matrix where only pulse positions belonging to state s are non-zero.

After substituting (5) into (3), and taking the logarithm, the following expression can be obtained for the log likelihood of the residual signal given the filters and pulse trains,

$$\log P[\mathbf{e}|H_v(z), H_u(z), t(n)] = -\frac{N}{2}\log(2\pi) + \frac{1}{2}\log(|\mathbf{G}^T\mathbf{G}|) - \frac{1}{2}\left[\mathbf{e} - \sum_{s=1}^{S}\mathbf{A}_s\mathbf{h}_s\right]^T \mathbf{G}^T\mathbf{G}\left[\mathbf{e} - \sum_{s=1}^{S}\mathbf{A}_s\mathbf{h}_s\right]. \quad (6)$$

3.1.2. Determination of $H_v(z)$

For a given state s, the corresponding vector of coefficients $\mathbf{h}_s$ which maximizes the log likelihood in (6) is determined from

$$\frac{\partial \log P[\mathbf{e}|H_v(z), H_u(z), t(n)]}{\partial \mathbf{h}_s} = 0. \quad (7)$$

The expression above results in

$$\mathbf{h}_s = \left[\mathbf{A}_s^T\mathbf{G}^T\mathbf{G}\mathbf{A}_s\right]^{-1}\mathbf{A}_s^T\mathbf{G}^T\mathbf{G}\left[\mathbf{e} - \sum_{\substack{l=1\\l\neq s}}^{S}\mathbf{A}_l\mathbf{h}_l\right], \quad (8)$$

which corresponds to the least-squares formulation for the design of a filter through the solution of an over-determined linear system [14].

3.1.3. Determination of $H_u(z)$

To visualize how the coefficients of $H_u(z)$ are derived, another expression which represent the log likelihood function should be considered. It can be noticed that

$$[\mathbf{e}-\mathbf{v}]^T\mathbf{G}^T\mathbf{G}[\mathbf{e}-\mathbf{v}] = \frac{1}{K^2}\sum_{n=0}^{N-1}\left[u(n) - \sum_{l=1}^{L} g(l)u(n-l)\right]^2, \quad (9)$$

and it can be verified [15] that

$$|\mathbf{G}^T\mathbf{G}|^{-1} = \prod_{n=0}^{N-1}\frac{K^2}{\left|1 - \sum_{l=1}^{L} g(l)e^{-j\omega_n l}\right|^2}. \quad (10)$$

After substituting (9) and (10) into (3), and taking the logarithm of the resulting expression, the following log likelihood function can be obtained,

$$\log P[u(n)|G(z)] = \sum_{n=0}^{N-1}\log\left(\left|1 - \sum_{l=1}^{L} g(l)e^{-j\omega_n l}\right|\right) - \frac{1}{2}\sum_{n=0}^{N-1}\left\{\log(2\pi K^2) + \frac{1}{K^2}\left[u(n) - \sum_{l=1}^{L} g(l)u(n-l)\right]^2\right\}. \quad (11)$$

Since $G(z)$ is minimum-phase, the first term in the right side of (11) becomes zero. By taking the derivative of the expression above with respect to K, it can be demonstrated that (11) is maximized with respect to $\{K, g(1), \ldots, g(L)\}$ when

$$K = \sqrt{\varepsilon_m}, \quad (12)$$

$$\varepsilon_m = \min_{g(1),\ldots,g(L)}\left\{\frac{1}{N}\sum_{n=0}^{N-1}\left[u(n) - \sum_{l=1}^{L} g(l)u(n-l)\right]^2\right\}, \quad (13)$$

that is, the problem can be interpreted as the autoregressive spectral estimation of $u(n)$ [15].

Considering segments of a particular state s as ensembles of a wide-sense stationary process, the mean autocorrelation sequence for s can be computed as the average of all short-time autocorrelation functions from all the segments belonging to s (analogous to the method presented in [16] for the periodogram), i.e.,

$$\bar{\phi}_s(k) = \frac{1}{\sum_{j=1}^{N_s} F_j}\sum_{j=1}^{N_s}\sum_{l=1}^{F_j}\phi_{s,j,l}(k), \quad (14)$$

where $\phi_{s,j,k}(k)$ is the short-term autocorrelation sequence obtained from the l-th analysis frame of the j-th segment of the state s; F_j is the number of analysis frames, and N_s is the number of segments of state s.

3.2. Pulse optimization

The second process carried out for the training of the excitation model consists in the optimization of the positions and amplitudes of $t(n)$. The procedure is conducted by keeping $H_v(z)$ and $H_u(z)$ constant for each state $s = \{1, \ldots, S\}$ and minimizing the mean squared error of the system of Figure 2. It can be noticed that regardless of $G(z)$ this error minimization is the same as maximizing (3).

The goal of the pulse optimization is to approach $v(n)$ to $e(n)$ so as to remove the short and long-term correlation of

[1] The entire database is considered to be contained in a single vector.

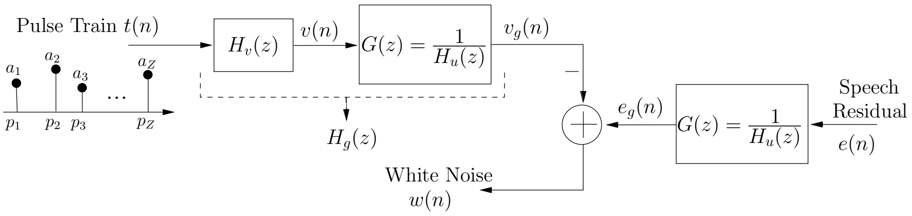


Figure 3: *Scheme for the amplitude and position optimization of the non-zero samples of* $t(n)$.

$u(n)$ during the filter computation process. The procedure is carried out in a similar way to the method employed by *Multi-pulse Excited Linear Prediction* speech coders [10]. These algorithms attempt to construct glottal excitations which can synthesize speech by using a few position and amplitude optimized pulses. In the present case, the optimization is performed in the neighborhood of the pulse positions.

3.2.1. Amplitude and position determination

To visualize the way the pulses are optimized, Figure 3 should be considered. The error of the system $\mathbf{w}$ is given by

$$\mathbf{w} = \mathbf{e}_g - \mathbf{v}_g = \mathbf{H}_g\mathbf{t}, \tag{15}$$

where $\mathbf{e}_g = [e_g(0) \ \cdots \ e_g(N-1+L)]$ is the $(N+L)$-length vector containing the overall residual signal $e(n)$ filtered by $G(z)$. The impulse response matrix $\mathbf{H}_g$ is

$$\mathbf{H}_g = \begin{bmatrix}\mathbf{h}_{g1} & \mathbf{h}_{g2} & \cdots & \mathbf{h}_{gN+L-1}\end{bmatrix}, \tag{16}$$

with each respective column given by

$$\mathbf{h}_{gj} = \begin{bmatrix}0\cdots0 & h_g\left(-\frac{M}{2}\right)\cdots h_g\left(\frac{M}{2}+L\right) & 0\cdots0\end{bmatrix}^T, \tag{17}$$

and the vector $\mathbf{t}$ contains non-zero samples only at certain positions, i.e,

$$\mathbf{t} = \begin{bmatrix}0 & \cdots & 0 & a_i & 0 & \cdots & 0 & a_{i+1} & \cdots & 0\end{bmatrix}^T. \tag{18}$$

Therefore, the voiced excitation vector $\mathbf{v}$ can be written as

$$\mathbf{v} = \mathbf{H}_g\mathbf{t} = \sum_{i=1}^{Z} a_i\mathbf{h}_{gi}, \tag{19}$$

where $\{a_1, \ldots, a_Z\}$ and $\{p_1, \ldots, p_Z\}$ are respectively the Z amplitudes and positions of $t(n)$ to be optimized.

The error to be minimized is

$$\varepsilon = \mathbf{w}^T\mathbf{w} = [\mathbf{e}_g - \mathbf{H}_g\mathbf{t}]^T[\mathbf{e}_g - \mathbf{H}_g\mathbf{t}]. \tag{20}$$

Substituting (19) into (20), the following expression results

$$\varepsilon = \mathbf{e}_g^T\mathbf{e}_g - 2\mathbf{e}_g\sum_{i=1}^{Z} a_i\mathbf{h}_{gi} + \sum_{i=1}^{Z} a_i^2\mathbf{h}_{gi}^T\mathbf{h}_{gi} + \sum_{i=1}^{Z}\sum_{\substack{j=1\\j\neq i}}^{Z} a_i a_j\mathbf{h}_{gi}^T\mathbf{h}_{gj}. \tag{21}$$

The optimal pulse amplitude a_i which minimizes (21) can thus be derived from $\frac{\partial\varepsilon}{\partial a_i} = 0$, which leads to

$$a_i = \frac{\mathbf{h}_{gi}^T\left[\mathbf{e}_g - \sum_{\substack{j=1\\j\neq i}}^{Z} a_j\mathbf{h}_{gj}\right]}{\mathbf{h}_{gi}^T\mathbf{h}_{gi}}, \tag{22}$$

and the best position, p_i, is the one which minimizes the resulting expression from the substitution of (22) into (21), i.e.,

$$p_i = \underset{p_i=1,\ldots,N}{\arg\max} \frac{\left[\mathbf{h}_{gi}^T\left(\mathbf{e}_g - \sum_{\substack{j=1\\j\neq i}}^{Z} a_j\mathbf{h}_{gj}\right)\right]^2}{\mathbf{h}_{gi}^T\mathbf{h}_{gi}}. \tag{23}$$

3.3. Recursive algorithm

The overall procedure for the determination of the filters $H_v(z)$ and $H_u(z)$, and optimization of the positions and amplitudes of $t(n)$ is described in Table 1. Pitch marks may represent the best choice to construct the initial pulse trains $t(n)$. The convergence criterion is the variation of the voiced filters.

Table 1: *Algorithm for joint filter computation and pulse optimization.* $\mathbf{I}_X$ *means identity matrix of size* X.

```
$t(n)$ initialization
1) For each utterance $l$
        1.1) Initialize $\{p_{l_1}, \ldots, p_{l_Z}\}$ based on
             the pitch marks
        1.2) Optimize $\{p_{l_1}, \ldots, p_{l_Z}\}$ according to (23),
             considering $\mathbf{H}_g = \mathbf{I}_{N+M+L}$
        1.3) Calculate $\{a_{l_1}, \ldots, a_{l_Z}\}$ according to (22),
             considering $\mathbf{H}_g = \mathbf{I}_{N+M+L}$
$H_v(z)$ initialization
1) For each state $s$
        1.1) Compute $\mathbf{h}_s$ from (8), considering $\mathbf{G} = \mathbf{I}_N$
2) Set voiced filter variation tolerance: $\epsilon_v$
3) Set the number of iterations: $N_{\text{iter}}$ and $N_{\text{iter}_{\text{max}}}$
Recursion
1) Make $\varepsilon_v = 0$
2) For each state $s$
        2.1) Make $\mathbf{h}_s{}^a = \mathbf{h}_s$
        2.2) Compute $\mathbf{h}_s$ by solving (8)
        2.3) Compute the voiced filter variation
             $\varepsilon_v = \varepsilon_v + [\mathbf{h}_s{}^a - \mathbf{h}_s]^T[\mathbf{h}_s{}^a - \mathbf{h}_s]$
3) For each state $s$
        3.1) Obtain the mean autocorrelation sequence
             of $u(n)$ under state $s$, from (14)
        3.2) Compute $\{g_s(1), \ldots, g_s(L)\}$ and $K_s$ from
             $\bar{\phi}_s(k)$ using the Levinson-Durbin algorithm
4) If $\varepsilon_v < \epsilon_v$ or $N_{\text{iter}} = N_{\text{iter}_{\text{max}}}$, go to (7)
5) For each utterance $l$
        5.1) Optimize $\{p_{1_l}, \ldots, p_{Z_l}\}$ according to (23)
        5.2) Calculate $\{a_{1_l}, \ldots, a_{Z_l}\}$ according to (22)
6) Return to (1)
7) End
```

4. Synthesis part

The synthesis of a given utterance is performed as follows. First, state durations, $F0$ and mel-cepstral coefficients are determined. Secondly, a sequence of filter coefficients is derived based on the state sequence of the referred input utterance. It can be noticed thus that while $F0$ and mel-cepstral coefficients vary at every 5 ms, filters change for each HMM state, as depicted in Figure 1. After that, pulse trains are constructed from $F0$ with no pulses assigned to unvoiced regions. Finally, speech is synthesized using the filters, pulse trains, mel-cepstral coefficients and white noise sequences.

Although it is not shown in Figure 1, the unvoiced component $u(n)$ is high-pass filtered with cutoff frequency of 2 kHz before being added to the voiced excitation $v(n)$. This procedure is performed to avoid the synthesis of *rough speech*.

5. Experiment

To verify the effectiveness of the proposed method, the CMU_ARCTIC database, female speaker SLT [17], was used to train the excitation model and the following HMM-based speech synthesizers:

- conventional system;
- system as described in [7], henceforth referred to as *blizzard system.*

The blizzard system was used to generate speech parameters (mel-cepstral coefficients, $F0$ and aperiodicity coefficients) for synthesis whereas the conventional system was employed to derive durations as well as the states of the excitation model. Filter orders were $M = 512$ and $L = 256$, and the residual signals were extracted by speech inverse filtering with the utilization of the Mel Log Spectrum Approximation (MLSA) structure [18].

5.1. The states

The states $\{1, \ldots, S\}$ were obtained by Viterbi alignment of the database using the trained HMMs of the conventional system. Eventually, these states were mapped onto a set of state clusters corresponding to leaves of specific decision-trees generated for the stream of mel-cepstral coefficients [18]. Therefore, in this sense, many states from the set $\{1, \ldots, S\}$ share the same pair of filter coefficients. The reason for clustering the distribution of mel-cepstral coefficients relies upon the assumption that residual sequences are highly correlated with their corresponding spectral parameters [19].

Aside from the decision of which parameters the states should be derived from, another important issue concerns the size of the tree and which information it should represent. According to experiments it was observed that good results can be achieved from small phonetic trees. Consequently, the decision-trees used to derive the *filter states* in this experiment were generated by using solely phonetic and phonemic questions. Furthermore, the parameter which controls the size of the trees was adjusted so as to generate a small number of nodes. At the end of the clustering process, 132 state clusters were achieved.

5.2. Effect of the CLT

Figure 4 shows a transitional segment of natural speech with three corresponding versions synthesized by natural spectra and $F0$, with the utilization of the following excitation schemes: (1) simple excitation; (2) parametric excitation created by the blizzard system; (3) the proposed approach. Residual and the corresponding excitations are also shown. One can see that the proposed method produces excitation and speech waveforms that are closer to the natural versions. This represents an effect of the CLT, where phase information from natural speech also tends to be reproduced in its synthesized version. For this example speech was synthesized using Viterbi aligned state durations.

5.3. Subjective quality

A comparison test was conducted with utterances generated by the blizzard system, simple excitation and proposed method. The results implied that the latter is similar in quality to the blizzard system. The overall preference for six listeners, each of them testing ten sentences (three comparison pairs per sentence), was:

- proposed: 60%;
- blizzard system: 58.3%;
- simple excitation: 31.7%.

It should be noted that these results, according to the directions given to the subjects, represent the overall quality provided by the excitation models, not the naturalness.

6. Conclusions

The proposed scheme synthesizes speech with quality considerably better than the simple excitation baseline. Furthermore, when compared with one of the best approaches thus far reported to eliminate the *buzziness* of HMM-based speech synthesis (the Blizzard Challenge 2005 version [7]), the proposed model presents the advantage of minimizing the distortion between natural and synthesized speech through a closed-loop training procedure. Although a full-fledged evaluation is necessary, it is expected that the excitation model in question may produce smooth and close-to-natural speech. Future steps towards the conclusion of this project include pulse train modeling for the waveform generation part and state clustering in a way to maximize the likelihood of residual sequences.

7. Acknowledgements

Special thanks to Dr. Shinsuke Sakai and Dr. Satoshi Nakamura for their important contributions.

8. References

[1] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. of EUROSPEECH*, 1999.

[2] J. Yamagishi, M. Tachibana, T. Masuko, and T. Kobayashi, "Speaking style adaptation using context clustering decision tree for HMM-based speech synthesis," in *Proc. of ICASSP*, 2004.

[3] K. Tokuda, H. Zen, and A. W. Black, "An HMM-based speech synthesis system applied to English," in *Proc. of IEEE Workshop in Speech Synthesis*, 2002.

[4] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Mixed-excitation for HMM-based speech synthesis," in *Proc. of EUROSPEECH*, 2001.

[5] A. McCree, K. Truong, E. George, T. Barnwell, and V. Viswanathan, "A 2.4 kbits/s MELP candidate for the U.S. Fdereal Standard," in *Proc. of ICASSP*, 2006.

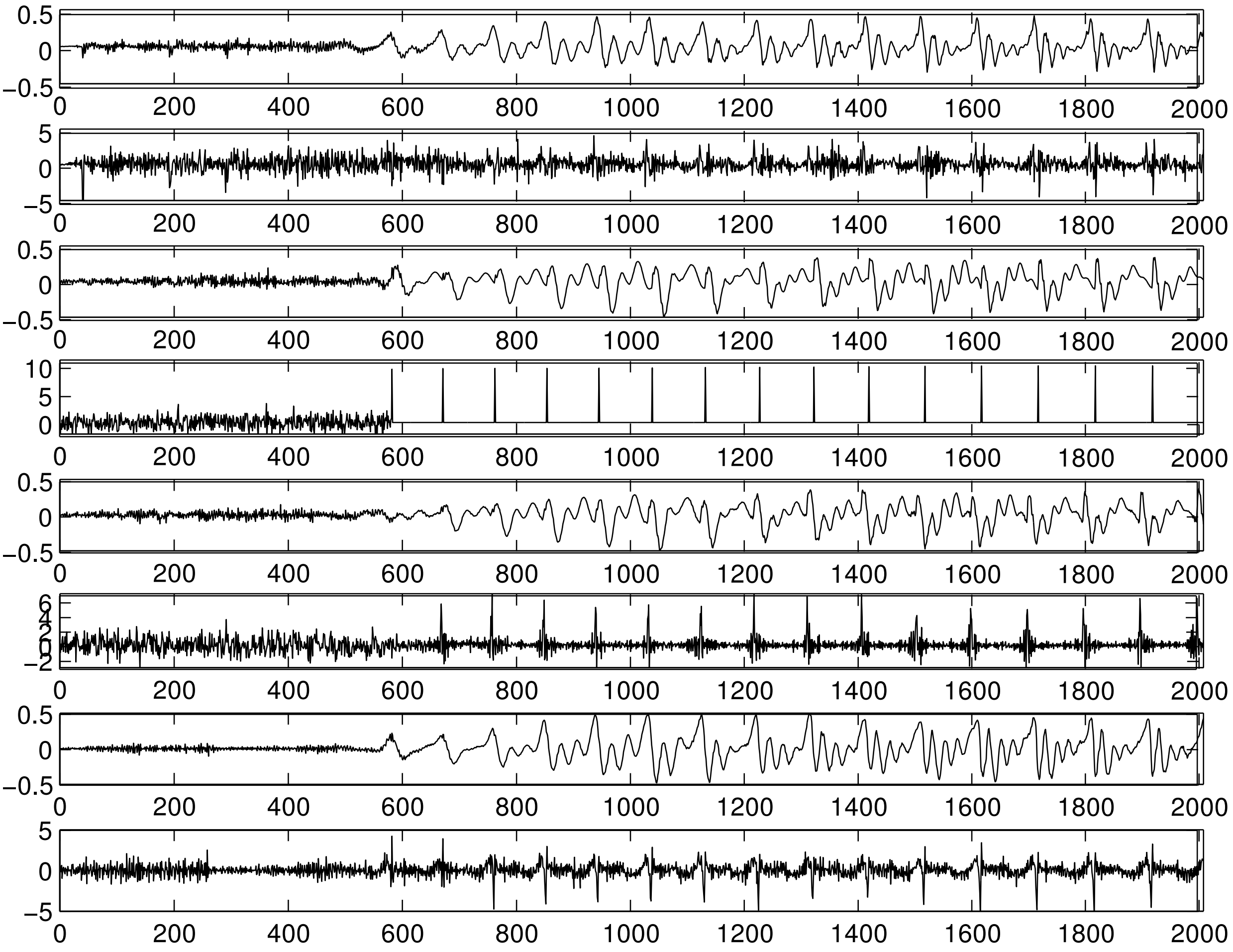


Figure 4: *Waveforms from top to bottom: natural speech, residual, speech synthesized by simple excitation, simple excitation, speech synthesized by the blizzard system, excitation constructed according to the blizzard system, speech synthesized by the proposed method, and excitation constructed according to the proposed method.*

[6] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, Apr. 1999.

[7] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of the Nitech HMM-based speech synthesis for Blizzard Challenge 2005," *IEICE Trans. on Inf. and Systems*, vol. E90-D, no. 1, 2007.

[8] S. J. Kim and M. Hahn, "Two-band excitation for HMM-based speech synthesis," *IEICE Trans. Inf. & Syst.*, vol. E90-D, 2007.

[9] O. Abdel-Hamid, S. Abdou, and M. Rashwan, "Improving the Arabic HMM based speech synthesis quality," in *Proc. of ICSLP*, 2006.

[10] W. Chu, *Speech Coding Algorithms*. Wiley-Interscience, 2003.

[11] M. Akamine and T. Kagoshima, "Analytic generation of synthesis units by closed loop training for totally speaker driven text to speech system (TOS drive TTS)," in *Proc. ICSLP*, 1998.

[12] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, vol. 9, Dec. 1990.

[13] R. Maia, T. Toda, H. Zen, Y. Nankaku, and K. Tokuda, "Mixed excitation for HMM-based speech synthesis based on state-dependent filtering," in *Proc. of Spring Meeting of the Acoust. Society of Japan*, 2007.

[14] L. B. Jackson, *Digital filters and signal processing*. Kluwer Academics, 1996.

[15] J. D. Markel and A. H. Gray, Jr., *Linear prediction of speech*. Springer-Verlag, 1986.

[16] P. Welch, "The use of Fast Fourier Transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms," *IEEE Trans. Audio and Electroacoustics*, vol. 15, June 1967.

[17] http://festvox.org/cmu_arctic.

[18] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. of ICASSP*, 1992.

[19] H. Duxans and A. Bonafonte, "Residual conversion versus prediction on voice morphing systems," in *Proc. of ICASSP*, 2006.

An HMM-Based Bilingual (Mandarin-English) TTS

Hui Liang[1] *, *Yao Qian*[2] *and* *Frank K. Soong*[2]

[1]School of Information Security Engineering, Shanghai Jiaotong University, P.R.China
[2]Microsoft Research Asia, Beijing, P.R.China
{v-huilia,yaoqian,frankkps}@microsoft.com

Abstract

We propose to build an HMM-based, Mandarin and English, bilingual TTS system. Starting with a simple baseline of two TTS systems built separately from Mandarin and English databases recorded by the same speaker, we construct a new, mixed-language TTS by designing language specific and independent questions to facilitate phone sharing across the two languages. With shared phones, the new system has a smaller footprint than the baseline system. The synthesis quality is either the same for non-mixed, Mandarin or English synthesis as the baseline or much better for mixed-language synthesis. The higher quality of mixed-language synthesis is confirmed by preference scores of 60.2% vs 39.8%, obtained in a subjective listening test. A preliminary Mandarin synthesis experiment was also performed by using the model parameters in the leaf nodes of an English decision tree where Kullback-Leibler divergence is used to establish the nearest neighbor based mapping between leaf nodes in the decision trees of the two languages. A subjective transcription test shows a character accuracy of 93.9%.

1. Introduction

The quality of text-to-speech synthesis has been greatly improved in the recent years. Various telecommunication applications, e.g. information inquiry, reservation and ordering, and email reading, demand higher synthesis quality than current TTS systems can provide. In these applications, a multilingual TTS system, in which one engine can synthesize multiple languages or even mixed-languages, is in a great demand to serve international business needs. However, most TTS systems can only deal with a single language in that sentences of voice databases are pronounced by a single native speaker. Although multilingual text can be correctly read by switching voices or engines at each language change, it is unfeasible for code-switched text in which the language changes occur within a sentence as words or phrases. Furthermore, with the widespread use of mobile phones or embedded devices, a small footprint of speech synthesizers is rather preferred for applications based on those devices. The requirement of multilingual TTS with a small footprint is a big challenge to the current research of TTS technologies.

There are many studies on multilingual TTS systems [1-6]. In [2], it defines a multilingual text-to-speech system that uses a common algorithm for multiple languages. In that context, a collection of language-specific synthesizers does not qualify as a multilingual system. Phonetic coverage can be achieved by collecting multilingual speech data, but language-specific information, e.g. specialized text analysis, is also required [5]. A global phone set, which uses the smallest phone inventory to cover all phones of the languages affected, has been tried in multilingual or language-independent speech recognition and synthesis [7]. It adopts phone sharing with the phonetic similarity measured by data-driven clustering methods [8,9] or phonetic-articulatory features defined by IPA [10,11]. There are also intense interests on the small footprint aspects of TTS systems. HMM-based speech synthesis [12] is a more successful one among them. The small footprint (≤ 2M) of an HMM synthesizer has made it an ideal choice for embedded systems. It has been successfully applied to speech synthesis of many monolinguals, e.g. English, Japanese and Mandarin [13,14], and multilingual [15]. In [15], an average voice is firstly trained by using mixed speech from several speakers in different languages, then the average voice is adapted to a specific speaker. Consequently the specific speaker is able to speak all the languages contained in the training data.

Nowadays, English words or phrases embedded in Mandarin utterances are getting more popularly used among students and educated people in China. Mandarin and English belong to different language families. Those two languages are highly unrelated in that seldom phones can be shared together according to their IPA symbols. A bilingual (Mandarin-English) TTS is conventionally built based on pre-recorded Mandarin and English sentences uttered by a bilingual speaker [6]. The unit selection module of the system is shared across languages, while phones from different languages are not shared with each other. Such an approach has certain shortcomings. The footprint of such a system is large, i.e., about twice the size of a single language system. In practice, it is also not easy to find plenty of professional bilingual speakers to build multiple bilingual voice fonts for various applications.

Although the phones from English and Mandarin are not highly sharable, but their subphonemic productions may still be similar. Complex phonemes may be rendered by two or three simple phonemes. Furthermore, numerous allophones, which are used in specific phonetic contexts, provide more chances for phone sharing between Mandarin and English. In this paper, we propose to use context-dependent HMM state sharing for our bilingual (Mandarin-English) TTS system. A state level mapping is also investigated for new language synthesis without recording data. The whole approach is based on the framework of HMM-based speech synthesis. In this framework, spectral envelopes, fundamental frequencies, and state durations are modeled simultaneously by corresponding HMMs. For a given text sequence, speech parameter trajectories and corresponding signals are then generated from trained HMMs in the Maximum Likelihood (ML) sense.

2. Phone Sharing

* An intern in Speech Group, Microsoft Research Asia

For building a bilingual, Mandarin and English, TTS system, the first step is to decide a phone set to cover all speech sounds in those two languages. Additionally, we also hope such a phone set can be compact enough to facilitate phone sharing across languages and make a reasonable sized TTS model. We use the following two approaches to find possible phone sharing candidates.

2.1. IPA Scheme

International Phonetic Alphabet (IPA) is an international standard to transcribe speech sounds of any spoken language. It classifies phonemes according to their phonetic-articulatory features. Phonemes of different languages labeled by the same IPA symbol should be considered as the same phoneme by ignoring the language-dependent aspects of speech perception. All phonemes in English and Mandarin are listed in Table 1, where the English phoneme set consists of 24 consonants, 11 simple vowels and five diphthongs, while the Mandarin phoneme set is a finer set [18] which consists of 27 simple consonants, 30 consonants with a glide and 36 tonal vowels.

	English	Mandarin
Unvoiced plosive	/kʰ/ /pʰ/ /tʰ/	/kʰ/ /pʰ/ /tʰ/ /k/ /p/ /t/
Voiced plosive	/b/ /d/ /g/	
Unvoiced fricative	/f/ /s/ /h/ /ʃ/ /θ/	/f/ /s/ /ʂ/ /x/ /ɕ/
Voiced fricative	/ʒ/ /ð/ /v/ /z/	/ʐ/
Unvoiced affricative	/ʧ/	/ʦʰ/ /tʂʰ/ /ʨʰ/ /ʦ/ /tʂ/ /ʨ/
Voiced affricative	/ʤ/	
Nasal	/m/ /n/ /ŋ/	/m/ /n/[1] /ŋ˥/ /ŋ˩/ /ŋ˧/ /n˥/ /n˩/ /n˧/ [2]
Lateral approximant	/l/	/l/
Approximant	/w/ /j/ /ɹ/	
Front rounded		/y/[3] /y˥/ /y˩/ /y˧/ [4]
Front unrounded	/ɛ/ /a/ /ɪ/ /æ/ /iː/	/a˥/ /a˩/ /a˧/ /ɛ˥/ /ɛ˩/ /ɛ˧/ /i/[3] /i˥/ /i˩/ /i˧/ [4] /ʅ˥/ /ʅ˩/ /ʅ˧/ /ɿ˥/ /ɿ˩/ /ɿ˧/
Central unrounded	/ə/ /əː/	/ɚ˥/ /ɚ˩/ /ɚ˧/
Back rounded	/ʊ/ /uː/ /ɔː/	/o˥/ /o˩/ /o˧/ /u/[3] /u˥/ /u˩/ /u˧/ [4]
Back unrounded	/ʌ/	/ɑ˥/ /ɑ˩/ /ɑ˧/ /ɤ˥/ /ɤ˩/ /ɤ˧/
Diphthong	/aʊ/ /aɪ/ /oʊ/ /ɔɪ/ /eɪ/	

Table 1: All IPA phonemes in English and Mandarin. [1] Used as a syllable onset (Initial); [2] Used as a syllable coda; [3] Used as a glide; [4] Used as a syllable nucleus or coda

By checking the table for sharable phones, we found only eight consonants, /kʰ/, /pʰ/, /tʰ/, /f/, /s/, /m/, /n/ and /l/, and two vowels (ignoring the tone information), /ɛ/ and /a/, can be shared between the two languages according to their IPA symbols.

2.2. K-L Divergence Measure

The Kullback-Leibler divergence (KLD) is an information-theoretic measure of (dis)similarity between two probability distributions. When the temporal structure of HMMs is aligned by dynamic programming, KLD can be further modified to measure the difference between HMMs of two evolving speech sounds [16,17]. For two given distributions P and Q of continuous random variables, the symmetric form of KLD between P and Q is:

$$D_{KL}(P,Q)=\int p(x)\log\frac{p(x)}{q(x)}dx+\int q(x)\log\frac{q(x)}{p(x)}dx \quad (1)$$

where p and q denote the densities of P and Q. For two multivariate Gaussian distributions, Eq. (1) has a closed form:

$$\begin{aligned}D_{KL}(P,Q)=\frac{1}{2}tr\{&(\mathbf{\Sigma}_p^{-1}+\mathbf{\Sigma}_q^{-1})(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)^T \\ &+\mathbf{\Sigma}_p\mathbf{\Sigma}_q^{-1}+\mathbf{\Sigma}_q\mathbf{\Sigma}_p^{-1}-2\mathbf{I}\}\end{aligned} \quad (2)$$

where $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ are the corresponding mean vectors and covariance matrices, respectively.

Each phone in Table 1 is acoustically represented by a context-independent HMM with 5 emitting states. Each state output pdf is a single Gaussian with a diagonal covariance matrix. The spectral feature used for measuring the KLD between any two given HMMs is the first 24 LSPs out of the 40-demensional LSP since the most perceptually discriminating spectral information is located in the lower frequency range. The data used for training those HMMs contain 1,024 English and 1,000 Mandarin sentences, respectively. We use Eq. (2) to calculate KLD between every pair of speech sounds, modeled by their respective HMMs. The 16 English vowels and their nearest neighbors measured by KLD from all vowels of English and Mandarin are listed in Table 2. We find that only six English vowels whose nearest neighbors are Mandarin vowels exist and there are two-to-one mappings, e.g. both /eɪ/ and /ɪ/ are mapped to /ɛ˥/, among those six vowels.

English Vowel	Nearest Neighbor	KLD
/a/	/ʌ/ (E)	8.09
/æ/	/ɛ/ (E)	2.85
/ʌ/	/a/ (E)	8.09
/ɔː/	/o˥/ (C)	13.84
/aʊ/	/a/ (E)	17.28
/ə/	/ɤ˥/ (C)	8.61
/aɪ/	/ʌ/ (E)	29.52
/ɛ/	/æ/ (E)	2.85
/əː/	/ɹ/ (E)	18.09
/eɪ/	/ɛ˥/ (C)	17.78
/ɪ/	/ɛ˥/ (C)	10.07
/iː/	/ɪ/ (E)	12.66
/oʊ/	/o˥/ (C)	10.87
/ɔɪ/	/o˥/ (C)	43.92
/ʊ/	/uː/ (E)	7.04
/uː/	/ʊ/ (E)	7.04

Table 2: English vowels and their nearest neighbors measured by KLD from all vowels of English and Mandarin

3. State Sharing and Mapping

Mandarin is a tonal language of the Sino-Tibetan family, while English is a stress-timed language of the Indo-European family. It is not too surprising that the analysis results shown in Section 2 suggest that English phonemes are quite different from Mandarin phonemes. However, since the speech production is constrained by limited movement of articulators, it may be possible to find more sharing of acoustic attributes at a granular, sub-phone level. Many complex phonemes can be well rendered by two or three phonemes, e.g. an English diphthong may be similar to a Mandarin vowel pair. Moreover, allophones, e.g., the Initial 'w' /u/ in Mandarin corresponds to [u] in syllable 'wo' and [v] in syllable 'wei', provide more chances for phone sharing between Mandarin and English under certain contexts. Therefore, we propose to use context-dependent HMM state level sharing for a bilingual (Mandarin-English) TTS system. A state level mapping is also investigated for new language synthesis without recording data.

3.1. Context-dependent State Sharing across Languages

In HMM-based TTS, phone models of rich contexts, e.g. tri-phone, quin-phone models or models with even more and longer contexts like phone positions and POS, are used to capture acoustic co-articulation effects between neighboring phonemes. In practice, however, limited by insufficient training data, we almost always have to tie models of rich contexts into more generalized ones so as to predict unseen contexts more robustly in testing. State tying via a clustered decision tree is commonly used.

The phone set we used is the union of all the phones in English and Mandarin, while states from different central phones across different languages are allowed to be tied together in training the bilingual HMMs. The questions used in growing decision trees include:

a) Language-independent questions: e.g. *Velar_Plosive*, Does the state belong to velar plosive phones, which contain /g/ (Eng.), /k^h/ (Eng.), /k/ (Man.) or /k^h/ (Man.)?
b) Language-specific questions: e.g. *E_Voiced_Stop*, Does the state belong to English voiced stop phones, which contain /b/, /d/ and /g/?

According to manner and place of articulations, supra-segmental features, etc., we construct questions so as to tie states of English and Mandarin phone models together.

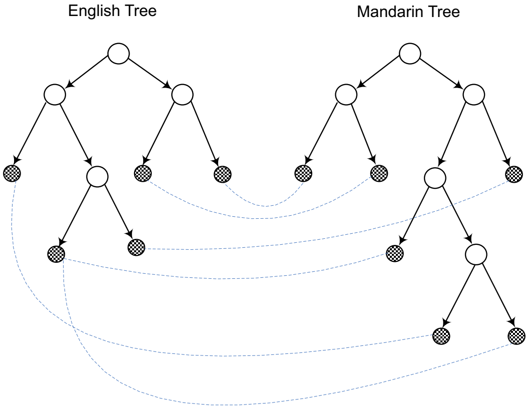
In total, 85,006*5 context-dependent states are generated. Among them, 43,491*5 states are trained from 1,000 Mandarin sentences and the rest from 1,024 English ones. All context-dependent states are then clustered into a decision tree. Such a mixed, bilingual, decision tree has only about 60% leaf nodes of a system by combining two separately trained, English and Mandarin TTS systems. We found that about one fifth of the states are tied across languages, i.e. 37,871 Mandarin states are tied together with 44,548 English states.

3.2. Context-dependent State Mapping

A straightforward way to build a bilingual, Mandarin and English, TTS system is to use pre-recorded Mandarin and English sentences uttered by the same speaker. However, it is not so easy to find professional speakers who are fluent in both languages when we need to build an inventory of bilingual voice-fonts of multi-speakers. Also, it is an open research topic on how to synthesize a different target language when only monolingual recording of a source language from a speaker is available. We propose to establish a tied, context-dependent state mapping across different languages from a bilingual speaker first and then use it as a basis to synthesize other monolingual speakers' voice in the target language.

First, we built two language-specific decision trees by using the bilingual data recorded by one speaker. Each leaf node in the Mandarin decision tree has a mapped leaf node, in the minimum K-L divergence sense, in the English one. The tied, context-dependent state mapping (from Mandarin to English) is shown in Fig. 1. The directional mapping from Mandarin to English can have more than one leaf nodes in the Mandarin tree mapped to one leaf node in the English tree. As shown in the figure, two nodes in the Mandarin tree are mapped into one node in the English tree. The mapping from English to Mandarin is similarly done but in a reverse direction; i.e., for every English leaf node, we find its nearest neighbor, in the minimum KLD sense, among all leaf nodes in the Mandarin tree.

Figure 1: The illustration of a tied, context-dependent state mapping from a Mandarin decision tree to an English decision tree.



In HMM-based speech synthesis, spectral and pitch features are separated into two streams and stream-dependent models are built to cluster two features into separated decision trees. Pitch features are modeled by MSD-HMM, which was proposed to model two, discrete and continuous, probability spaces, discrete for unvoiced regions and continuous for voiced F0 contours [19]. The upper bound of KLD between two MSD-HMMs is written as:

$$
\begin{aligned}
D_{KL}(P,Q) \le {} & (w_0^p - w_0^q)\log\frac{w_0^p}{w_0^q} + (w_1^p - w_1^q)\log\frac{w_1^p}{w_1^q} \\
& + \frac{1}{2} tr\{(w_1^p \mathbf{\Sigma}_p^{-1} + w_1^q \mathbf{\Sigma}_q^{-1})(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)^T \\
& + w_1^p(\mathbf{\Sigma}_p \mathbf{\Sigma}_q^{-1} - \mathbf{I}) + w_1^q(\mathbf{\Sigma}_q \mathbf{\Sigma}_p^{-1} - \mathbf{I})\} \\
& + \frac{1}{2}(w_1^q - w_1^p)\log\left|\mathbf{\Sigma}_p \mathbf{\Sigma}_q^{-1}\right|
\end{aligned} \tag{3}
$$

where w_0 and w_1 are prior probabilities of unvoiced and voiced subspaces, respectively. Both English and Mandarin have trees of spectrum, pitch and duration. Each leaf node of those trees is used to set a mapping between English and Mandarin.

To synthesize speech in a new language without pre-recorded data from the same voice talent, we can utilize the mapping established with bilingual data and new monolingual data recorded by a different speaker. For example, a context-dependent state mapping trained from speech data of a bilingual (English-Mandarin) speaker A can be used to choose the appropriate states trained from speech data of a different, monolingual Mandarin speaker B to synthesize English sentences. The same structure of decision trees should be used for Mandarin training data from speakers A and B.

4. Experiments and Evaluations

4.1. Experimental Setup

A broadcast news style speech corpus recorded by a female speaker is used in this study. The training data consist of 1,000 Mandarin sentences and 1,024 English sentences, which are both phonetically and prosodically rich [14]. The testing data consist of 50 Mandarin, 50 English and 50 mixed-language sentences. Speech signals are sampled at 16 kHz, windowed by a 25-ms window with a 5-ms shift, and the LPC spectral features are transformed into 40th-order LSPs and their dynamic features. Five-state left-to-right HMMs with single, diagonal Gaussian distributions are adopted for training phone models. We built three HMM TTS systems as follows.

System I: Direct combination of HMMs (Baseline)

This is the baseline system, where language-specific, Mandarin and English HMMs and decision trees are trained separately. In the synthesis part, input text is converted first into a sequence of contextual phone labels through a bilingual TTS text-analysis frontend (Microsoft Mulan) [6]. The corresponding parameters of contextual states in HMMs are retrieved via language-specific decision trees. Then LSP, gain and F0 trajectories are generated in the maximum likelihood sense. Finally, speech waveforms are synthesized from the generated parameter trajectories. In synthesizing a mixed-language sentence, depending upon the text segments to be synthesized is Mandarin or English, appropriate language-specific HMMs are chosen to synthesize corresponding parts of the sentence.

System II: State Sharing across languages

In this system, both 1,000 Mandarin sentences and 1,024 English sentences are used together for training HMMs. Context-dependent state sharing across languages as discussed in Section 3.1 is applied. In synthesis, decision trees of mixed-languages are used instead of the language-specific ones in System I. Since there are no mixed-language sentences in the training data, the context of phones at a language switching boundary, e.g. the left phone or the right phone, is replaced with the nearest context in the language which the central phone belongs to in the text analysis module. For example, the triphone /ɔɪ/(E)-/ʂ/(C)+/ʅ˥/(C) will be replaced with /o˥/(C)-/ʂ/(C)+/ʅ˥/(C), where the left context /o˥/(C) is the nearest Mandarin substitute for /ɔɪ/(E) according to the KLD measure.

System III: State Mapping across languages

This is an oracle experiment. A preliminary study is carried out to synthesize sentences of a language without pre-recorded data. We built language-specific decision trees and used a tied state mapping across different languages, as discussed in Section 3.2, for the experiment. To evaluate the upper bound quality of synthesized utterances in the target language, we use the same speaker' voice in this experiment when extracting state mapping rules and synthesizing the target language.

4.2. Evaluations and Analysis

Table 3 shows a comparison of the number of tied states or leaf nodes in decision trees of LSP, log F0 and duration, and corresponding average log probabilities of System I and System II in training. In the table, it is observed that the total number of tied states (HMM parameters) of System II is about 40% less, when compared with those of System I. But the log probability per frame obtained in training System II is almost the same as that of System I.

Table 3: The numbers of tied states and average log probabilities of System I and System II in the training phrase

		System I		System II
		Mandarin	English	
The num of states	LSP	1728	1791	2064
	Log F0	2971	4337	3518
	Duration	2389	2402	1607
Average log prob per frame		5.699e+02	5.659e+02	5.661e+02

4.2.1. *Evaluation Results of System I and System II*

Objective evaluation

Synthesis quality is measured objectively in terms of distortions between original speech and speech synthesized by System I and II. Since the predicted HMM state durations of generated utterances are in general not the same as those of original speech, we first measure the root mean squared error (RMSE) of phone durations of synthesized speech. Spectra and pitch distortions are then measured between original speech and synthesized speech where the state durations of the original speech (obtained by forced alignment) are used for speech generation. In this way, both spectrum and pitch are compared on a frame-synchronous basis between the original and synthesized utterances.

Table 4 shows the averaged log spectrum distance, RMSE of F0 and phone durations evaluated in 100 test sentences (50 Mandarin and 50 English) generated by system I and system II. It indicates that the distortion difference between Systems I and II in terms of log spectrum distance, RMSEs of F0 and duration are negligibly small.

Table 4: Log spectrum distance, RMSE of F0 and duration of the test sentences generated in Systems I, II and the original

	System I		System II	
	Mandarin	English	Mandarin	English
Log spectrum distance (dB)	3.964	4.485	4.022	4.524
RMSE of F0 (Hz)	17.17	23.31	17.69	22.81
RMSE of Duration (s)	0.0366	0.0578	0.0370	0.0571

Subjective evaluation

Informal listening to the monolingual sentences synthesized by Systems I and II confirms the objective measures shown in Table 4: i.e. there is hardly any difference, subjective or objective, in 100 sentences (50 Mandarin, 50 English) synthesized by Systems I and II.

The 50 mixed-language sentences generated by the two systems are evaluated subjectively in an AB preference test by nine subjects. The results of preference test are shown in Figure 2. The preference score of System II (60.2%) is significantly higher than System I (39.8%) (α = 0.001, CI = [0.1085, 0.3004]). The main perceptually noticeable difference in the paired sentences synthesized by Systems I and II is at the transitions between English and Chinese words in the mixed-language sentences. State sharing through tied states across Mandarin and English in System II helps to alleviate the problem of segmental and supra-segmental discontinuities between Mandarin and English transitions. Since all training sentences are either exclusively Chinese or English, there is no specific training data to train such language-switching phenomena. As a result, System I, without any state sharing across English and Mandarin, is more prone to the synthesis artifacts at the switches of English and Chinese words.

Overall, System II, which is obtained via efficient state tying across different languages and with a significantly smaller HMM model size than System I, can produce the same synthesis quality for non-mixed language sentences and better synthesis quality for mixed-language ones.

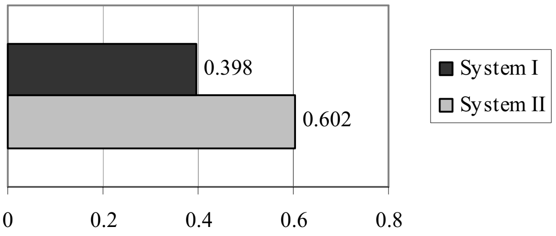


Figure 2: The preference test result of 50 mixed-language sentences

4.2.2. *Evaluation Results of System III*

Fifty Mandarin test sentences are synthesized by English HMMs in System III. Five subjects were asked to transcribe the 50 synthesized sentences to evaluate their intelligibility. A Chinese character accuracy of 93.9% is obtained.

An example of F0 trajectories predicted by Systems I (broken line) and III (solid line) is shown in Fig 3. As shown in the figure, possibly due to the MSD modeling of voice/unvoiced stochastic phenomena and KLD measure used for state mapping, the voice/unvoiced boundaries are well aligned between the two trajectories generated by System I and III. Furthermore, the rising and falling trend of F0 contours in those two trajectories is also well-matched. However, F0 variation predicted by System III is smaller than that by System I. After analyzing the English and Mandarin training sentences, we find that the variance of F0 in Mandarin sentences is much larger than that in English ones. Both means and variances of the two databases are shown in Table 5. The much larger variance of Mandarin sentences is partially due to the lexical tone nature of Mandarin where the variation in four (or five) lexical tones increases the intrinsic variance or the dynamic range of F0 in Mandarin. This is clearly shown in Table 5.

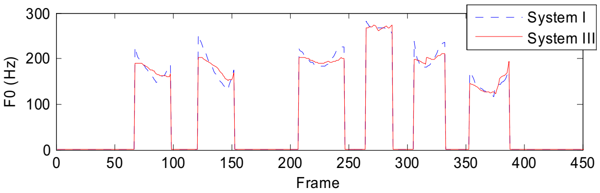


Figure 3: An example of F0 trajectories predicted by System I and system III.

Table 5: The mean and variance of Mandarin and English training sentences.

	Mandarin	English
Mean (Hz)	198.5	198.3
Variance	2462.1	1398.1

5. Conclusions

In this paper, we propose to build an HMM-based bilingual (Mandarin-English) TTS system. Language-specific and language-independent questions are designed for clustering states across two languages in one single decision tree. The experimental results show that the new TTS system with context-dependent HMM state sharing across languages outperforms the simple baseline system where two language-dependent HMMs are used together. In addition, state mapping across languages based upon the Kullback-Leibler divergence is used to synthesize Mandarin speech using model parameters in an English decision tree. The preliminary experimental results show that thus synthesized Mandarin speech is highly intelligible.

6. Acknowledgements

The authors are grateful to Prof. Keiichi Tokuda and Dr. Heiga Zen, Department of Computer Science, Nagoya Institute of Technology, Japan for providing us the HMM-based TTS training tool, HTS on their website: http://hts.sp.nitech.ac.jp/.

7. References

[1] Mareüil, P. B. and Soulage, B., "Input/output Normalization and Linguistic Analysis for a Multilingual Text-to-speech Synthesis System", *Proc. of 4th ISCA Speech Synthesis Workshop*, 2001.

[2] Sproat, R.(Editor), *Multilingual text-to-speech synthesis: the Bell Labs approach*, Kluwer Academic Publisher, 1998.

[3] Quazza, S., Donetti, L., Moisa, L. and Salza, P. L., "Actor®: A Multilingual Unit-selection Speech Synthesis System", *Proc. of 4th ISCA Speech Synthesis Workshop*, 2001.

[4] Black, A.W. and Lenzo, K. A., "Multilingual Text-to-Speech Synthesis", *Proc. of ICASSP*, 2004.

[5] Pfister, B. and Romsdorfer, H., "Mixed-lingual Text Analysis for Polyglot TTS Synthesis", *Proc. of Eurospeech*, 2003.

[6] Chu, M., Peng, H., Zhao, Y., Niu, Z. Y. and Chang, E., "Microsoft Mulan - a Bilingual TTS System", *Proc. of ICASSP*, 2003.

[7] Schultz, T. and Waibel, A., "Language-independent and Language-adaptive Acoustic Modeling for Speech Recognition", *Speech Communication, Vol. 35, pp. 31-51, August 2001. Issues 1-2.*

[8] Kohler, J., "Multilingual Phone Model for Vocabulary-Independent Speech Recognition Tasks", *Speech Communication 35 (2001) 21-30.*

[9] Yu, S., Zhang, S. and Xu, B., "Chinese-English Bilingual Phone Modeling for Cross-Language Speech Recognition", *Proc. of ICASSP*, 2004

[10] Bandino, L., Claudis, B. and Silvia, Q., "A General Approach to TTS Reading of Mixed-Language Texts", *Proc. of ICSLP*, 2004.

[11] Stuker, S., Metze, F., Schultz, T., and Waibel, A., "Integrating Multilingual Articulatory Features into Speech Recognition", *Proc. of Eurospeech*, 2003.

[12] Tokuda, K., Kobayashi, T., Masuko, T., Kobayashi, T., and Kitamura, T., "Speech Parameter Generation Algorithms for HMM-based Speech Synthesis", *Proc. of ICASSP*, 2000.

[13] Tokuda, K., Zen, H. and Black, A.W., "An HMM-based Speech Synthesis System Applied to English", *2002 IEEE Speech Synthesis Workshop*, 2002.

[14] Qian, Y., Soong, F. K., Chen, Y. and Chu, M., "An HMM-Based Mandarin Chinese Text-to-Speech System", *Proc. of ISCSLP*, 2006.

[15] Latorre, J., Iwano K., and Furui S., "New approach to the polyglot speech generation by means of an HMM-based speaker adaptable synthesizer", *Speech Communication, Vol. 48, pp. 1227-1242*, 2006.

[16] Myrvoll, T. A. and Soong, F. K., "Optimal Clustering of Multivariate Normal Distributions Using Divergence and its Application to HMM Adaptation", *Proc of ICASSP*, 2003.

[17] Zhao, Y., Zhang, C., Soong, F. K., Chu, M. and Xiao, X., "Measuring Attribute Dissimilarity with HMM KL-Divergence for Speech Synthesis", *Accepted by the 6th ISCA Speech Synthesis Workshop*, 2007.

[18] Huang, C., Shi, Y., Zhou, J. L., Chu, M., Wang, T., and Chang, E., "Segmental Tonal Modeling for Phone Set Design in Mandarin LVCSR", *Proc. of ICASSP 2004*, 2004.

[19] Tokuda, K., Masuko, T., Miyazaki, N., and Kobayashi, T., "Multi-space Probability Distribution HMM", *IEICE Trans. Inf. & Syst., E85-D(3):pp. 455-464*, 2002.

Data-driven Approach to Rapid Prototyping Xhosa Speech Synthesis

Justus C. Roux, Albert S. Visagie

Centre for Language and Speech Technology, Stellenbosch University, South Africa
jcr@sun.ac.za, avisagie@dsp.sun.ac.za

Abstract

This paper presents work in progress towards building a Xhosa speech synthesizer. HTS is being used for this purpose due to certain desirable properties. As a minority language, linguistic resources for Xhosa are limited despite a variety of impressionistic phonetic studies, prompting a minimalist approach and a preference for data-driven methods. Xhosa is an agglutinative language, and is also held to be a tonal language, which therefore requires morphological analysis and tonal information in order to generate intelligible speech. By taking into account more recent findings on the nature of Xhosa prosody, it appears that a minimalist approach that excludes tone information is possible. We implement the system using HTS. Such a data-driven TTS system is a useful tool to test various syntactic and other features in text that influence Xhosa prosody.

1. Introduction

This paper reflects on ongoing work towards the development of a text-to-speech (TTS) system for an African tone language, Xhosa. No serious attempts have yet been made to develop a general TTS system for this language. A limited domain synthesis application has been built in the *African Speech Technology project* (http://www.ast.sun.ac.za). An attempt towards the development of such a system for a sister language, Zulu, has recently been made by Louw et al. [1]. A feature of this 'general-purpose' synthesizer, however, was that it did not attempt to model intonation.

Xhosa like many other minority languages, lacks linguistic resources that are required for TTS. For example, the language is held to be a tone language, however, impressionistic tonal descriptions are extremely diverse in nature as has been previously indicated by Roux [2], [3]. This leads us to seek a minimalistic approach. In this work, we join current debate in the field, cf. Roux [4], Downing [5], and Kuun et al. [6], and suggest that tone assignment on individual syllables may not be that necessary to construct a highly intelligible Xhosa TTS system. This position challenges entrenched assumptions about the tonal nature of the language. In Section 4, we discuss subjective tests with encouraging results.

The immediate research aims of a broader study in this field are

- to determine what linguistic features are salient for text-to-speech synthesis of Xhosa,
- to build a front-end capable of deriving the features from the text, and
- to create a test bed from which the tonal and/or accentual properties of the language could be assessed through further experimentation.

This paper will reflect on a particular approach followed to create an intelligible Xhosa TTS system. We chose to use HTS for its ability to automatically draw correlations between symbolic features derived from the text and the observed acoustics cf. [7]. This is ideal for this work, since the text-analysis front-end is the only language dependent part of the resulting synthesizer. We also hope that using HTS will let us gauge the importance of various features by judging their effect on the output, and so provide further insight into what is needed for Xhosa TTS.

2. Linguistic features of Xhosa

In this section some of the basic linguistic features of Xhosa are listed which need to be taken into account in the development of a TTS system for this language.

2.1. Tone and vowel duration

Xhosa is regarded as a **tone language** belonging to the Nguni group of Bantu languages. It is spoken in South Africa by approximately 7,5 million people, i.e. by nearly 16% of the total population.

The language is highly agglutinative which means words are formed by combining a wide range of morphemes with word stems either as prefixes, infixes or suffixes. Hence, the word for a preacher 'umfundisi' derives from a verbal stem / –fund- / 'teach' with the following morphemes attached:

/u + m(u) + fund + is + i /(1) (1)

Although tone is not indicated orthographically lexical tone is realized on each syllable in the final surface form, hence

[úmfúndì:sì] (preacher/ one who teaches) (2)

The low tone of the deleted /u/ is maintained on the preceding nasal /m/. As morphemes are added to this form the tonal pattern may change:

/u + m(u) + fund + is + ana
(-ana denotes diminutive) (3)
[úmfùndísà:nà] (small preacher)

Note the H(igh) tone shifts to the antepenultimate syllable, whilst syllable length on /i/ is likewise shifted to the penultimate syllable. This is an important point that will permeate further discussions.

Three tones are traditionally distinguished in Xhosa, i.e. H(igh) [΄], L(ow) [`] and F(alling) [^]. Apart from dialectical variations in tonal patterns, impressionistic descriptions are extremely inconsistent, as has been pointed out in some detail by Roux [3]. Claughton [8] for example, introduces the use of superscript x tonal markings to indicate "free variation" in tonal realization in Xhosa, whilst trying to establish particular tonological rules. The point is that the empirical bases of impressionistic tonal descriptions in Xhosa are suspect; descriptions rarely stretch beyond observations of the production of a single "ideal" mother-tongue speaker of the language. Tonological descriptions more than often reflect the impressionistic interpretations of the researcher, generalizing on the performance of a single mother-tongue speaker; references or access to large speech databases from which conclusions are drawn are non-existent.

An important observation by Downing [5] regarding tone, stress and focus in phonological phrases, provides a new angle when she argues that High tone realisations in Xhosa shows "culminativity effects" that make the tone system resemble stress-accent systems. In stress-accent systems main stress tends to occur on syllables "...at the edge of a stem or word." Likewise High tones in Xhosa are restricted to occur at word edges, i.e. they regularly appear on the antepenultimate, penultimate or final syllable. Compare examples (2) and (3) above where the High tone on the antepenultimate syllable /fú/ in (2) shifted to the antepenultimate /dí/ in (3) when more syllables were added. This perceived preference for a High tone to appear in an antepenultimate syllable corroborates results of an informal investigation by Roux [4] on the allocation of 'prominence' (expressed in terms of H and concomitant increase in amplitude) to successive syllables by mother-tongue speakers of Xhosa. Results obtained for Zulu nouns and adjectives in the experimental work of Kuun et al. [6] also suggest a positional bias for H tones in the penultimate or antepenultimate syllable of the sister language of Xhosa.

Another important phenomenon that contributes to the metrical structure of Xhosa is the predictable assignment of **length (duration)** to particular syllables in a phonological word and/or phrase. Vowel lengthening normally takes place in the penultimate syllable of a word (in isolation), a phrase (demarcated by a following colon, or particular conjunctive words) or sentence (demarcated by a following full stop).

Given the query above on the representativeness of existing tonal data for Xhosa, and taking the observations of Downing [5], Roux [4] and Kuun et al. [6] into account, we adopt a simple syllable counting approach as features for the prediction of tone and duration as mentioned in 3.2.1 and 4.1 below. The observed position of High tone placement on the antepenultimate syllable of a long word, indicating some form of prominence (accent), as well as the predictability of vowel duration, are two aspects under investigation with the aim to create acceptable intonation contours for Xhosa.

2.2. Orthography, morphemes and letter-to-sound rules

Xhosa employs a conjunctive orthography, which together with the agglutinative nature of the language, poses a challenge for the construction of a lexicon.

Hence, a single 'word' may actually represent a phrase or a sentence:

/u + za + ku + ba + fund + is + a/ > uzakubafundisa
"He/she will teach them." (4)

The form above actually comprises concordial morphemes (/u/ and /ba/), morphemes indicating future tense (/za/, /ku/ and /a/), a verbal stem (/fund/), and a causative morpheme (/is/). In order to identify these morphemes (and other parts of speech such as nouns, verbs, adverbs) it is necessary to invoke a **morphological analyzer** for Xhosa. This analyzer identifies parts of speech, which may be useful for experimentation with prosody prediction in HTS (see also 3.2.2 below).

Fortunately the orthographic representation of Xhosa is very phonetic in nature which simplifies the creation of **grapheme-to-phoneme rules** for the language. An original set of rewrite rules developed by Roux [9] was recently updated and improved by Louw [10], and forms the basis for transforming orthographic forms into appropriate canonical phonetic representations for synthesis.

2.3. Segmental phonetic issues

In the development of a TTS system for Xhosa a few idiosyncratic segmental features of the language need to be taken into account. One of the most characteristic features of the language is the presence of **click sounds**; three different click sounds are identified: a dental click (represented orthographically as 'c'), an alveo-palatal click (represented orthographically as 'q'), and an alveo-lateral click (represented othographically as 'x'). Each of these (unvoiced) click types have four further phonetic attributes, rendering a total of fifteen different click sounds as listed below as represented in the orthography:

	Dental	Alveo-palatal	Alveo-lateral
Unvoiced	c	q	x
Aspirated	ch	qh	xh
Voiced	gc	gq	gx
Nasalized	nc	nq	nx
Voiced Nasal	ngc	ngq	ngx

Due to the fact that many of these clicks are rare, and in view of the desire to minimize the size of the phoneme set of the synthesizer to the most succinct possible set, the unvoiced and aspirated varieties, as well as the nasalized and voiced nasalized were lumped together.

The phenomenon of **tonal depression** is widely mentioned in literature. It implies that an H tone following a voiced consonant will be relatively lower in pitch than an H tone following a voiceless consonant. This phenomenon as well as other phenomena such as **segmental deletions** and **vowel devoicing** at word endings have not been treated in any special way as this will be derived from context information by HTS.

3. Implementation

This work used HTS for synthesis and Festival [11] for front-end processing. Following [12], we implement various standard Festival modules for Xhosa. The resulting Festival utterance structures are used to obtain features for HTS.

The only language resources available to us at the outset were the aforementioned manually developed letter-to-sound rules. Consistent with the plight of all minority languages, this scarcity of resources is a major constraint in building Xhosa TTS systems.

3.1. HTS back-end

HTS was chosen as a synthesizer for its desirable characteristics [7]. Specifically, HTS draws correlations between acoustic features and symbolic input features derived from text, making it possible to use it as a black-box, more than other methods. It is reported to work well on small datasets [13].

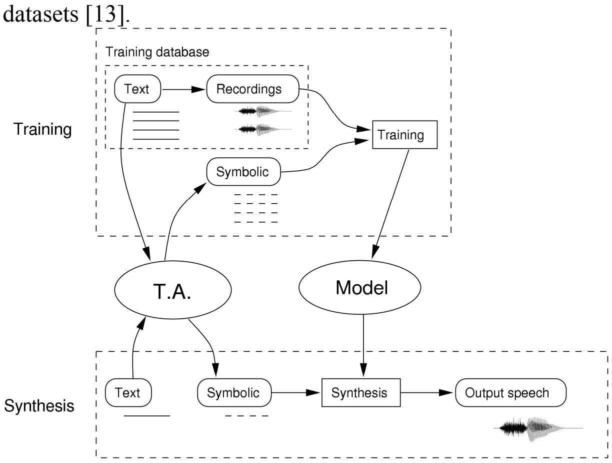


Figure 1: High-level overview of HTS system development. The text-analysis (T.A) component is constructed first. It is used both in training and during synthesis.

Figure 1 shows a high level summary view of how HTS works. The input to the system is a training database consisting of matching text and audio – one prompt per file.

The text component is converted to a phoneme sequence in the front-end (labeled Text-Analysis (T.A.)). Each phoneme label is augmented with symbolic features that describe its context. The contexts used in our system are described in Section 3.3.1.

The signal processing step extracts spectral information and average F0 and a voiced/unvoiced decision every 5ms. These features alone allow resynthesis of the audio.

The training process then finds the locations of the HMM-state-sized segments, 5 per phoneme, and clusters acoustically similar segments together using decision tree clustering.

An important characteristic of the clustering is that the duration of each HMM-state is performed separately for F0, duration and spectrum. Each of these factors are influenced by different contextual features, and as such it does not segment the training set unnecessarily.

HTS allows experimentation with various different features and tree clustering questions. To this end, it is an excellent tool to explore and test theories about the influence of various syntactical and morphological properties of the text on the synthesizer's ability to predict prosody, and allow guided incremental improvement of the text processing front-end.

The HMM synthesis framework trains Hidden Markov Models on a set of features extended from the normal speech recognition usage. The features contain Mel cepstrum parameters for modeling the spectrum, and F0 and duration distributions. It performs context clustering using decision trees separately for the spectrum, duration and F0 components, since different contextual information influences these properties of the surface realization.

3.2. Data

Text was collected in the form of two issues of a local tabloid, a novel used in Xhosa language teaching and several government documents explaining various services. The goal was to use edited texts such as these in order to get good quality sentences. It proved to be rather difficult to find appropriate material in electronic format.

Finally, we had 357 recordings, some containing entire paragraphs. There are 3339 words in the recordings. After cutting the recordings into 759 phrases, 43 minutes of speech remained.

The front-end of the synthesizer (described below) was used to obtain phoneme sequences for each utterance. Initial phonetic alignments were made using eHMM, bundled in the FestVox distribution [12]. eHMM produces alignments using forced alignment with a set of HMM models in the Festival voice's own phoneme set. The means and variances of the Gaussian components of the models are flat-started to the global mean and variance of the acoustic data, and then trained using embedded re-estimation. Roughly 10% of the alignments were checked manually, and all were found to be very accurate.

3.3. Front-end

Festival applies several modules during its text processing stages: tokenization, POS tagging, syntactic analysis, phrasing, orthographic to phonetic conversion, syllabification and post-lexical rules. The remaining modules' functions (F0, duration, loudness etc.) are performed in HTS.

The aforementioned letter-to-sound rules fit perfectly in Festival's module for rewrite rules, and so were easy to incorporate. The synthesizer training set contained only handful of loan words, and these constituted the lexicon. The lexicon had no stress or tone assignment.

The phoneme-set was determined by the output of the letter-to-sound rules – a total of 82 phonemes. Of these, many were deemed to be very close to each other, and were merged, yielding a final phoneme set of 63 symbols. The variety of consonants mentioned above explains the need for such a relatively large set.

We used the punctuation decision tree in Festival for phrasing.

The current system does not perform any post-lexical changes on the utterances. As it seems very context dependent, and open to speaker specific interpretation, we relied on the data available to HTS.

3.3.1. Symbolic features & questions for HTS

At the time of writing, the system outputs these features into the HTS label files:

- Phonetic context, two segments preceding and two following.
- Word position in the sentence.
- Syllable counts from the end of the utterance, and end of the phrase. The observation that the phrase-penultimate syllable is always lengthened to indicate the end of a phrase motivates this.
- Syllable position in the word, both from the start of the word, as well as from the end of the word. For example:

“Okubaluleke” yields these segments and syllable positions: `O: 1-6, k: 2-5, u: 2-5, b: 3-4, a: 3-4, l: 4-3, u: 4-3, l: 5-2, E: 5-2, k: 6-1` and `E: 6-1.`

Although Xhosa is generally held to be a tone language, recent studies [4,5,6] showed that the location of high tones is dependent on position within words and is regularly tied to the antepenultimate or penultimate syllable. The syllable position feature is a minimalist attempt to exploit this regularity in light of the absence of linguistic resources, and recent opinion in the field of Xhosa intonation study.

The question set includes the usual (in HTS) questions about various phoneme properties, such as phonemes types, voicing, place of articulation etc., adapted for Xhosa.

3.3.2. Role of morphological analysis

The next step in improving the synthesizer is to perform morphological analysis on the words.

As shown in examples (1-3) above, the tonological structure of the language is influenced by the specific prefixes and suffixes used to compose the word, whether or not each prefix or suffix carries its own high or low tone.

Morphological analysis will enable experimentation with prefix and suffix types as features for predicting prosody in HTS.

An analyzer for Zulu has been developed by Bosch and Pretorius [14], and work towards adapting it for Xhosa is currently underway. The first prototype, used in this work, contains a lexicon of all the morphological roots in the training set.

It is still possible however to interpret isolated words as containing various root morphemes or even different parts-of-speech. Some form of disambiguation given the sentence context of the word remains to be done.

Some classes of words, such as conjunctions, are not composed morphologically, or can be enumerated easily and therefore form small closed sets. Work is underway to produce a lexicon of these words that provides their parts-of-speech and supposed tone-markings. The system will consult this lexicon before attempting morphological parsing.

4. Experiments

The synthesizer was evaluated in a very small intelligibility test. Eight stimuli from two versions of the synthesizer, and eight obtained by resynthesizing the extracted spectral and F0 features were played to three mother-tongue, and two second language speakers.

The two versions of the synthesizer differed only in that one excluded the features indicated syllable position in words.

The mother-tongue speakers could understand all the stimuli nearly perfectly. Each of the three mother-tongue speakers indicated that they had trouble with at least one or two of the stimuli. Each of them had difficulty with different prompts. In each of these cases, the listeners were still able to give a very nearly correct “phonetic” transcription.

In each such case we feel that the segmental realization of the prompt was good, and that confusion was caused by bad prosody.

Both second language listeners understood the resynthesized prompts perfectly. However, they only understood slightly more than half of the synthesized prompts. This shows that the mother-tongue speakers’ results are not quite as encouraging as it might seem.

4.1. Syllable position and accent or stress

Subjective comparisons between the same synthesized utterances before and after adding only the word-level syllable counts indicates a significant positive effect of syllable position in words on the rendition of rhythm and intonation. This is obtained without including any explicit accent or stress markings. Several comparative examples, including natural speech may be found at http://www.sun.ac.za/su_clast/tts.html.

That this one feature made such a significant difference to the prosody seems to support the stress-accent side of recent debate about Xhosa tonology.

Syllabic prominence was generally predicted well for longer words. Short words such as pronouns were usually de-emphasized compared to the naturally pronounced versions. The classical tone markings and better parts-of-speech tagging are being explored as a means of providing information for predicting better prosody for these shorter words.

4.2. Clicks

As mentioned before, we lumped together aspirated and unvoiced click sounds. One listener felt that the unvoiced version was produced in a word that contains the aspirated version in one example. The dental click sound is dominated by examples of the unvoiced version. Experimentation is still needed to test the perception of clicks as produced by the synthesizer given information at various granularities.

That said, HTS models click sounds well. In initial subjective tests, listeners generally had no trouble distinguishing between the renditions of types of clicks.

5. Future work

We plan to experiment with various ideas of placing accent or predicting tone in the near future. Morphological analysis forms an integral part. The current system only used the morphological parsing results to determine (still ambiguous) parts-of-speech. In the near future we will incorporate information about the boundary between prefixes and the root morpheme first, and then add morpheme types, such as those indicating tense, negatives and diminutive forms.

Explicitly marking syllable prominence, especially for short the words in the current training and development set prompts, should form an interesting experiment to determine the validity of the stress-accent point of view.

Once subjective listening tests indicates acceptable performance, we want to construct a Blizzard style test [15], incorporating preference tests between a small number of systems and intelligibility tests.

The tests should incorporate synthesis of minimal pairs currently considered to be distinguished by tone. There are very few, and they tend to have different parts-of-speech.

6. Conclusions

The Xhosa and Zulu languages’ agglutinating nature and tone structure are generally held to be the greatest hurdles to building TTS systems. We feel that the minimalist approach taken here indicates that good synthesis is already possible with simpler features. The modern data-driven approach relieves one from much of the theoretical effort.

This work is to be used in embedded applications for two projects building translation and educational reference systems at Stellenbosch.

7. Acknowledgements

Thanks to Nick Campbell, Kamiya Tosirou and the team at ATR for an enriching and stimulating visit to Keihanna, Japan.

Special thanks to Prof. Sonja Bosch and her team at the University of South Africa (UNISA) for the use of the morphological analysis tools for Zulu and Xhosa.

We express our appreciation towards the National Research Foundation for the sponsorship of this project (GUN 2074784) in terms of the Japan-South African Intergovernmental Science and Technology Cooperation programme. All opinions expressed here are that of the authors.

8. References

[1] Louw, J.A., Davel, M. and Barnard, E. "A general purpose isiZulu TTS system." *South African Journal of African Languages*, 25(2): 92-100, 2005.

[2] Roux, J.C. "On the perception and production of tone in Xhosa." *South* African *Journal of African Languages*, 15(4): 196-204, 1995.

[3] Roux, J.C. "On the perception and description of tone in the Sotho and Nguni languages." *Proc. of the 3rd Int. Symposium on Cross Linguistic Studies of Tonal Phenomena*. Tokyo University of Foreign Studies, Tokyo. Ed. S Kaji, pp 155- 176, 2003.

[4] Roux, J.C. "Xhosa: A tone-or pitch-accent language?" *South African Journal of African Languages*, Supplement 36, 33-50, 1998.

[5] Downing, L.J. "Stress, Tone and Focus in Chichewa and Xhosa." *Stress and Tone: The African Experience.* Ed. R-J. Anyanwu. Ruediger Koeppe Verlag, Cologne, 2003.

[6] Kuun, C., Zimu, V., Barnard, E. and Davel, M. "Statistical investigations into isiZulu intonation." *Proc. of the 16th Annual Symposium of the Pattern Recognition Association of South Africa*, 111-115, 2005.

[7] http://hts.sp.nitech.ac.jp/

[8] Claughton, J.S. *The Tonology of Xhosa.* Unpublished Doctoral Thesis, Rhodes University, South Africa, 291pp.

[9] Roux, J.C. "Grapheme to phoneme conversions in Xhosa." *South African Journal of African Languages*, 9(2): 74-78, 1989.

[10] Louw, P. "A new definition of Xhosa grapheme-to-phoneme rules for automatic transcription." *South African Journal of African Languages*, 25(2):71-91, 2005.

[11] http://festvox.org/

[12] Dijkstra, J., Pols, L.C.W., van Son, R.J.J.H., "Frisian TTS, an Example of Bootstrapping TTS for Minority Languages", in *5th ISCA Speech Synthesis Workshop*, 2004.

[13] Maia, R. da S., Zen, H., Tokuda, K., Kitamura, T., Resende, F.G.V. Jr., "Towards the development of a Brazilian Portuguese text-to-tpeech system based on HMM", in *Eurospeech, Geneva*, 2003, pp. 2465-2468.

[14] Bosch, S.E., Pretorius, L., "Finite-state computational morphology: An analyzer prototype for Zulu", *Machine Translation,* 18:191–212, 2003.

[15] http://festvox.org/blizzard/

CRF-based Statistical Learning of Japanese Accent Sandhi for Developing Japanese Text-to-Speech Synthesis Systems

Nobuaki Minematsu[†], *Ryo Kuroiwa*[‡], *Keikichi Hirose*[‡], *Michiko Watanabe*[†]

† Graduate School of Frontier Sciences, The University of Tokyo
‡ Graduate School of Information Science and Technology, The University of Tokyo
{mine,kuroiwa,hirose,watanabe}@gavo.t.u-tokyo.ac.jp

Abstract

In Japanese, every content word has its own H/L pitch pattern when it is uttered isolatedly, called accent type. In a TTS system, this lexical information is usually stored in a dictionary and it is referred to for prosody generation. When converting a written sentence to speech, however, this lexical H/L pattern is often changed according to the context, known as word accent sandhi. This accent change is troublesome for speech synthesis researchers because it is difficult even for native speakers to describe explicitly what kind of mechanism is working for the change although young Japanese learn the mechanism without trouble. For developing a good Japanese TTS system, this implicit and phonological knowledge has to be built in the system. In our previous study [1], we developed a rule-based module for the accent sandhi but it is true that it produced an unignorable number of errors. In this paper, the development of a corpus-based module is described using Conditional Random Fields (CRFs) to predict the change. Although the new module shows the better performance for the prediction than the previous rule-based module, the new module is tuned further by integrating the rule-based knowledge acquired in the previous study.

1. Introduction

Several functions, such as text analysis, grapheme to phoneme conversion, and speech waveform generation, need to be developed to realize a TTS conversion system. Among them, the generation of prosodic features from an input text is very important and requires a sophisticated process, since no information on prosody is directly given in the text. Especially in the case of Japanese, the control of fundamental frequency (henceforth F_0) movement is crucial to achieve the high quality in the synthetic speech. In order to realize a good prosody control, the location of the accent nucleus should be adequately estimated for each accentual phrase as well as the boundaries of prosodic clauses (breath groups), prosodic phrases, and accentual phrases.

An accentual phrase of Japanese is often composed of two words or more, typically a content word followed by a function word. Although all the content words (and some function words) have their own accent nucleus position as their lexical attribute, the accent nucleus of an accentual phrase often shifts due to the accent sandhi. This accent shift has to be correctly predicted in TTS conversion. Some rules of the accent sandhi can be found in some accent dictionaries such as [2] but they are in abstract form and not adequate to be used for TTS conversion systems. Sagisaka *et al.* formulated these rules in a good shape [3, 4], which were widely adopted in Japanese TTS conversion researches [5]. In our previous study [1], a rule-based module was developed by extending Sagisaka's rules partly.

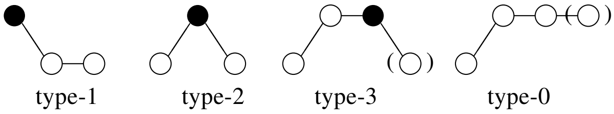


Figure 1: Accent types observable in 3-mora words of Tokyo dialect of Japanese

However, it is true that covering all the accent sandhi phenomena by rules is very difficult. In [3, 4], only the locations of primary accent nuclei were considered with the problem of secondary accent nuclei unsolved. Further, the sentences including function word concatenation were not adequately treated, either. To solve these problems, a corpus-based approach has been taken recently. In [6], n-gram models were used to develop a morphological analyzer which can produce the H/L attribute for each mora[1] of an input sentence. To take a corpus-based approach, a large corpus with accurate accent labeling is naturally required but we don't have any publicly available accent corpus. In this paper, at first, we developed a text corpus with accurate accent labeling, which will be publicly available in the near future. Using the corpus built so far, we developed a corpus-based module of predicting the accent change for adequate prosody generation. Further, the module was tuned by integrating the rule-based knowledge acquired in the previous study.

2. Word accent sandhi rules of Japanese

2.1. Word accent of Japanese

Word accent is one of the lexical attributes specific to each word and it is represented by a sequence of binary F_0 levels (H/L) in mora unit. Although it implies 2^N different accent types for N-mora words, the number of accent types for N-mora words of Tokyo dialect is reduced to $N+1$ due to the following properties.

1. A rapid rising or falling of F_0 has to occur between the first mora and the second one.
2. The number of the rapid falling pattern(s) of F_0 between two consecutive morae in a word is one at most.

Accent type showing a rapid downfall of F_0 immediately after the n-th mora is called type-n word accent and the n-th mora in this case is called accent nucleus. Fig. 1 shows the four accent types of 3-mora words of Tokyo dialect and their accent nuclei indicated by filled black circles. It should be noted that type-0 accent means that there is no accent nucleus and that type-0 accent and type-n accent of n-mora words are identical if they

[1] Mora is the minimum linguistic unit for speech production in Japanese, the size of which is rather similar to that of syllable.

are uttered isolatedly. The difference between the two is observed only when they are produced in connected speech. When a function word follows a type-n word, a falling pattern of F_0 is found immediately after the word. On the other hand, there is no falling patterns for type-0 words. In Fig. 1, a parenthesized circle represents the first mora of the following function word.

2.2. Word accent sandhi rules of Japanese

When a word is concatenated with another to form an accentual phrase, the resulting position of the accent nucleus of the phrase is often different from any positions of the original nuclei of the constituent words. The word accent sandhi can be categorized into three types;

1. **Shift** of the accent nucleus
   ア⌝カ + エンピツ → アカエ⌝ンピツ
   red pencil
2. **Generation** of the accent nucleus
   ケイタイ+ デンワ → ケイタイデ⌝ンワ
   portable telephone
3. **Deletion** of the accent nucleus
   ケ⌝イザイ + テキ → ケイザイテキ
   economy (suffix) economical

The word accent sandhi in Japanese was well formulated for TTS research in [3, 4]. The following sections briefly describe the rules, which are composed of three sets of rules and several control rules over them. For each word, (a part of) three accentual attributes of concatenation manner (CM), nucleus position (NP), and concatenation type (CT) have to be prepared.

2.2.1. *Concatenation of a content word and a function word to form an accentual phrase*

Suppose that the concatenation of a content word of N_1 morae and type-M_1 accent and a function word (an auxiliary verb or a particle) of N_2 morae and NP being $\widetilde{M_2}$ produces an accentual phrase of N_c morae and type-M_c accent. NP is an attribute indicating the accent nucleus position in the produced accentual phrase. If the resulting accent nucleus is located as the last mora of the first word in the phrase, NP is zero. If the first mora of the second word is the accent nucleus, NP is one. It should be noted that NP can take a negative value.

If every word which can appear as the second word has its own value of NP, CM is not needed. This is because, as told above, the location of the accent nucleus is determined only by NP. In some cases, however, the accent nucleus of the first word remains after the concatenation. In these cases, the nucleus position of the phrase cannot be predicted only by the accentual attributes of the second function word. To sum up, it can be said that the accent nucleus position of an accentual phrase composed by a content word and a function word is determined by the length and the accent type of the first word and CM and NP of the second word. Table 1-(a) shows these word accent sandhi rules. As shown in the table, all of the factors above are not always required to determine the nucleus location in the phrase.

Table 1: Word accent sandhi rules of Japanese
word of N_1 morae and type-M_1 accent +
word of N_2 morae and nucleus position (NP) being $\widetilde{M_2}$
→ accentual phrase of N_c morae and type-M_c accent

(a) Concatenation of a content word and a function word

concatenation manner	M_c	
	M_1=0	$M_1 \neq 0$
(F1) 従属型*	M_1	
(F2) 不完全支配型*	$N_1+\widetilde{M_2}$	M_1
(F3) 融合型*	M_1	$N_1+\widetilde{M_2}$
(F4) 支配型*	$N_1+\widetilde{M_2}$	
(F5) 平板化型*	0	

(b) Concatenation of a content word and a noun

concatenation type	conditions of the 2nd word	M_c
(C1) 保存型*	$N_2 \geq 2, M_2 \neq 0, N_2{}^{\dagger}$	$N_1 + M_2$
(C2) 生起型*	$N_2 \geq 2, M_2 = 0, N_2{}^{\dagger}$	$N_1 + 1$
(C3) 標準型*	$N_2 \leq 2$	N_1
(C4) 平板型*	$N_2 \leq 2$	0

(c) Concatenation of a prefix and a content word

concatenation type	M_c	
	M_2=0, $N_2{}^{\dagger}$	$M_2 \neq 0$, $N_2{}^{\dagger}$
(P1) 一体化型*	0	N_1+M_2
(P2) 自立語結合型*	N_1+1	N_1+M_2
(P3) 分離型*	M_1	M_1 (and N_1+M_2)
(P4) 混合型*	N_1+1 (or)M_1	M_1 (and/or) N_1+M_2

$*$: In Sagisaka's original paper in Japanese, as shown here, each value of CM and CT has a meaningful name, not a label. Due to limited space, however, these values are referred to by the labels of Fx, Cx, and Px in this paper.

† : If the final syllable of the second word is comprised of two morae, N_2 should be decremented by one.

2.2.2. *Concatenation of two content words*

Word accent sandhi observed when concatenating two content words can be characterized by adequately setting the CM and NP values of the second *content* word. It means that these values have to be prepared for every content word. But when the second word is a verb or an adjective, the accent nucleus of the resulting phrase is always found as the last mora but one in the phrase (M_c=N_1+N_2−1). This property of Japanese requires that the values of CM and NP should be prepared only for the nouns which can occur as the second word. In this case, unlike function words described in the previous section, the CM value of the second noun word is always F4 or F5. Then, the NP value has only to be prepared for the noun word. Tab. 1-(b) shows the word accent sandhi rules in concatenating a content word and a noun. Although concatenation types (CT) are newly defined in the table, they are functionally the same as NP. C1 to C4 correspond to the NP values of M_2, 1, 0, $-N_1$ respectively. As the NP values of nouns of three morae or longer can be automatically calculated by their length and accent types, only the nouns of two morae or shorter should be considered.

2.2.3. *Concatenation of a prefix and a content word*

To make an accentual phrase by attaching a suffix to a content word, the rules in Section 2.2.2 can be basically applied as they are. For a phrase composed by a prefix and a content word, new rules should be prepared, which are shown in Table 1-(c). It should be noted that, for P3 and P4, semantic analysis is sometimes required to adequately locate the accent nucleus.

In addition to the above rules, several control rules have to referred to when the above rules are used in a TTS system. Due to the limit of space, the control rules are not shown here.

3. Assignment of accent labels to a text corpus by a single labeler

In our previous study [1], the accentual attributes required by the accent sandhi rules were estimated experimentally and they were used in some TTS system developments [5]. However, covering all the accent sandhi phenomena by rules is very difficult. In the rules, only the locations of primary accent nuclei were considered with the problem of secondary accent nuclei unsolved. Further, the sentences including function word concatenation were not adequately treated, either. To solve these problems, a corpus-based approach has been adopted recently. This new approach, however, naturally requires a text corpus with accurate accent labeling but it does not exist publicly.

In the current section, the development of a text corpus with accent labeling is described in detail and its actual use in the TTS system will be shown in the following section.

3.1. What kind of labeling should be done?

If one tries to build a rule-based module to predict the accent change, for each word, he has to prepare the values of the accentual attributes described in the previous section. In the corpus-based prediction of the accent change, the values of these rather-complicated accentual attributes are not required explicitly. For example in [6], n-gram models were used to develop a morphological analyzer which can predict the H/L attribute for each mora of an input sentence. In this work, the accentual attributes of the previous section were not used at all. For training the n-gram models, only the lexical attributes, often used in the text analyzer, were referred to in addition to the H/L values of each mora of the training sentences. It should be noted that the H/L values of each mora of the constituent words when they are uttered isolatedly were not used in [6]. Even with this strategy, the prediction performance was shown to be very high. In the current paper, another statistical and machine-learning method was adopted, which is Conditional Random Fields (CRFs) [7].

CRFs are a probabilistic framework for labeling and segmenting structured data, such as sequences, trees and lattices. The underlying idea is that of defining a conditional probability distribution over label sequences given a particular observation sequence, rather than a joint distribution over both label and observation sequences. Also in the case of using CRFs, the values of the accentual attributes discussed in the previous section are not needed. In the current study, the following three labels were added manually to some existing text corpora.

1. Location of the accentual phrase boundary
   A sentence utterance can be divided into several segments according to the global F_0 movement. At the beginning of each segment, F_0 rises and then, it gradually falls without a F_0 rise in the segment. The mora with the F_0 rise is the first mora of an accentual phrase and all the phrase boundaries were manually annotated. As the boundary location depends on speaking rate, the annotation was done so that a labeler could assign the boundary by looking at the reading rate indicator (See Fig. 2). The labeler was asked to read a given sentence silently according to the indicator before the assignment.

2. Location of the accent nucleus in every accentual phrase
   In an accentual phrase, according to the lexical attribute of the constituent words, one or sometimes plural rapid F_0 downfalls are observed. The mora immediately before the downfall is called accent nucleus. If plural F_0 downfalls are found in a phrase, it is considered that the first one is primary and the others are secondary accents.

3. Location of the accent nucleus in every content word when uttered isolatedly
   In this work, unlike [6], to predict the accent change, the nucleus location of each content word when uttered isolatedly was considered. The labeler was asked to indicate the nucleus position of every content word.

3.2. Selection of the single labeler

Two speakers even of the same dialect sometimes claim different accent nucleus positions for the same sentence. As phonological knowledge such as accent sandhi rules is implicit and, exactly speaking, is considered to be speaker-dependent, we decided to ask a single labeler to assign the above three labels to the whole text corpus by reading each sentence silently. As told in Sect. 2, the word accent in Japanese is mainly controlled by F_0. Then, at first, we selected 6 university students who had a good ear for the height of tone. They were members of chorus clubs and born and brought up in Tokyo. After teaching them Japanese phonology and the accent sandhi rules, we examined how sensitive they could be to linguistic sounds. In other words, we examined how well they could explicitly describe what they had in their brains implicitly. Finally, we selected a single student as labeler and asked her to assign the three kinds of labels.

As will be told in the following section, the total number of the sentences which the labeler had to deal with was more than 15 thousands. Due to the large size of the task, annotation errors may be unavoidable. Then, out of the remaining five students, we selected a few examiners, who were asked to check all the annotations. If they found some strange labels, these were fed back to the labeler, who evaluated these labels again.

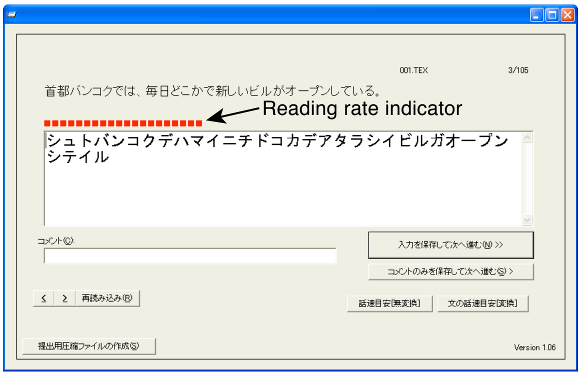
3.3. Selection of the text corpus

The sentences used in the Japanese Newspaper Article Sentence database (JNAS) [8] were adopted as the text corpus. The sentences can be divided into two parts, 16,178 sentences from The Mainichi Newspapers and 503 from ATR phoneme-balanced sentences. The reasons for selecting JNAS were that all the sentences had been assigned their phonographic representation[2] and that a speech corpus for all the sentences already existed. Since the speech corpus is composed of 306 speakers, each reading a part of the corpus, it is not adequate to ask the labeler to determine the accent nucleus positions by hearing them. Further, she claimed that it was easier by reading than hearing.

3.4. Morphological analysis done on the text corpus

Every kind of content word in JNAS was separately assigned its accent nucleus position. Further, in developing a module to predict the accent sandhi using CRFs, many lexical and phonological attributes of every word of the JNAS sentences are needed. Then, morphological analysis was done on the whole sentences. Chasen [9] and UniDic [10] was adopted as morphological analyzer and dictionary. As for part-of-speech (POS), UniDic-based POS was used. The combination of Chasen and UniDic can automatically generate the phonographic representation of

[2] Japanese has two types of writing systems, phonographic (Kana) and ideographic (Kanji) systems. The sentences in newspapers are usually represented using the both systems and it is sometimes difficult to automatically determine how to convert the ideographic part into its phonographic representation.



首都バンコクでは、毎日どこかで新しいビルがオープンしている。

シュトバンコクデハマイニチドコカデアタラシイビルガオープン
シテイル

Figure 2: GUI for labeling the JNAS corpus

an input text and they showed how to read the individual sentences in JNAS. A small part of the outputs were different from the phonographic representations prepared in Sect. 3.3. For uniformity, these mismatches were manually fixed. When assigning the labels, in the case that the labeler pointed out some strange phonographic symbols of a given sentence, we gave the highest priority to the judgment of the labeler and adopted it.

3.5. Procedure of the actual accent labeling

As described in Sect. 3.1, the labeler was asked to read a given sentence silently according to the reading rate indicator (See Fig. 2). The indicator shows the rate of 7 [morae/sec] because this value is widely accepted in developing TTS systems. After reading, the labeler determined the locations of the phrase boundaries and those of the accent nuclei. As for assigning the accent nucleus position separately for each content word, a dummy word was added if necessary to follow the focused word. As told in Sect. 2.1, type-0 and type-n words are not discriminable if they are presented isolatedly. To avoid this confusion, we asked the labeler to add particle “ガ” after the given word when it was a noun and to add noun word “コト” when the given word was an adjective. The followings are examples.

首都バンコクでは，毎日どこかで新しいビルがオープンしている。
シュトバンコクデハマイニチドコカデアタラシイビルガオープンシテイル

becomes

シュ’ト／バ’ンコクデ’ハ／マ’イニチ／ド’コカデ／アタラシ’イ／ビ’ルガ／オ’ープン／シテイル.

As for the separate labeling,

シュト (ガ)
アタラシイ (コト)

becomes

シュ’ト (ガ)
アタラシ’イ (コト) .

“／” means the position of the accent phrase boundary and “ ’ ” indicates that of the accent nucleus.

3.6. Discussions

As of the end of March 2007, the accent labeling of 4,166 sentences, about a fourth of the corpus, were completed and the rest of the sentences will be dealt with later. Tab. 2 shows the number of morphemes in an accentual phrase. It is found that the phrases whose word-based length is less than 5 occupy more than 90% of all the phrases. Tab. 3 shows the number of POS patterns in all the phrases, where the top 10 frequent patterns are listed. From this table, we can say that the phrases below the top 10 occupy about a half of the phrases. In the following sections, using the corpus built so far, CRF-based statistical learning is investigated to predict the word accent sandhi.

Table 2: The number of morphemes in an accentual phrase

#morphemes	#occurrences	
1	5,079	(17.4%)
2	9,829	(33.6%)
3	7,902	(27.0%)
4	3,972	(13.6%)
5	1,586	(5.4%)
6	554	(1.9%)
>6	303	(1.0%)

Table 3: The number of POS patterns in the accentual phrases

POS pattern	POS pattern	#occurrences	
[名][助]	[N][P]	5,273	(18.0%)
[名]	[N]	2,639	(9.0%)
[名][名][助]	[N][N][P]	2,180	(7.5%)
[名][接尾][助]	[N][S][P]	1,409	(4.8%)
[動][助動]	[V][AV]	792	(2.7%)
[動]	[V]	788	(2.7%)
[名][名]	[N][N]	758	(2.6%)
[名][接尾]	[N][S]	739	(2.5%)
[動][助]	[V][P]	571	(2.0%)
[名][助][助]	[N][P][P]	541	(1.9%)
others		13,535	(46.3%)

名:Noun, 助:Particle, 接尾:Suffix,
動:Verb, 助動:Auxiliary Verb

4. CRF-based statistical learning of the word accent sandhi

4.1. Conditional Random Fields (CRFs)

CRFs are a probabilistic framework and it defines a conditional probability distribution over label sequences given a particular observation sequence, rather than a joint distribution over both label and observation sequences. In CRFs, conditional probability $P(\boldsymbol{y}|\boldsymbol{x})$, where $\boldsymbol{y}$ and $\boldsymbol{x}$ are random variables for label and observation, is trained in the following way. Here, independent features fs are prepared about the temporal transition from y_t to y_{t+1}, called transition feature, and the generative relation between y_t and x_t, called observation feature. Let θ_f be the degree of importance of feature f and $\phi_f(\boldsymbol{x}, \boldsymbol{y})$ be the frequency of feature f being observed in the training data. Using these parameters, $P(\boldsymbol{y}|\boldsymbol{x})$ is modeled as

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp \sum_f \theta_f \phi_f(\boldsymbol{x}, \boldsymbol{y})}{\sum_{\boldsymbol{y} \in Y} \left\{ \exp \sum_f \theta_f \phi_f(\boldsymbol{x}, \boldsymbol{y}) \right\}}.$$

In the training, θ_f is optimized to maximize $P(\boldsymbol{y}|\boldsymbol{x})$ for the training data. In this paper, CRF++ toolkit [11] was utilized.

4.2. What to learn with what?

In the text corpus with accent labeling, the positions of the accentual phrase boundaries and those of the accent nuclei are annotated. All the sentences are divided into accentual phrases and, on each phrase, the accent type of the constituent words is learned as $\boldsymbol{y}$ using their various lexical and phonological attributes as $\boldsymbol{x}$. It should be noted that there is a big difference between our previous study and the current study in interpreting the resulting accentual property of each phrase generated by concatenating some words. For example, オンセーゴーセー (speech synthesis) is generated by concatenating two separate words of オンセー (speech) and ゴーセー (synthesis).

オンセー (1) + ゴーセー(0) → オンセーゴーセー (5)

In the previous study, this accent sandhi was interpreted as follows. A type-1 word and a type-0 word are concatenated to form a long compound word of type-5. In the current study, however, it is interpreted as follows.

オンセー (1) + ゴーセー(0) → オンセー(0) + ゴーセー (1)

Through the concatenation, a type-1 word and a type-0 word are transformed into type-0 and type-1. Considering the function of the word accent, this interpretation seems weird linguistically because the accent sandhi is considered to function as grouping plural words into one entity. We can say that the CRF-based statistical learning of the accent sandhi only captures the superficial transformation of the accent type of the individual words through the concatenation. The validity of this kind of approach should be carefully investigated but this paper aims to report only the performance of CRFs to predict the accent sandhi.

3,581 sentences (25,692 accentual phrases) were used to train the CRF models and the remaining 527 sentences (3,533 accentual phrases) were used to test the models.

4.3. CRF-based learning as step 1

As the first step, CRFs were examined without using the accent type of the constituent words when they are uttered isolatedly. This condition is the same as that in [6]. In the rest of the paper, the accent type of the word when uttered isolatedly is called *isolated* accent type and the accent type observed when the word is embedded in a phrase is referred to as *embedded* accent type. As observation feature, each of the followings was considered as $\boldsymbol{x}$; POS, inflection types, and the mora-based length of w_{t-2}, w_{t-1}, w_t, w_{t+1} and w_{t+2}. The POS and the inflection types of UniDic are defined using multiple granularity. Following this definition, various kinds of POS and inflection types were investigated. As for transition feature, the embedded accent type of any of two consecutive words of w_{t-1} and w_t was considered. The performance of CRFs is calculated in three different ways, shown in Tab. 4. The prediction performance for all the accentual phrases, that for the phrases comprised of only two words as (noun|verb|adjective)+(auxiliary verb|particle), called as *simple* phrases henceforth, and that for the phrases including a compound noun word comprised of several consecutive nouns, called as *compound* phrases. The morpheme-based performance is also calculated for reference. Tab. 4 shows that the overall performance is 82.1%. Although it improves up to 85.9% for the simple phrases, it reduces down to 77.9% for the compound phrases[3]. It is more difficult to predict the nucleus position correctly in the compound phrases.

4.4. CRF-based learning as step 2

In addition to the observation features used above, the isolated accent type was also used here. The three kinds of performance are shown in Tab. 4. Although the overall performance and the simple performance are much improved, with the compound phrases, a slight reduction is observed. In the simple phrases, in many cases, the accent nucleus position is unchanged through concatenating the two words. On the other hand, in the compound phrases, the nucleus position is often changed through concatenating two nouns to form a compound noun. Examples are shown in Sect. 2.2 (アカエンピツ and ケイタイデンワ).

4.5. CRF-based learning as step 3

In the above experiments, the embedded accent nucleus position was directly predicted. Therefore, the following two cases were separately handled and modeled. A case that both the isolated nucleus and the embedded nucleus are located at the first mora and another case that they are located at the second mora. These two cases can be commonly and simply treated as "not changed" if the *relative* change of the nucleus position from isolated to embedded is predicted. In the current section, the target of the prediction was set to the relative change in the nucleus position and the following labels were prepared.

When both the isolated accent and the embedded accent had the nucleus, the labels of [0],[+1],[+2],...,[-1],[-2],...were prepared to represent the relative change of the nucleus position. When the embedded accent did not have the nucleus, the label of [none] was prepared and CRFs were trained to predict that label. When the isolated accent did not have the nucleus, the nucleus position was directly predicted as in the last section.

The performance of the relative change prediction is also shown in Tab. 4. In all the cases of all, simple, and compound, the performance is successfully increased. Especially, the increase is larger in the compound phrases. By introducing the relative change prediction, it seems that what was difficult to predict in the previous section can be adequately handled.

4.6. CRF-based learning as step 4

In this section, the training of CRFs is tuned to the accent sandhi rules described in Sect. 2. In the experiments so far, as observation feature, the generative relation between label y and lexical or phonological attribute x of observed word w was used. Referring to the accent sandhi rules, however, some kinds of the relation should be additionally considered such as that between y and x of some plural words, w_t and w_{t+1}, for example. As described in the Sect. 4.3, various kinds of the syntactic categories with multiple granularity were provided by UniDic. By carefully observing the accent sandhi rules, we prepared some word combinations to fit the CRF training to the rules. The followings are examples. [POS of w_t/POS of w_{t-1}], [POS of w_t/POS of w_{t+1}], and [fundamental lexical attributes of w_t/POS of w_{t+1}]. The fundamental lexical attributes are a set of the attributes selected adequately from the whole set of syntactic categories provided by UniDic. They included POS, inflection types, phonographic and logographic representations, and so forth.

The performance is shown in Tab. 4, again. Although a very slight reduction is found in the simple phrases, the overall performance is improved. Especially, as in the previous section, the increase is larger in the compound phrases. As described in Sect. 4.3, the nucleus prediction is more difficult in the phrases including compound nouns. We consider that the complicated phonological phenomena could be modeled by CRFs better by means of additionally introducing a set of rather complicated word combinations as better observation features.

4.7. CRF-based learning as step 5

Some additional tuning to the accent sandhi rules was investigated. In Sect. 4.5, the labels were prepared to indicate the

[3] For the phrases where plural nucleus positions are annotated, only the first nucleus is considered because it is the primary accent nucleus.

Table 4: The performance of the CRF-based statistical learning of the Japanese word accent sandhi

	morpheme-based	phrase-based		
		all	simple	compound
Step 1	9080 /9908 (91.6%)	2833 /3533 (82.1%)	703 /822 (85.9%)	530 /688 (77.0%)
Step 2	9272 /9908 (93.6%)	3081 /3533 (87.2%)	775 /822 (94.3%)	523 /688 (76.0%)
Step 3	9319 /9908 (94.1%)	3137 /3533 (88.8%)	791 /822 (96.2%)	553 /688 (80.4%)
Step 4	9424 /9908 (95.1%)	3214 /3533 (91.0%)	790 /822 (96.1%)	578 /688 (84.0%)
Step 5	9458 /9908 (95.5%)	3238 /3533 (91.7%)	792 /822 (96.4%)	589 /688 (85.6%)

Table 5: The performance of the CRF-based statistical learning based on the labeler's judgment

	phrase-based		
	all	simple	compound
Step 5	3307 /3533 (93.6%)	808 /822 (98.3%)	605 /688 (87.9%)

relative change of the nucleus position. In this section, the categories of these relative labels were modified to fit the module to the rules much better. As the detailed description of the modification may be tedious to readers, we show only some examples.

When the isolated accent had the nucleus, the following labels were prepared. 1) When the embedded accent did not have the nucleus, the label was [none], 2) When the embedded accent was the same as the isolated accent, the label was [same], 3) When the embedded accent nucleus was located at the last mora of the word, the label was [morae], 4) When the embedded accent type was smaller than the isolated type by 1, the label was [same-1], 5) When the embedded accent type was 1, the label was [one], 6) When the embedded accent nucleus was located at the last mora but one in the word, the label was [morae-1], 7) When the embedded accent did not correspond to any case from 1) to 6), the labels of the relative change such as [0], [+2], and [-1] were used. To assign a label to w_t, it was possible that the word could satisfy plural conditions and, in this case, the condition of the smallest number was applied. In other words, the above conditions were examined in the incremental order because the order reflected the frequency of the labels. In the last condition, the relative change labels were assigned. Before that, the labels introduced newly in this section were given to w_t. Some good readers may wonder why these cases should be treated as special cases. All of these cases were directly treated by the accent sandhi rules and, in this section, only the exceptional cases were treated by the relative change labels.

Some other tuning was also done with respect to observation features including the phonographic representation of the second mora of w_t, that of the mora located at the isolated accent nucleus position, and so forth. These features were required to fully implement the accent sandhi rules in Sect. 2 as observation features in the CRF training.

The performance is shown in Tab. 4 and only the small improvements are observed in all the three cases.

5. Discussions and conclusions

In the previous section, only the nucleus positions which the single labeler had provided were treated as correct. It is true, however, that the labeler did not reject all the other positions. We asked the labeler to judge the degree of acceptance of the accent nucleus positions predicted *incorrectly* by the CRF module. The judgment was done by using a 4-degree scale. If the nucleus positions of acceptance level 3 or 4 are re-considered as correct, the final performance is shown in Tab. 5. 93.6% of all the phrases showed the correct position and, in the simple phrases, the performance reached 98.3%. Considering that the performance of the rule-based module for the same testing data is 76.8% and 94.5% respectively for the two cases, we can claim strongly that the proposed method showed the remarkably better performance and it is very effective practically. As discussed in Sect. 4.2, however, the CRF-based implementation of the accent nucleus prediction is somewhat weird linguistically. At least, the proposed module can predict the accent change but it does not know the linguistic function of the change at all. We may have to reconsider how to train CRFs for this task.

6. References

[1] N. Minematsu, R. Kita, and K. Hirose, "Automatic estimation of accentual attribute values of words for accent sandhi rules of Japanese text-to-speech conversion," Trans. IEICE, vol.E86-D, no.3, pp.550–557 (2003)

[2] "Word accent dictionary of Japanese pronunciation," published by NHK (Nippon Hoso Kyokai) (1998, in Japanese).

[3] Y. Sagisaka, and H. Sato, "Accentuation rules for Japanese text-to-speech conversion," Review of the Electrical Communication Laboratories, vol.32, no.2, pp.188-199 (1984).

[4] Y. Sagisaka, and H. Sato, "Accentuation rules for Japanese word concatenation," Trans. IECE Jpn., vol.66D, no.7, pp.849–856 (1983, in Japanese).

[5] S. Kawamoto, *et al.*, "Galatea: open-source software for developing anthropomorphic spoken dialogue agents," in *Life-like Characters, Tools, Affective Functions, and Applications*, Helmut Prendinger *et al.* (Eds.), Springer, pp.187–212 (2003)

[6] T. Nagano, S. Mori, and M. Nishimura, "An N-gram-based approach to phoneme and accent estimation for TTS," Trans. IPS Japan, vol.47, no.6, pp.1793–1801 (2006)

[7] J. Lafferty, A. McCallum, F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," Proc. Int. Conf. Machine Learning, pp.282–289 (2001)

[8] JNAS: Japanese Newspaper Article Sentences,
http://www.mibel.cs.tsukuba.ac.jp/jnas

[9] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda, K. Takaoka, M. Asahara, "Japanese Morphological Analysis System: ChaSen,"
http://chasen.aist-nara.ac.jp/chasen/doc/chasen-2.2.9.pdf
http://chasen.aist-nara.ac.jp/chasen/doc/chasen-2.3.3-j.pdf

[10] Y. Den, A. Yamada, H. Ogura, H. Koiso, T. Ogiso, "Japanese Morphological Analysis Dictionary: UniDic,"
http://download.unidic.org

[11] T. Kudo, "CRF++: Yet Another CRF Toolkit,"
http://crfpp.sourceforge.net

Two-Step Generation of Mandarin F_0 Contours Based on Tone Nucleus and Superpositional Models

Qinghua Sun, Keikichi Hirose**, and Nobuaki Minematsu****

*Graduate School of Engineering, **Graduate School of Information Science and Technology,
***Graduate School of Frontier Sciences, University of Tokyo, Japan

{qinghua, hirose, mine}@gavo.t.u-tokyo.ac.jp

Abstract

A 2-step scheme was developed in our method for synthesizing sentence fundamental frequency (F_0) contours of Mandarin speech. The method is based on representing a sentence logarithmic F_0 contour as a superposition of tone components on phrase components as in the case of generation process model (F_0 model). The tone components are realized by concatenating tone nucleus F_0 patterns generated by a corpus-based method, while the phrase components are generated by rules under the F_0 model framework. In the 2-step scheme, the phrase components are first generated and their information is added to the inputs for the prediction of tone nucleus F_0 patterns. Result of listening tests on synthetic speech with the synthesized F_0 contours verified the validity of the developed scheme. For comparison, we also generated F_0 contours without decomposing them into tone and phrase components as most existing methods did. Although from the viewpoint of naturalness of synthetic speech, the result did not show clear advantage of the proposed method, from the viewpoint of flexibility the advantage came clear: by manipulating phrase components in the proposed method, a better focus control was realized.

1. Introduction

Introduction of selection-based waveform concatenation in speech synthesis largely improved quality of synthetic speech. However, there still remain problems if we view from the aspect of prosodic features. Although the control of prosodic features is an important issue in speech synthesis for any languages, it becomes quite critical for speech quality in the case of Mandarin. As it is well known, Mandarin is a typical tonal language and each syllable with the same phoneme constitution has up to four tone types, each indicating different meaning. F_0 contours of utterances should include these local tonal features in addition to the sentential intonation corresponding to syntactic/utterance structures. This situation makes F_0 movements of Mandarin sentences be more complicated than non-tonal languages like English, Japanese and so on. Therefore, control of F_0 contours (together with other prosodic features) becomes an important (and tough) issue in Mandarin speech synthesis.

Benefit of corpus-based methods over rule-based methods increases when handling complicated features. Naturally, most F_0 controls adopted in Mandarin speech synthesis are corpus-based using decision trees, neural networks, hidden Markov models, and linear regression analysis [1-3]. However, most of them predict syllable F_0 contours without explicit consideration on the F_0 movement in longer units such as word, phrase, and so on.

A better control of prosodic features for the F_0 movement in longer units in synthetic speech is possible using the F_0 model, which represents a logarithmic F_0 contour as the sum of phrase and tone components [4]. This model was already used in our corpus-based generation of F_0 contours of Japanese successfully [5]. In the method, speech corpus with F_0 model commands is necessary for training process, and was arranged efficiently using the automatic method of F_0 model command extraction from speech waveform. However, in the case of Mandarin speech, automatic extraction comes difficult because of its complicated features in F_0 movements. Although several efforts are going on, corpus-based F_0 contour generation fully based on the F_0 model is less feasible in the case of Mandarin.

These considerations led us to propose a method of F_0 contour generation for Mandarin speech synthesis, where the tone components were generated by concatenating F_0 patterns of tone nuclei, predicted by a corpus-based method, and were superposed onto the phrase components, which were generated by a rule-based scheme on the basis of F_0 model [6]. Here, "tone nucleus" is defined as a portion of syllable, which possesses a stable F_0 pattern regardless of the context [7]. By first generating F_0 patterns for tone nuclei of constituting syllables and then concatenating them, a smooth sentence F_0 contour can be generated.

The F_0 contours are considered to consist of both language specific and universal characteristics. Features for tone components may be mostly language specific, while those for phrase components may be mostly language universal, because they are tightly related to higher-level linguistic information, such as syntactic structure, discourse structure, and so on. Therefore, rules developed for other languages are somewhat applicable for the control of phrase components in Mandarin. We tried to apply the rules, which have been developed for the control of phrase components of Japanese, to Mandarin, and found out some differences in phrase components between two languages: in the case of Mandarin, phrase components occur more frequently than Japanese. This is considered to be due to the fact that, in Mandarin, tone components can have negative values (which is not the case in Japanese) and phrase components should keep a certain level to give a margin for tone components taking negative values. Taking the differences into account, rules were constructed for generating phrase components in Mandarin [8].

Although speech synthesized using generated F_0 contours sounded natural, there were often degraded sounds when phrase components and tone components were generated independently. In the case of Japanese, independent generation of components did cause no degradation [9]. However, for Mandarin, since phrase and tone components are tightly related to each other, independent generation occasionally causes "strange" F_0 movements because of mismatches between two components. To cope with this situation, we newly developed a two-step scheme, where the phrase components were generated first, and then the tone components were generated taking the features of generated phrase components into account.

The most significant benefit of the proposed method over others without decomposition is the flexibility in F_0 contour generation: by manually controlling phrase components, we can easily generate F_0 contours with different utterance structures. In Mandarin, it is claimed that a word with emphasis is usually accompanied by a new phrase component with a large magnitude. Following to this claim, an experiment was conducted whether the control of emphasis position in a sentence is possible or not, by manually changing phrase component and generating F_0 contours using the proposed method.

The rest of the paper is organized as follows. Section 2 describes the method of tone component generation. Then the generation rules for phrase are given in Section 3. The detail of using phrase component features for tone component prediction (two-step generation) is given in Section 4 together with experimental results in Section 5. Section 6 gives samples of word emphasis in Mandarin speech, and then an experiment on the word emphasis control is conducted in Section 7 to show the flexibility of the two-step scheme. Section 8 concludes the paper.

2. Generation of tone component

2.1 Tone nucleus model

In Mandarin, there are four lexical tones attachable to a syllable. They are referred to as T1, T2, T3 and T4, and are characterized by high-level, mid-rising, low-dipping, and high-falling F_0 contours, respectively. Besides the lexical tones, there is also a so-called neutral tone (T0), which does not possess its inherent shape in the F_0 contour. Its F_0 contour varies largely with the preceding and following tones.

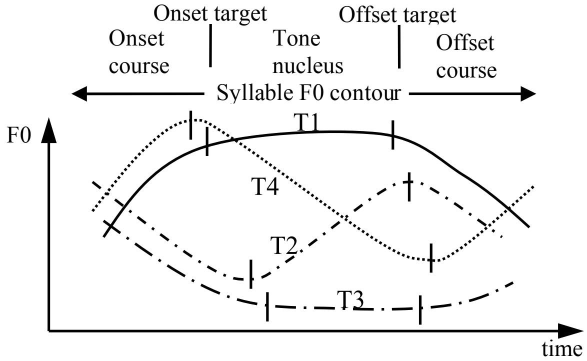


Figure 1: Tone nuclei for the four lexical tones.

For a syllable, not only its early portion but also voicing period at the ending portion is regarded as physiological transition period to/from the neighboring syllables. Based on this observation, a tone nucleus model, which divides a syllable F_0 contour into three segments according to their roles in the tone generation process, was proposed and applied to tone recognition successfully [7]. The three segments are called onset course, tone nucleus, and offset course, respectively. Only the tone nucleus is a portion where F_0 contour keeps the intrinsic pattern of the tone; the others are only the portions for physiological transitions.

Figure 1 illustrates syllable F_0 contours for the four lexical tones with possible articulatory transitions. It shows how the three segments are defined on the F_0 contours. Among the three segments, only tone nucleus is obligatory, whereas the other two segments are optional; their appearance depends on voicing characteristics of initial consonant, syllable duration, context, and etc.

2.2 Method of tone component generation

The proposed method generates a tone component for sentence F_0 contour by first predicting tone components of tone nuclei (tone nucleus components) for all the constituting syllables by a corpus-based method and then concatenating them. In the current paper, binary decision tree was used as the corpus-based method. The parameters adopted for representing the tone nucleus components are as follows:

1. For T1 and T3, tone nuclei are defined as the flat portion, which is represented by a single parameter, i.e., average F_0 value.
2. For T0, T2 and T4, tone components for nuclei are normalized both in time and pitch range, and the normalized contours are then clustered into several groups. The average contour for each group serves as a template to represent the tone nucleus component. The parameters include the absolute pitch range, average F_0 value, and template identity.

The tone components are generated through the following process:

1. For each syllable in the sentence to be synthesized, the onset and offset times of tone nucleus are predicted.
2. For each tone nucleus, parameters representing the shape of tone nucleus component are predicted.
3. Based on the predicted parameters, tone nucleus components are generated.
4. The tone nucleus components are concatenated with each other to produce the entire tone components (of the speech to be synthesized). As for the interpolation at the portions corresponding to onset and offset courses, linear one is adopted in the current experiments.

In the first and second steps above, the parameters are predicted using binary decision trees trained separately for each parameter. Inputs to a tree are the information extracted from text, such as phonemic constitutions of syllables, number of syllables in words, depths of syntactic boundaries, and so on [6]. Information on phoneme durations and pauses are also used, which may be predicted in a different process in a total text-to-speech system.

3. Generation of phrase component

In our earlier report on the generation of tone components [6], phrase components of the original utterances were used to produce sentence F_0 contours. Recently, we developed a rule-based method to generate the phrase components [8]. In the method, "prosodic word" is first defined as a chunk of syllables usually uttered in a tight connection: a prosodic word can be a word, a compound word, or a word chunk uttered together frequently. For example, the sentence shown in Figure 2 can be segmented as follows:

(yu4ji4) | (quan2nian2) | (liang2shi6)(zong2chan3liang4) | (ke3da2) | (er4shi0dian3) | (qi1wu3yi4)(gong1jin1)

Here, a pair of parentheses embraces an element (syntactic) word, while "|" indicates prosodic word boundary. Then, the following rules are constructed based on the observations of F_0 contours of 100 utterances by a female native speaker of Mandarin:

Rule 1: Place a phrase command with magnitude 0.6 at the silence at the beginning of the sentence (SilB) or after a short pause (SP) longer than 300 ms. Also, place a phrase command with magnitude 0.47 after a SP shorter than 300 ms but longer than 200 ms.

Rule 2: Check all the prosodic word boundaries without an SP in a left-to-right manner from the utterance initial. If phrasal F_0 at the current boundary falls into lower (threshold) range (set at 150Hz ~ 190Hz according to statistic), place a phrase command with magnitude as shown in Table 1, depending on the number of preceding phrase commands between preceding SilB/SP and current phrase command (counting current one).

Rule 3: During the process of rule 2, when phrasal F_0 at the current prosodic word boundary falls below the lower range, go back to the preceding boundary and place a phrase command there with magnitude shown in Table 2 depending on the feature of preceding phrase commands. If a phrase command has already been placed at the preceding boundary, or if "number of phrase commands" or "phrasal F_0" is out of the cases of Table 2, skip to rule 4.

Rule 4: Split the prosodic word before the current word boundary into two smaller prosodic words. Then go back to apply rules 2 and 3 on the newly inserted prosodic word boundary.

An additional rule is applied to the timing of phrase commands. The distance of the phrase command ahead of the corresponding prosodic boundary is set as follows: 150 ms for the phrase commands larger than 0.5, 50 ms for the commands smaller than 0.3, and 80 ms for those in between.

Table 1: Magnitude of phrase command placed at the current prosodic word boundary when phrasal F_0 falls into the lower range.

Number of phrase commands	2	3	4	5	≥6
Magnitude of phrase command	0.36	0.35	0.35	0.29	0.29

Table 2: Magnitude of phrase command placed at the preceding prosodic word boundary when phrasal F_0 falls below the lower range at the current prosodic word boundary.

F_0 at immediately preceding prosodic word boundary	190Hz~230Hz				230Hz~280Hz
Number of phrase commands	2	3	4	5	2
Magnitude of phrase command	0.32	0.28	0.28	0.26	0.29

To evaluate the generated phrase components through speech synthesis experiment, sentence F_0 contours were synthesized using tone components of original utterances, to avoid errors in tone component prediction affect the evaluation. Speech synthesis was conducted by replacing original F_0 contour by synthesized F_0 contour using TD-PSOLA scheme.

Figure 2 shows the waveform of synthesized speech for "*yu4 ji4 quan2 nian2 liang2 shi0 zong3 chan3 liang4 ke3 da2 er4 shi0 dian3 qi1 wu3 yi4 gong1 jin1* (It is estimated that the output of grain can be improved to 2.075 billion kilograms in the whole year)," together with original F_0 contour (middle) and synthesized F_0 contour (bottom). Although the difference between two contours seems minor, a hearing test indicated a considerable degradation in the synthetic speech quality.

Since the phrase components generated by the rule do show no unnatural movements (even though they are different form the original ones), the reason of the degradation is considered to be the mismatch between phrase components and tone components. Since phrase components are tightly related to the structure of utterance, which also affects tone contour range and so on, tone components need to be predicted using information of phrase components.

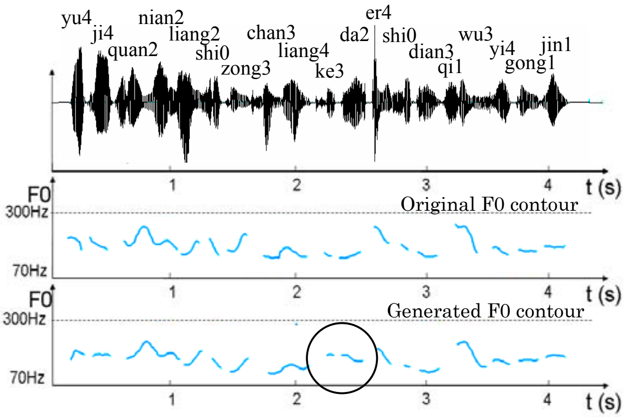


Figure 2: From top to bottom: waveform of synthesized speech, observed F_0 contour of target speech, and F_0 contour generated. The sign denotes the "strange" portion. Vertical axes of the second and third panels are frequency in logarithmic scale.

4. Two-step F_0 contour generation

To solve the problem, a two-step scheme showed in Figure3 was proposed for F_0 contour generation, where phrase components were generated firstly, and then the tone component were generated using the information of phrase components generated in the first step. To achieve the two-step generation, the information (related to phrase component) shown in Table3 is added to the inputs of predictors for tone component parameters. Henceforth, the scheme not using phrase component information is denoted as "one-step scheme."

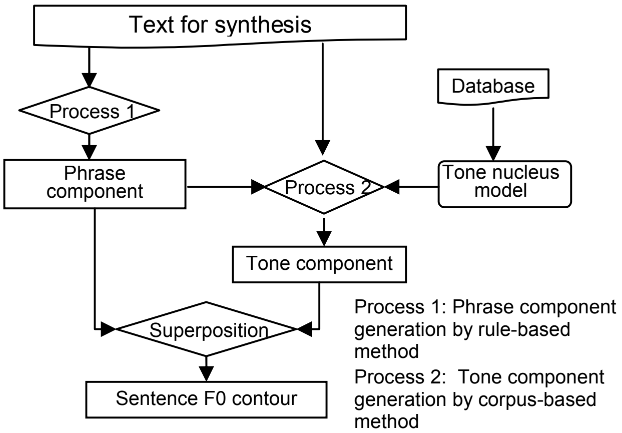


Figure 3: Two-step scheme of F_0 contour generation

Figure 4 shows the waveform of synthesized speech, together with synthesized F_0 contour by two-step scheme (middle) and by one-step scheme (bottom) for the same sentence shown in Figure 2. As clearly indicated, the F_0 contour generated by the two-step scheme is closer to the one observed in the original utterance (in Figure 2) than that generated by the one-step scheme. Also, the "strange" portion (circled portion) is corrected when the two-step scheme is used. Of course, a hearing test indicated a considerable upgrade in quality of the synthetic speech by the two-step scheme than the one-step scheme.

Table 3: Inputs added to the predictor.

Inputs added to the predictor	Category
Position of syllable in current phrase	Natural num.
Number of syllables in current phrase	Natural num.
Number of phases in current breath group	Natural num.
Position of phrase in current breath group	Natural num.
Position of breath group in sentence	Natural num.
Current phrase command magnitude	Continuous
Timing of current phrase	Continuous

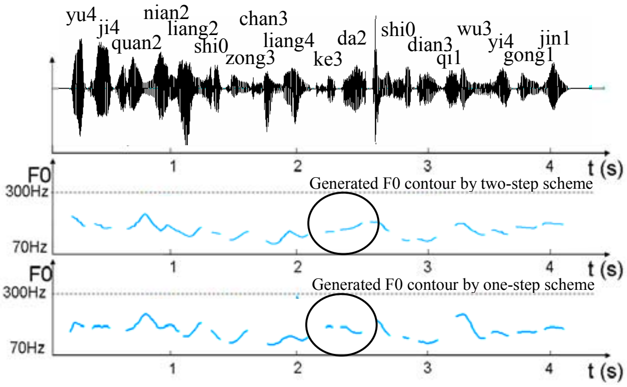


Figure 4: From top to bottom: waveform of synthesized speech, F_0 contour generated by two-step scheme and one-step scheme. The signs denote the "strange" and corrected portions.

5. Experiments on F_0 contour generation

5.1 Comparison of one-step and two-step schemes

In order to show the advantage of the two-step scheme over the one-step scheme, listening experiment is conducted for synthetic speech with F_0 contours generated by the two schemes. The 100 news utterances used in constructing rules for phrase component generation in section 3 are again used for the experiment. Each utterance consists of about 50 syllables, totally 4839 syllables. First, all the F_0 contours were manually decomposed into tone and phrase components. Then, tone nucleus was searched for each syllable. For T2 and T4, a nucleus can be detected rather easily by searching for peaks and valleys of F_0 contours. On the other hand, it is rather difficult to automatically find the flat F_0 portion for T1 and T3. Therefore their tone nuclei were manually extracted. 4389 syllables are used to train binary decision trees for predicting tone component parameters.

Out of 100 utterances, we selected 9 consisting only of syllables not used in the training. Speech synthesis (TD-PSOLA) was then conducted by substituting the original F_0 contours to the generated F_0 contours.

Five native speakers of Mandarin were asked to evaluate the quality of synthetic speech with a focus on prosody, using a five-point scoring: 5 (excellent), 4 (good), 3 (acceptable), 2 (poor), and 1 (very poor). Speech stimuli were presented in a random order. The result is shown for each listener and as an average in Figure 5. In the right-hand-side square, the letters "o" and "r" before "-" depict the original and rule-generated phrase components, respectively. The letters "gt" and "go" after "-" depict generated tone components by the two-step scheme, and those by the one-step scheme.

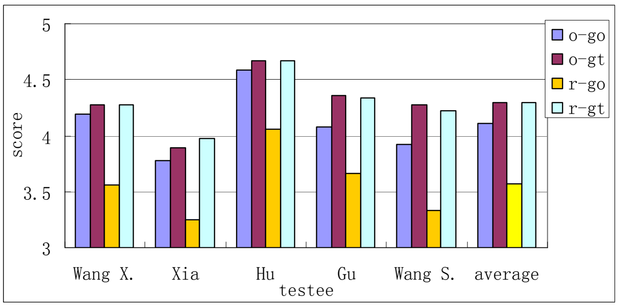


Figure 5: Results of listening test for various combinations of phrase and tone components.

When the scores for "r-go" and "r-gt" are compared, it is clear a sharp improvement in naturalness is possible by the two-step scheme. It should be noted, even when the original phrase components are used, a better result is obtained by introducing the two-step scheme. To assure the advantage of the two-step scheme over the one-step scheme being not sentence dependent, evaluation was further conducted for 30 sentences not included in the training data using 3 of the five speakers. Results are quite similar to those shown in Figure 5. From these results, we can say that the information of phrase components can help a lot to improve the accuracy of tone component prediction, especially when the phrase component is different from that observed in target speech. In other words, the two-step scheme can avoid "strange" F_0 movements for a variety of phrase components.

5.2 Method without decomposition

For comparison, F_0 contours were also generated without decomposing them into phrase and tone components; a sentence F_0 contour was generated as concatenation of F_0 contours (not tone components) of tone nuclei. These tone nucleus F_0 contours are predicted using binary decision trees with the same inputs for the one-step scheme. (The method uses tone nucleus F_0 contours instead of tone components for training the predictors.) Henceforth, the proposed method (with two-step scheme) is denoted as Method 1, while the method without decomposition is denoted as Method 2.

F_0 contours were generated for 30 sentences, which were not included in the training corpus. Then, TD-PSOLA-based speech synthesis was conducted by substituting the original F_0 contours to the generated ones.

Quality of synthetic speech was evaluated with a focus on prosody, using the same five-point scoring scheme. We used 60 synthetic utterances for the listening test: 30 using F_0 contours by Method 1 and 30 by Method 2. These utterances were presented in a random order to four native speakers. The scores for each speaker are shown in Figure 6. The result showed that the synthetic speech using the F_0 contours generated by the method 2 sounded more natural than that using those by method 1, though the differences are quite small. Although this result may obscure the advantage of the Method 1 over Method 2, the major merit of method 1 is its ability for "flexible" control of prosody. An experiment on the control of word emphasis was further carried out to prove the "flexibility" of the two-step scheme.

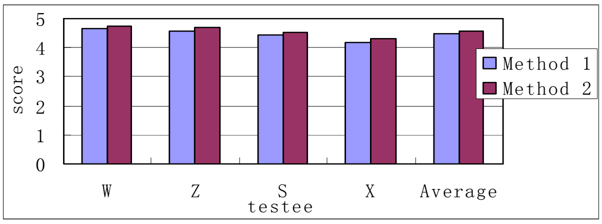


Figure 6: Results of evaluation. (Average scores of Method 1 and Method 2 are 4.47 and 4.56, respectively.)

6. Word Emphasis

Although word emphasis is not handled explicitly in most of current speech synthesis systems, its control becomes important in many situations, such as when the systems are used for generating reply speech in spoken dialogue systems: words conveying key information to the user's question need to be emphasized. Word emphasis associated with narrow focus in speech can be achieved by contrasting the F_0's of the word(s) to be focused from those of neighboring words. In the case of Japanese, this contrast is mostly realized by increasing the amplitudes of the accent commands corresponding to the word(s) to be emphasized and/or by reducing those corresponding to the neighboring words [10]. On the contrary, it is reported that the main effect of word emphasis is not on tone command but on phrase command in Cantonese, which is a major dialect of Chinese and is known as tone language with nine tones [11]. In the case of Mandarin, it is necessary to investigate how phrase and tone components will change when words are emphasized.

For this purpose, we selected 6 sentences and asked a male speaker of Chinese to pronounce by placing 4 degrees of focus on selected words. Several words were selected for a sentence, causing 18 sentences when the focal position is counted. The 4 focus levels are; without, low, middle, and high emphases. The speaker uttered twice, and F_0 contours of resulting 144 (18*4*2) utterances were investigated. The analysis result indicates a feature similar to Cantonese: a phrase command being placed immediately preceding to the word(s) to be emphasized, with larger magnitudes for higher focus levels. However, larger amplitudes of the tone commands are also associated with the word(s) with emphasis in the case of Mandarin.

Figures 7, 8 and 9 show F_0 contours (in gray lines) of various utterances for "jin1 tian1 bang4 wan3 ke3 yi3 xi3 hao3 ((*He can*) complete washing by evening)," together with their phrase components (in dark lines). If we compare the one without emphasis (in Figure 7) to those with (in Figures 8 and 9), it is clear that a phrase command is added immediately before the target word. It is also observable that when the focus level becomes higher, both of corresponding phrase and tone components increase, and, in turn, other phrase and tone components tend to shrink.

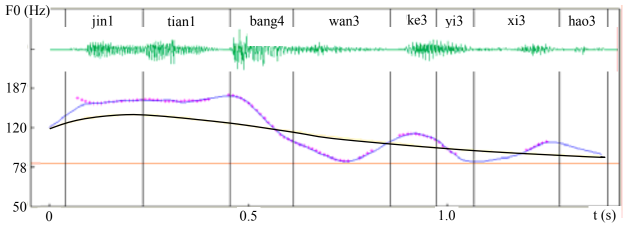


Figure 7: F_0 contour and phrase component for the utterance without emphasis.

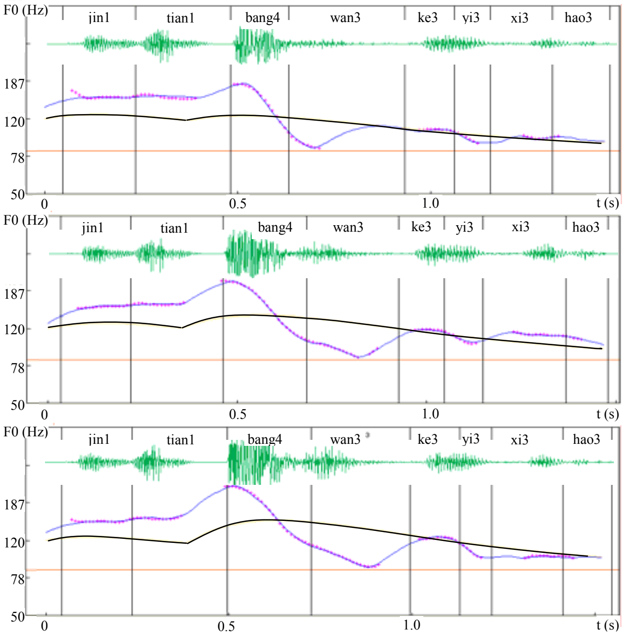


Figure 8: F_0 contour and phrase component for the utterances with three focus levels on the word "bang4 wan3." (From top to bottom: low-, middle-, and high-emphases.)

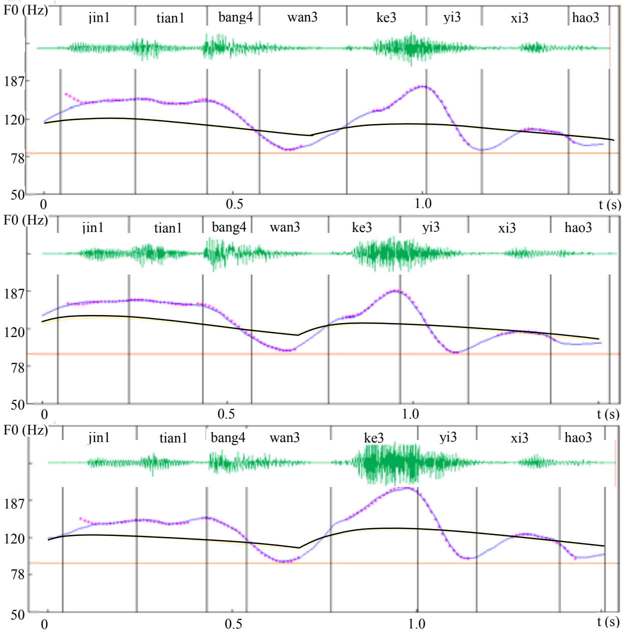


Figure 9: F_0 contour and phrase component for the utterances with three focus levels on the word "ke3 yi3." (From top to bottom: low-, middle-, and high-emphases.)

Considering the primary role of phrase component in realizing word emphasis, we tried to realize word emphasis in synthetic speech by adding a phrase command immediately before the word to be emphasized. By adopting the two-step scheme, a larger tone command for the emphasized word is also realized as shown in the next section.

7. Experiment on word emphasis

Ten sentences were selected randomly from the 100 utterances used in section 5. These sentences do not include syllables used to train binary decision trees of tone nucleus parameter prediction. For each sentence, focuses were placed on one of 3 words preselected. A phrase command was inserted immediately before the word to be emphasized. After generating other phrase commands by rule, tone commands were predicted by the two-step scheme. By doing so, 3 different F_0 contours were generated for a sentence. TD-PSOLA type speech synthesis was then conducted by substituting the original F_0 contours to the generated ones. Totally, 30 test utterances were synthesized. For the phone durations, we used the original ones extracted from the target speech: control of the duration was left for the future study.

These 30 synthetic utterances were randomly presented to four native speakers of Chinese, who were asked to mark the word where he/she perceived an emphasis. The marked parts coincided with the original emphasis assignment in more than 80 % on average. This result indicates that an appropriate emphasis control is achieved. Quality of the synthetic speech was also checked in the same way (in 5-rank scoring) as explained in section 4. The result in Table 4 again confirms that a good quality is obtainable by the two-step scheme. If we compare F_0 contours shown in Figure 10, it is clear that tone components are generated differently for different phrase components.

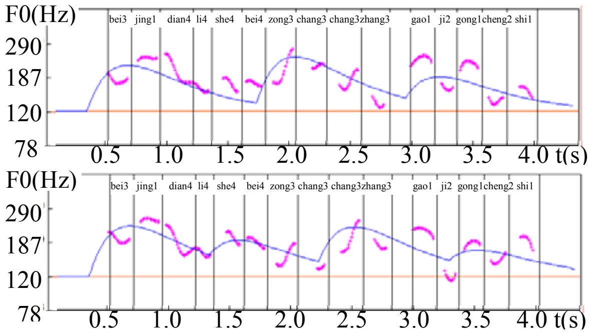


Figure 10: Two generated F_0 contours of Chinese sentence "bei3 jing1 dian4 li4 she4 bei4 zong3 chang3 chang3 zhang3, gao1 ji2 gong1 cheng2 shi1. ((*He is*) the director of Beijing Power Equipment Group and senior engineer.)" The first and the second panels show when "zhong2 chang2" and "chang2 zhang3" are emphasized, respectively. Stars indicate generated F_0 contours, while solid curves indicate phrase components.

Table 4: Results of listening test.

Testee	W	Z	S	X	Average
Correctness of focus position	86.7%	83.3%	80.0%	76.7%	81.6%
Score of evaluation	4.3	4.77	4.42	4.31	4.45

Surely, more precise control of F_0 contours can be realized for word emphasis by training the binary decision trees using corpus with word emphasis. However, we should note that focus control in this section is realized without such a corpus. This comes from the ability of "flexible" F_0 contour control of the proposed method with the two-step scheme.

8. Conclusion

A new scheme with 2-step F_0 contour generation was proposed in our F_0 contour synthesis method of Mandarin speech. The validity of using phrase component information for tone nucleus component prediction was clearly shown. Experiments on Mandarin speech synthesis were conducted. The result showed that, by using the 2-step scheme, an empirical control of word emphasis is possible keeping a good quality in synthetic speech. Future research includes realization of various styles in synthetic speech by the proposed method.

9. Acknowledgement

The authors' sincere thanks are due to Prof. Renhua Wang in the University of Science and Technology of China for his providing us the Mandarin speech corpus.

10. References

[1] S. Chen, S. Hwang, and Y. Wang, "An RNN-base prosodic information synthesizer for Mandarin Text-to-speech," *IEEE Trans. on Speech and Audio Processing*, Vol.6, No.3, pp.226-239, 1998.

[2] J. Tao, and L. Cai, "Clustering and feature learning based F_0 prediction for Chinese speech synthesis," *Proc. ICSLP*, pp.2097-200, 2002.

[3] J. Ni, and K. Hirose, "Synthesis of fundamental frequency contours of standard Chinese sentences from tone sandhi and focus conditions," *Proc. ICSLP*, pp.195-198, 2000.

[4] K. Hirose, H. Lei, and H. Fujisaki, "Analysis and formulation of prosodic features of speech in standard Chinese based on a model of generating fundamental frequency contours," *J. Acoust. Soc. Japan*, Vol.50, No.3, pp.177-187, 1994.

[5] K. Hirose, and H. Fujisaki, "A system for the synthesis of high-quality speech from texts on general weather conditions," *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E76-A, No.11, pp.1971-1980, 1993.

[6] Q. Sun, K. Hirose, W. Gu, and N. Minematsu, "Generation of fundamental frequency contours for Mandarin speech synthesis based on tone nucleus model," *Proc. Eurospeech*, pp.3265-3268, 2005.

[7] J. Zhang, and K. Hirose, "Tone nucleus modeling for Chinese lexical tone recognition," *Speech Communication*, Vol. 42, Nos. 3-4, pp.447-466, 2004.

[8] Q. Sun, K. Hirose, W. Gu, and N. Minematsu, "Rule-based Generation of Phrase Components in Two-step Synthesis of Fundamental Frequency Contours of Mandarin," *Proc. Speech Prosody*, May 2-5, pp.561-564, 2006.

[9] K. Hirose, M. Eto, and N. Minematsu, "Improved Corpus-based Synthesis of Fundamental Frequency Contours using Generation Process Model," *Proc. ICSLP,* pp.2085-2088, 2002.

[10] K. Hirose, H. Fujisaki and H. kawai, "Generation of prosodic symbols for rule-synthesis of connected speech of Japanese," *Proc. ICASSP*, pp.2415-2418, 1986.

[11] W. Gu, K. Hirose and H. Fujisaki, "Analysis of the effects of word emphasis and echo question on F_0 contours of Cantonese utterances," *Proc. Eurospeech*, pp.1825-1828, 2005.

Design of Tree-based Context Clustering for an HMM-based Thai Speech Synthesis System

Suphattharachai Chomphan, Takao Kobayashi

Interdisciplinary Graduate School of Science and Engineering,
Tokyo Institute of Technology, Yokohama, 226-8502 Japan
{suphattharachai,takao.kobayashi}@ip.titech.ac.jp

Abstract

This paper proposes an approach to improving the correctness of tone of the synthesized speech which is generated by an HMM-based Thai speech synthesis system. In the tree-based context clustering process, tone groups and tone types are used to design four different structures of decision tree including a single binary tree structure, a simple tone-separated tree structure, a constancy-based-tone-separated tree structure, and a trend-based-tone-separated tree structure. A subjective evaluation of tone correctness is conducted by using tone perception of eight Thai listeners. The simple tone-separated tree structure gives the highest level of tone correctness, while the single binary tree structure gives the lowest level of tone correctness. Moreover, the additional contextual tone information which is applied to all structures of the decision tree achieves a significant improvement of tone correctness. Finally, the evaluation of syllable duration distortion among the four structures shows that the constancy-based-tone-separated and the trend-based-tone-separated tree structures can alleviate the distortions that appear when using the simple tone-separated tree structure.

1. Introduction

For tonal languages such as Thai, Mandarin, Cantonese, and Vietnamese, tone is a very important suprasegmental feature of syllable. The words with the same phoneme sequence may have different meanings if they have different tones [1]. Thus, tone must be carefully taken into account in speech synthesis systems of tonal languages.

Meanwhile, HMM-based speech synthesis system is becoming popular in the present day. It was firstly developed for Japanese by Tokuda et al. [2]-[4], and has also been developed for several other languages such as Korean, English, Portuguese, Slovenian, Chinese, and German as indicated in [5]. It has been shown that the HMM-based speech synthesis can be applied successfully to speech synthesis of tonal languages.

In this context, we have attempted to develop an HMM-based Thai speech synthesis system [6]. In the developed system, a group of contextual factors which affect spectrum, pitch, and state duration, such as tone type and part of speech are taken into account especially for the purpose of producing natural sounding prosody of the tonal language. We have found that it can provide speech with the better reproduction of prosody over the unit-selection-based Vaja TTS system [7]. Specifically, a decision tree with a tone-separated structure shows the significant improvement of tone correctness of the synthesized speech. However, some distortion of syllable duration is obviously noticeable when the system is trained with a small amount of data. To overcome this problem, this paper proposes some other structures of the decision tree designed for not only the purpose of maximal correctness of tone but also the purpose of elimination of the syllable duration distortion. In addition to using the designed tree structures, we also apply contextual tone information (tone types of the preceding and the succeeding syllables) to the designed decision-tree structures.

2. Study of Thai tones

2.1. Characteristics of Thai tones

According to the comprehensive study of Thai sound system by Lukseneeyanawin [8], [9], Thai sound is often described in a syllable unit as depicted in Figure 1. The basic Thai textual syllable structure is composed of consonants, vowels, and tone, where Ci, V, Cf, and T denotes an initial consonant, a vowel, a final consonant, and a tone, respectively.

For tonal languages such as Thai, tone, which is indicated by contrasting variations in contour of fundamental frequency (F0) at the syllabic level, is an important part of spoken language because the meaning of words with the same sequence of phonemes can be different if they have different tones. In Thai, there are five tonal variations traditionally named according to the characteristics of their F0 contours within a syllable as shown in Figure 2 [10], [11]. Five IPA tone markers are generally used to indicate Thai tone types; / ¯/ for middle tone (tone 0), / ˋ / for low tone (tone 1), / ˆ / for falling tone (tone 2), / ˊ / for high tone (tone 3), and / ˇ / for rising tone (tone 4). The effect of tone on the linguistic meaning is shown in the following examples: the syllable /k^hāː/ (/คา/ in Thai) has tone 0 and means “to get stuck”, the syllable /k^hàː/ (/ข่า/ in Thai) has tone 1 and means “galangal, a kind of spice”, the syllable /k^hâː/ (/ฆ่า/ in Thai) has tone 2 and means “to kill”, the syllable /k^háː/ (/ค้า/ in Thai) has tone 3 and means “to trade”, and the syllable /k^hǎː/ (/ขา/ in Thai) has tone 4 and means “leg”. By investigating tone occurrence statistics in TSynC speech database (see Section 4.1), we found that 77,413 syllables are occupied firstly by tone 0 (38%), tone 1 (22%), tone 2 (17%), tone 3 (15%), and finally tone 4 (8%).

2.2. Categorizations of Thai tones

Two criteria are used to categorize Thai tones into tone groups as follows. First, by considering the constancy of the F0 contour, Abramson divided the tones into two groups [12]: the static group consists of three tones, high tone, middle tone,

T
Ci(Ci) V(V) Cf

Figure 1: Thai tonal syllable structure.

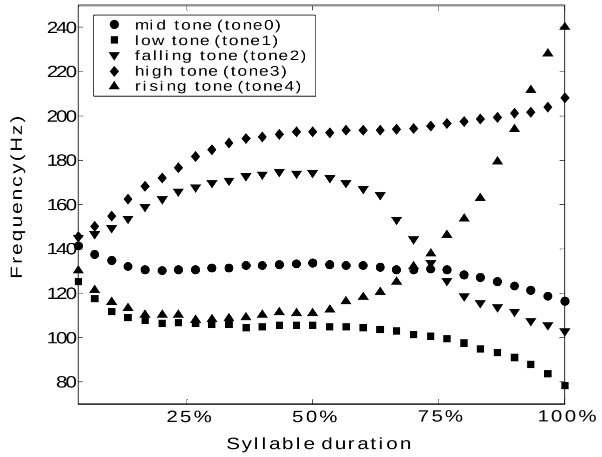


Figure 2: Standard F0 contours for Thai tones.

and low tone; the dynamic group consists of two tones, rising tone and falling tone. Secondly, by considering each contour of figure 2, we can see that the pitch patterns of the mid, low, falling, high, and rising tones are relatively mid-fall, fall, rise-fall, rise, and fall-rise, respectively. As a result, they can be divided according to the final trend of their contours: the upward trend group consists of two tones, high tone, and rising tone; the downward trend group consists of three tones, mid tone, low tone, and falling tone.

3. Tree-based context clustering

In the HMM-based speech synthesis system, context clustering is an important process in the training stage to treat the problem of limitation of training data. Information sharing of training data in the same cluster or leaf node in the decision-tree-based context clustering is the essential concept, therefore construction of contextual factors and design of tree structure for the decision-tree-based context clustering must be done appropriately. The following subsections describe our approach to the issues.

3.1. Construction of contextual factors

A number of contextual factors which is language dependent has been constructed for Thai in [6] to model context dependent HMMs. The following 13 contextual factor sets in 5 levels of speech unit were constructed according to 2 sources of information, including the phonological information [9] (for phoneme and syllable levels), and the utterance structure from Thai text corpus named ORCHID [13] (for word, phrase, and utterance levels).

- Phoneme level

S1. {preceding, current, succeeding} phonetic type
S2. {preceding, current, succeeding} part of syllable structure

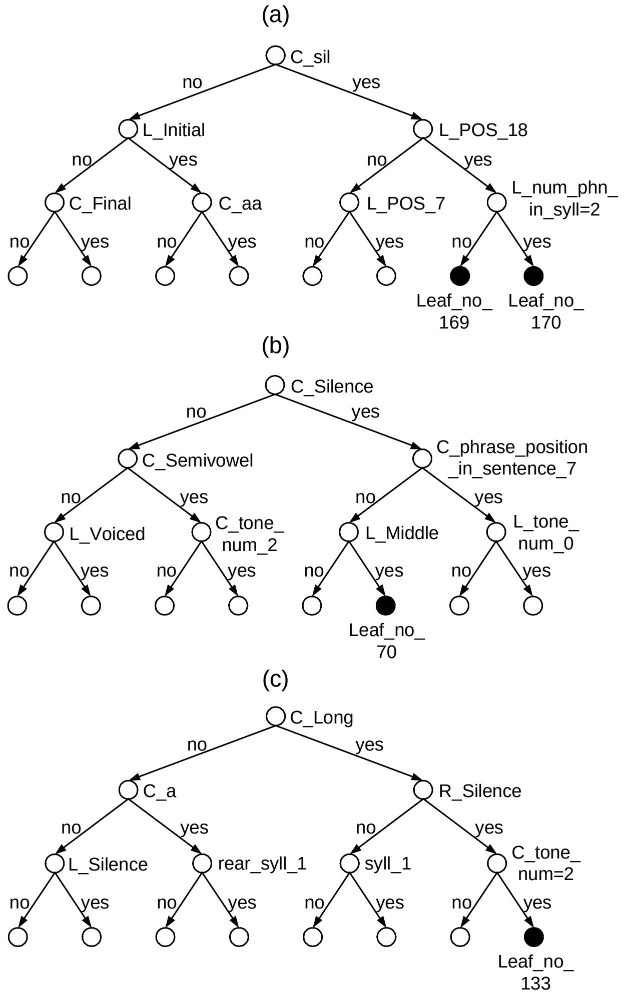


Figure 3: Example of decision trees for: (a) spectrum (3^{rd} state), (b) pitch (2^{nd} state), and (c) state duration.

- Syllable level

S3. {preceding, current, succeeding} tone type
S4. the number of phones in {preceding, current, succeeding} syllable
S5. current phone position in current syllable

- Word level

S6. current syllable position in current word
S7. part of speech
S8. the number of syllables in {preceding, current, succeeding} word

- Phrase level

S9. current word position in current phrase
S10. the number of syllables in {preceding, current, succeeding} phrase

- Utterance level

S11. current phrase position in current sentence
S12. the number of syllables in current sentence
S13. the number of words in current sentence

Subsequently, these contextual information sets were transformed into question sets which finally applied at the context clustering process in the training stage with the total question number of 1156. An analysis of these question sets was conducted in [6] to evaluate the contribution of each set. Figure 3 shows an example of decision trees for spectrum, pitch and state duration by using all of the constructed question sets for the single binary tree context clustering (style (e) in Section 3.3). It can be seen that the root node question in each tree (C_sil from the spectrum tree, C_Silence from the pitch tree, and C_Long from the state duration tree) is of the phonetic type question set. It corresponds to the previous analysis that phonetic type question set is the most important set among all thirteen sets.

3.2. Design of decision-tree structures

The single binary tree structure normally used in the decision tree-based context clustering process is shown in Figure 4 (a). The imbalance of tone frequency causes the prevalence of some tones to the others, as a result, the single binary tree context clustering gives high tone error percentage as indicated in [6]. To improve the tone correctness of the synthesized speech, the simple *tone-separated* decision-tree structure was proposed in [6] as depicted in Figure 4 (b). It has been found that the significant distortion of the generated syllable duration are unavoidable when using the simple tone-separated tree context clustering with small training data due to the limited data in each tone. To treat this problem, the other two structures were designed by taking into account of tone groups and tone types as explained in Section 2.2 and Section 2.1, respectively.

Tone groups categorized in terms of constancy of the F0 contour (proposed by Abramson) were used to design the structure of *constancy-based-tone-separated* tree as depicted in Figure 4 (c). Meanwhile, tone groups categorized by the final trend were used to design the structure of *trend-based-tone-separated* tree as depicted in Figure 4 (d). In the static tone group of the constancy-based-tone-separated tree and the downward trend group of the trend-based-tone-separated tree, no tone-separations are applied because the data sharing among the tones within those groups is expected to alleviate the problem of syllable duration distortion.

3.3. Design of context clustering styles

First, the four structures of decision tree as described in Section 3.2 are applied directly in the context clustering process of the training stage without using the tone type question set (S3 in Section 3.1). The first four styles of context clustering are listed as follows.

(a) Single binary tree context clustering without tone type questions.

(b) Simple tone-separated tree context clustering without tone type questions.

(c) Constancy-based-tone-separated tree context clustering without tone type questions.

(d) Trend-based-tone-separated tree context clustering without tone type questions.

Note that only tone information of the current syllable is concerned in the tone-separated tree structures, while no tone information is concerned in the single binary tree structure. Moreover no other tone information in the neighboring syllables is taken into account in all structures. To exploit the ignored tone information, the tone type question set is incorporated into all of the designed tree structures to form another four styles of context clustering. Those styles of context clustering process are listed as follows.

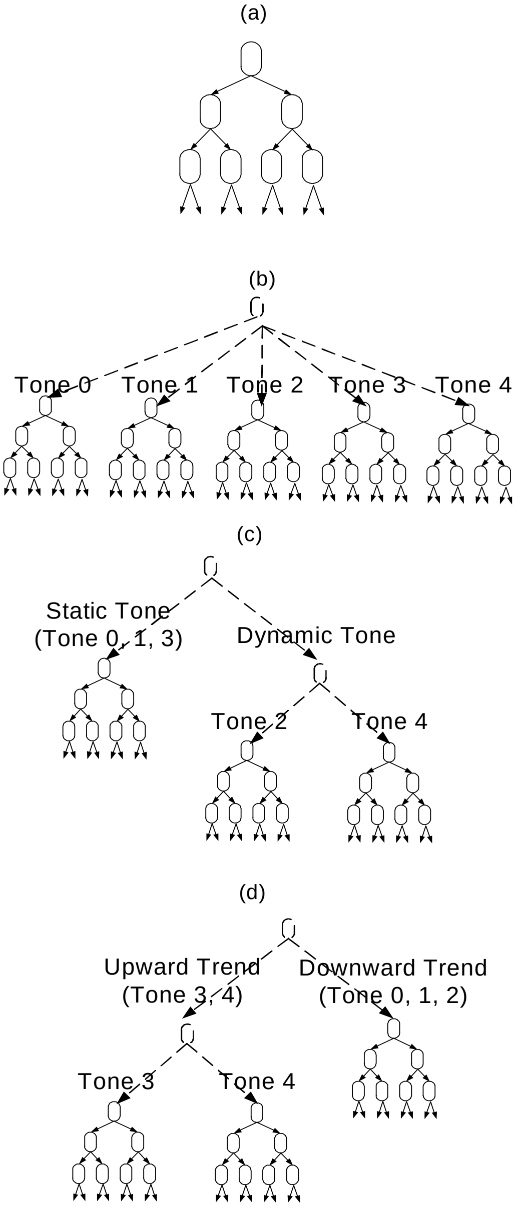


Figure 4: Tree structures for context clustering: (a) single binary tree structure, (b) simple tone-separated tree structure, (c) constancy-based-tone-separated tree structure, and (d) trend-based-tone-separated tree structure.

(e) Single binary tree context clustering with tone type questions.

(f) Simple tone-separated tree context clustering with tone type questions.

(g) Constancy-based-tone-separated tree context clustering with tone type questions.

(h) Trend-based-tone-separated tree context clustering with tone type questions.

4. Experiments

4.1. Speech database and training condition

A set of phonetically balanced sentences of Thai speech database named TSynC from National Electronics and Computers Technology Center (NECTEC) [7] was used for training HMMs. The whole sentence text was collected from Thai part-of-speech tagged ORCHID corpus. The speech in the database was uttered by a professional female speaker with clear articulation and standard Thai accent. The phoneme labels included in TSynC and the utterance structure from ORCHID were used to construct the context dependent labels [6] with 79 different phonemes including 65 phonemes from original Thai words, 12 phonemes from some loan words, and 2 phonemes of silence and pause.

Speech signals were sampled at a rate of 16kHz and windowed by a 25ms Blackman window with a 5ms shift. Then mel-cepstral coefficients were extracted by mel-cepstral analysis. The feature vectors consisted of 25 mel-cepstral coefficients including the zeroth coefficient, logarithm of F0, and their delta and delta-delta coefficients [2].

We used 5-state left-to-right phoneme-sized HSMMs in which the spectral part of the state was modeled by a single diagonal Gaussian output distribution. The number of training utterances was varied as follows: 100, 200, 300, 400, 500, 1000, 1500, 2000, and 2500.

4.2. Evaluation of synthesized speech

Three experiments were performed to evaluate the synthesized speech. First, evaluation of the overall tone correctness of the synthesized speech generated from the HMM-based system with eight different tree-based context clustering styles was done. Secondly, evaluation of the tone correctness for each tone type was performed separately. Finally, the evaluation of syllable duration distortion for four different tree structures was conducted.

4.2.1. Evaluation of overall tone correctness

This section presents how the overall tone correctness of the synthesized speech is improved by using eight different tree-based context clustering styles described in Section 3.3. Figure 5 shows an example of F0 contours of the natural speech and synthesized speech with different clustering styles. The first full-shape syllable of Figure 5 pronounced as /thǎ/ ("magic" in English) conveys tone 4 or rising tone. Figure 5 (a) is of the single binary tree context clustering without tone type questions, however this syllable contour is misshaped. As a result, most listeners perceived it with wrong tone. Meanwhile Figures 5 (b) - (h) are of the other styles, and they show the improvement of the F0 contour shape conforming to that of the natural speech as depicted in Figure 5 (i). To evaluate the overall tone correctness of our implemented system, a subjective test was conducted. The 2,289 syllables of 100 synthesized speech utterances were presented to eight native subjects. Then the subjects were requested to decide whether the syllables have the same tones as the given texts or not. The average tone error percentages for the first four styles and another four styles are summarized in Figures 6 and 7, respectively.

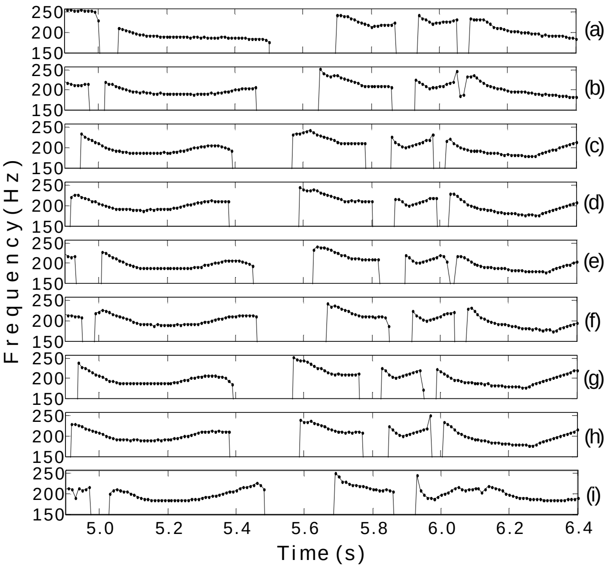


Figure 5: F0 contours of synthesized speech from 8 different clustering styles; (a) single binary tree without tone type questions, (b) simple tone-separated tree without tone type questions, (c) constancy-based-tone-separated tree without tone type questions, (d) trend-based-tone-separated tree without tone type questions (e) single binary tree with tone type questions, (f) simple tone-separated tree with tone type questions, (g) constancy-based-tone-separated tree with tone type questions, (h) trend-based-tone-separated tree with tone type questions, and (i) F0 contour of natural speech.

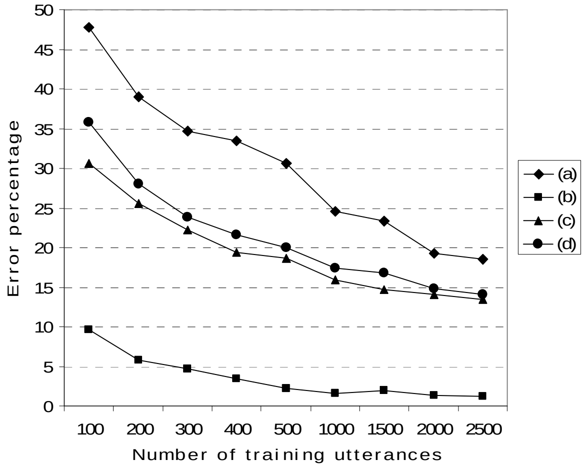


Figure 6: Tone error percentages of synthesized speech from 4 different clustering styles; (a) single binary tree without tone type questions, (b) simple tone-separated tree without tone type questions, (c) constancy-based-tone-separated tree without tone type questions, and (d) trend-based-tone-separated tree without tone type questions.

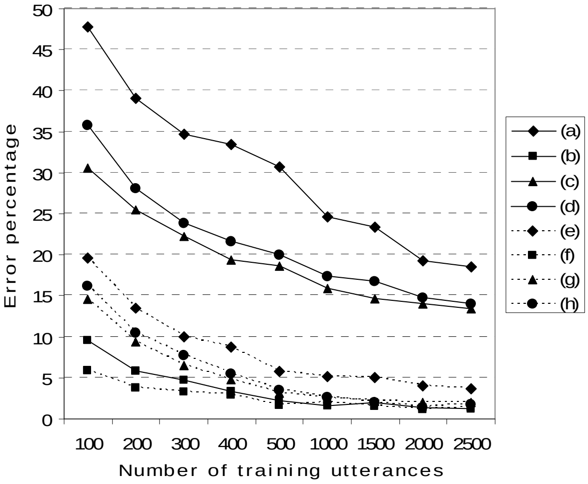


Figure 7: Tone error percentages of synthesized speech from 8 different clustering styles; (a) single binary tree without tone type questions, (b) simple tone-separated tree without tone type questions, (c) constancy-based-tone-separated tree without tone type questions, (d) trend-based-tone-separated tree without tone type questions (e) single binary tree with tone type questions, (f) simple tone-separated tree with tone type questions, (g) constancy-based-tone-separated tree with tone type questions, and (h) trend-based-tone-separated tree with tone type questions.

Figure 6 shows the tone error percentages of synthesized speech from the context clustering styles (a)-(d) as described in Section 3.3. Comparing the four designed tree structures, we can see that the style (a) (single binary tree without tone type questions) gives the highest level of error percentage, the style (c) (constancy-based-tone-separated tree without tone type questions) and the style (d) (trend-based-tone-separated tree without tone type questions) can reduce the error percentage significantly, while the style (b) (simple tone-separated tree without tone type questions) gives the lowest error percentage. In other words, the context clustering with the tone-separated tree structure has more effectiveness than the context clustering with the single binary tree structure. We can also see that the tone error percentage is decreased as the number of training utterances is increased.

Figure 7 shows the tone error percentages of synthesized speech from the context clustering styles (e)-(h) relative to the styles (a)-(d), respectively. The tone type question set was applied to those four styles. It can be seen that the contextual tone information in syllable level causes a drastic reduction of the tone error percentage for all tree structures except the simple tone-separated tree context clustering. The reason is that the simple tone-separated tree structure exploits all tone type questions of the current syllable in the separation of tree structure, while the single tree structure does not exploit the tone information at all and the constancy-based-tone-separated, trend-based-tone-separated tree structures exploit partly of the tone information. Therefore the effect of the tone type question set which is employed afterward to the simple tone-separated tree structure is smallest among that of all other tree structures.

From Figures 6 and 7, it can be also seen that the constancy-based-tone-separated tree structure is more effective in giving a little lower error percentage than the trend-based-tone-separated tree structure.

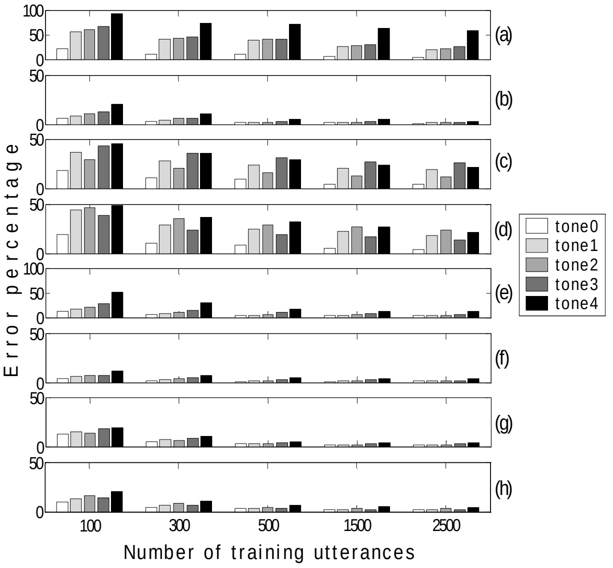


Figure 8: Tone error percentages of synthesized speech from 8 different clustering styles categorized by tone types; (a) single binary tree without tone type questions, (b) simple tone-separated tree without tone type questions, (c) constancy-based-tone-separated tree without tone type questions, (d) trend-based-tone-separated tree without tone type questions (e) single binary tree with tone type questions, (f) simple tone-separated tree with tone type questions, (g) constancy-based-tone-separated tree with tone type questions, and (h) trend-based-tone-separated tree with tone type questions.

4.2.2. Evaluation of tone correctness for each tone type

This section presents the tone correctness in terms of the tone types. The result is presented in Figure 8. For 100 synthesized speech utterances from Section 4.2.1, the numbers of the syllables with tone 0, tone 1, tone 2, tone 3, and tone 4 are 750, 560, 449, 339, and 191, respectively.

From Figure 8 (a) or the single binary tree without tone type questions and (b) or the simple tone-separated tree without tone type questions, the error percentage of tone 4 is mostly highest among all tones, on the other hand, the error percentage of tone 0 is mostly lowest. The reason is that the proportion of training data of tone 4 is smallest while the proportion of training data of tone 0 is largest according to the statistics of tone occurrence in the speech database as described in Section 2.2. From Figure 8 (c) or the constancy-based-tone-separated tree without tone type questions, the error percentages of tone 2 and 4 are noticeably reduced as compared to (a) or the single binary tree without tone type questions. Meanwhile, from Figure 8 (d) or the trend-based-tone-separated tree without tone type questions, the error percentages of tone 3 and 4 are reduced as compared to (a) or the single binary tree without tone type questions.

As for Figures 8 (e) – (h) in which the tone type question set is employed, it can be seen that the tone error percentages of tone 0 – 4 are rather close to each other and also much less than those of Figures 8 (a) – (d).

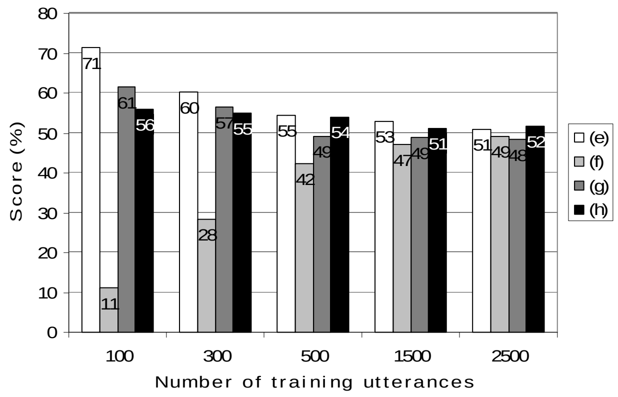


Figure 9: Scores of a paired-comparison test for natural duration among 4 different clustering styles; (e) single binary tree with tone type questions, (f) simple tone-separated tree with tone type questions, (g) constancy-based-tone-separated tree with tone type questions, and (h) trend-based-tone-separated tree with tone type questions.

4.2.3. *Evaluation of syllable duration distortion*

A paired-comparison test among all tree structures (the styles (e)-(h)) of the context clustering with tone type questions was performed. Ten test sentences selected randomly from 100 synthesized speech utterances in Section 4.2.1 were used in this evaluation. For each comparison, a pair of utterances from two of the four structures is presented to eight subjects and then the subjects are requested to choose the one which has more natural duration without considering the correctness of tone. The average scores in percentage of each tree structure with the different number of training utterances are shown in Figure 9.

The single binary tree structure gives the least distortion among all tree structures. On the other hand, the simple tone-separated tree structure gives the worst distortion compared to the other structures, because there is no sharing of data between each tone for the simple tone-separated tree structure, meanwhile there is some data sharing for the single binary tree structure, the constancy-based-tone-separated tree structure and the trend-based-tone-separated tree structure as seen in Figure 4. However it can be seen that the distinction between the scores disappears when the number of training utterances is increased above 1000.

The synthesized speech samples are available on the website: http://www.kbys.ip.titech.ac.jp/demo/thai/index.html

5. Conclusions

An analysis of tree-based context clustering of an HMM-based Thai speech synthesis system has been conducted in this paper. Four structures of decision tree were designed according to tone groups and tone types to obtain higher correctness of tone of synthesized speech. The results show that the tone-separated tree structures can reduce the tone error percentage of the synthesized speech compared to the single binary tree structure significantly. As for using the contextual tone information in the syllable level, it can improve the tone correctness for all structures of decision tree. There are some distortions of the syllable duration appearing in the case of using the simple tone-separated tree context clustering with a small amount of training data, however it can be relieved when using the constancy-based-tone-separated or the trend-based-tone-separated tree context clustering.

The analysis of tone correctness of the average-voice-based speech model and the intonation analysis issues are anticipated to be studied in the future.

6. Acknowledgements

The authors are grateful to NECTEC for providing us the TSynC speech database.

7. References

[1] Seresangtakul, P., and Takara, T., "Analysis and synthesis of pitch contour of Thai tone using Fujisaki's model", IEICE Trans Inf. & Syst., Vol.E86-D, No.10, pp.2223-2230, 2003.

[2] Tokuda, K., Kobayashi, T., and Imai, S., "Speech parameter generation from HMM using dynamic features", Proc. ICASSP-95, Vol.1, pp.660-663, 1995.

[3] Masuko, T., Tokuda, K., Kobayashi, T., and Imai, S., "Speech synthesis using HMMs with dynamics features", Proc. ICASSP-96, pp.389-392, 1996.

[4] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T., "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis", Proc. EUROSPEECH-99, pp.2347-2350, 1999.

[5] Black A.W., Zen H., and Tokuda, K., "Statistical parametric speech synthesis", Proc. ICASSP-2007, pp.1229-1232, 2007.

[6] Chomphan, S., and Kobayashi, T., "Implementation and evaluation of an HMM-based Thai speech synthesis system", submitted to Proc. INTERSPEECH-2007, 2007.

[7] Hansakunbuntheung, C., Rugchatjaroen, A., and Wutiwiwatchai, C., "Space reduction of speech corpus based on quality perception for unit selection speech synthesis", Proc. SNLP-2005, pp.127-132, 2005.

[8] Wutiwiwatchai, C., and Furui, S., "Thai speech processing technology: a review", J. Speech Communication, Vol.49, pp.8-27, 2007.

[9] Luksaneeyanawin, S., "Linguistics research and Thai speech technology", Proc. of the 5th International Conference on Thai Studies, School of Oriental and African Studies, 1993.

[10] Thathong, U., Jitapunkul, S., and Ahkuputra, V., "Classification of Thai consonants naming using Thai tone", Proc. ICSLP-2000, Vol.3, pp.47-50, 2000.

[11] Chompun, S., "Fine granularity scalability for MP-CELP based speech coding with HPDR technique", Proc. APCCAS-2004, pp.197-200, 2004.

[12] Abramson, A. S., "Lexical tone and sentence prosody in Thai", Pro. International Congress of Phonetics Science, pp.380-387, 1979.

[13] Sornlertlamvanich, V., Takahashi, N., and Isahara, H., "Thai part-of-speech tagged corpus: ORCHID", Proc. Oriental COCOSDA Workshop, pp.131–138, 1998.

Development of a BOSS unit selection module for tone languages

Arne Bachmann, Stefan Breuer

Institute of Communication Sciences (IfK)
University of Bonn, Germany

arne.bachmann AT web.de, breuer AT ifk.uni-bonn.de

Abstract

The Bonn Open Synthesis System (BOSS) is a toolkit for the efficient development of speech synthesis applications. To facilitate adaptation to tone languages, we added support for tone contour quantization and prediction. Now it is possible to integrate syllable and word tone templates into the system and predict as well as select them efficiently. The simple model presented here is trained automatically and works independently of the morphophonemic rules specific to a certain tone language. Its feasibility is exemplified for the African language *Ibibio*.

1. Introduction

1.1. BOSS

The *Bonn Open Synthesis System* BOSS [1] is a research and development platform originally written for unit selection-oriented speech synthesis, but also applicable to other approaches [2]. Its building blocks are reusable libraries and language modules (German, Polish, Ibibio) in C/C++. BOSS also provides tools for creating and optimizing corpora. The system communicates and stores data using XML files and their DOM representations; runtime access to corpus data is optimized for speed by use of MySQL [3] databases (DB). BOSS is a network-enabled application. Communication between synthesis server and clients works over a simple protocol that hosts XML and audio data. Synthesis can be executed from the command line or by a Java GUI client. BOSS provides a bootstrap install mechanism and thus can be installed and run on Unix-based platforms. German online-synthesis is available at the BOSS website [1]. As a platform intended for research, BOSS is not optimized for limited resources, although many ideas for optimization are conceivable and waiting for implementation. Feel free to contribute to the BOSS project.

1.2. Objective

Our first aim was to write a BOSS intonation/unit selection module for the African tone language *Ibibio*. In cooperation with Prof. Urua (University of Uyo, Uyo, Nigeria), the fundamentals of the module were planned and worked out in [4].

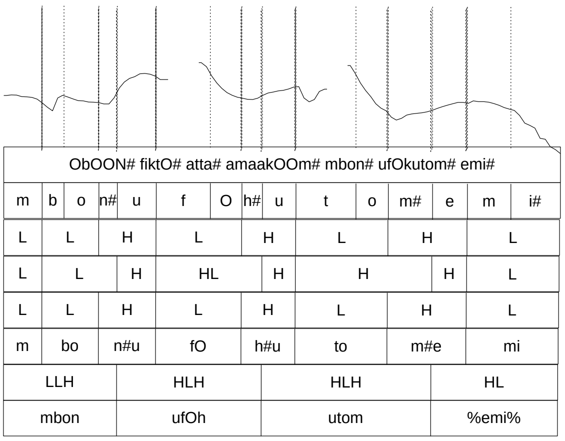
The main objectives for a general tone language intonation module are adaptability, extensibility, simplicity, ease and speed of development, run-time speed, universality and knowledge gain through machine learning (ML). Since tone languages like Ibibio exhibit intriguingly complex intonation, e. g. may combine phenomena such as declination, downdrift, downsteps, final fall and tone assimilations, it is very hard to derive rules for a rule-based intonation synthesis manually. Some rough rules for Ibibio can be found in the literature, e. g. lowering of the topline by 30 Hz after downsteps and final fall of about 10 Hz [5], but there is no integrated model to describe the interactions between the various influences on the suprasegmental structure of the language. To avoid the tedious search for uncertain regularities, we leave it to the machine to learn the patterns of intonation and therefore gain a reusable and easily retrainable system.

1.3. Ibibio

Ibibio is one of over 1500 Niger-Congo languages and is spoken in the southwestern part of Nigeria by about 5 million people. The language has three tonemes (high **H**, low **L** and downstepped high **D**), plus two non-contrasting surface contour tones (rising **R** and falling **F**) [6]. Usually, tones are not represented in orthography. Ibibio shows interesting tonal features: Tones are lexically and grammatically distinctive. There are complex morphophonemic word tone templates (cf. [5]):

Ibibio	**English**
sé	look
áà-sèè-hè	one who looks
áà-!ké-séé-hé	one who looked
áà-!dî-sé	one who will look
nò̥	give
áà-nò̥ò̥-hò̥	one who gives
áà-!ké-nò̥ò̥-hó̥	one who gave
áà-dî-nò̥	one who will give

In this notation, **!** is a downstep, the subring shows the presence of a deleted underlying (floating) tone. There is also downdrift [7]. This means that like consecutive tones have the same fundamental frequencies; they may need some time to reach the target frequency, though, a phenomenon called *start-up effect* [5].



ObOON# fiktO# atta# amaakOOm# mbon# ufOkutom# emi#

m b o n# u f O h# u t o m# e m i#

L L H L H L H L

L L H HL H H H L

L L H L H L H L

m bo n#u fO h#u to m#e mi

LLH HLH HLH HL

mbon ufOh utom %emi%

Figure 1: *Sample corpus sentence. The annotation tiers shown are (from top to bottom): Sentence transcription, phones, surface tones, underlying tones, mapping of tier 3 to single letter tone symbols, syllables, word tones, words.*

For the research presented in this paper, we used a small annotated Ibibio corpus of 94 sentences spoken by Prof. Urua. An excerpt of the corpus is shown below.

121: /ana ekop nsɔŋidem odo ɔjɔhɔ ɔjɔhɔ ke ifuuro uwem/
122: /ekæriku odo akwa owo eto/
123: /mmɔɔŋ idɪm ukana amaasideŋŋe/

2. Methods and models

2.1. The linear approach

As for most languages we do not know the optimal intonational decomposition in advance, we must leave it to the ML to learn it. Our model predicts a sequence of syllable tone contours on the basis of the symbolic description of a target utterance's tonal surface structure. We simply feed it alongside all other available parameters such as number of preceding hightones or downsteps into the ML, thus not anticipating any language-specific features or assuming erroneous regularities. To adapt our module to a new language, the only necessary change to the ML is the choice of features, derived from recommendations in the literature. Other reasons for preferring a linear approach over superposition are:

- Ease of data extraction: Obtaining the observable surface intonation contour is straightforward
- No global component necessary (e. g. phrase commands)
- The tone-bearing unit (TBU) is usually a syllable, so complexity is reduced by concatenating syllable contours

One problem arising from the linear tone sequence model is the possibility of F_0 discontinuities at concatenation boundaries. This problem seems neglectable, because a well-trained prediction module should produce only small differences at syllable boundaries which can be smoothed easily in unit selection and signal manipulation.

2.2. F_0-Stylization

For automatic extraction of fundamental frequency contours, we used ESPS' *get_f0* [8] as well as *Praat* [9]. For the application to our Ibibio corpus, we faced the problem that syllable boundaries were not annotated. Thus we had to resort to the boundaries marked for surface tones.

In contrast to modeling intonational events known from accent languages e. g. by using the Tilt model [10], we use quartic polynomials to stylize syllable contours. Our method does not require any suprasegmental markup and therefore rather resembles the PaIntE model [11]. One advantage of using polynomial regression is ease of implementation. On the downside, there may be unsolvable systems of equations due to a lack of data points, e. g. in voiceless sections or where the extraction algorithm calculated F_0 values out of a sensible range.

To determine the optimal polynomial order, we tested the quality of stylization for different regression settings on the Ibibio corpus. To this end, we stylized and resynthesized the intonation contours using original syllable durations. The differences in the extracted F_0 values between original and polynomial contours were measured by means of a Perl script. Contour accuracy is shown in table 1. The values represent root mean square error (RMSE) and mean absolute errors (MAE) in Hertz plus Pearson's correlation coefficient r for different polynomial orders. We also experimented with stylization on a logarithmic scale, but results were slightly less satisfactory.

In the end, stylization was done using polynomials of fourth order, as this gave the best ratio between approximation quality and the number of solvable equation systems and thus the number of syllables that could be used for training. The polynomials are stored using both their coefficients and a data point representation. The latter is derived by computing the polynomial values for the left and right syllable boundaries, the middle, and two points between the middle position and the borders. This way, we get tone contour descriptions that are independent of the syllable durations. We don't apply an F_0 normalization to the contour shapes, because the absolute data values may serve to distinguish different intonational functions.

2.3. Data reduction

We use a simple vector quantization (VQ) approach to reduce the syllable contour data. This serves two purposes: firstly, to create a reasonably concise amount of distinct syllable contours for machine learning, and secondly to gain knowledge over the most common syllable contours and their linguistic/phonological distribution. The data reduction method used in our module is the well-known LBG algorithm [12]. To reduce the set of observed syllable contours, their polynomial functions are taken to form a vector space by using five equidistant data points. Thus, each syllable can be represented by a quartic polynomial. We don't use the polynomial coefficients at this stage, as their values have different dimensions and cannot be compared by simple distance measures as necessary in the VQ.

The LBG algorithm successively divides the vector space into halves. The resulting 2^N prototypes, collected in a *codebook*, represent an optimal partitioning of all data points (read: syllable contours) in the vector space. As can be seen in figure 2, we chose a codebook size of 64 entries, which was the best choice for the current small size of the corpus.

Afterwards, the codebook's prototypes themselves were vector quantized to get a set of superclasses — a further layer of abstraction to be used as a fallback in unit selection whenever there is no unit available for a certain prototype. For our corpus, the best combination of codebook and codebook classes in terms of distortion minimization was achieved for 64/16 codevectors, as can be seen in table 2. Re-quantizing the existing codebook, as if it were a set of original contour shapes attempts to reduce data that is already optimally distributed in the vector space. Additionally, the original frequency distribution (number of data vectors per codevector) is neglected. This approach can be improved. One possibility would be to create a smaller codebook from the original data and to assign the original data points to these classes. When looking at the produced protoypical syllables shapes of the codebook in figure 2, one can observe several properties of the algorithm: The overall fundamental frequency curve rises with the number of codevectors and 2^N neighbouring codevectors share some features, e. g. rising or falling shape, while varying in others.

Order	RMSE [Hz]	MAE [Hz]	r
1	10.55	6.74	0.964
2	6.98	4.26	0.985
3	5.47	3.17	0.991
4	**4.53**	**2.49**	**0.994**
5	4.02	2.14	0.995

Table 1: *F_0-stylization accuracy for various polynomial orders. The order used is printed in bold letters.*

The VQ automatically tries to make its codevectors represent the data vectors best, so we can assume that a fairly large codebook represents the most frequent tone contours of a language. The codebook provides the essential interface between surface acoustic and surface symbolic-phonetic information and with that, the phonological categories[1].

2.4. Prediction

Our choice for a prediction method started with the following considerations:

- There should be tools available for training
- Robust creation and prediction
- Low implementation and integration cost for BOSS
- Human-understandable ML knowledge gain

Thus, neural networks (NN) and support vector machines (SVM) were discarded in favor of classification and regression trees (CART) [13].

Advantages of CARTs include:

- Very fast execution and low memory usage in working phase (binary trees)
- Already implemented in BOSS as a parser for LISP-like decision tree files
- Good tools available (wagon [14]), simple setup and training
- No black box: Human-comprehensible and extensible decisions in a simple tree structure. Potential linguistic knowledge gain concerning tonal phenomena

In the German BOSS module, regression trees are already used for phone duration prediction. We extended the source code to predict classes in addition to mean/standard deviation pairs on the tree leaves in order to be able to use the implementation both for tone contour and duration prediction in the Ibibio module. By using CARTs, it is possible to recognize superpositions in the tree structure as similar returning decisions in different branches. We should thus be able to detect the individual influences of the input parameters on the resulting tonal contours. Disadvantages of the CART include: The importance of single training parameters may vary strongly upon but small changes to the DB. Secondly, like all data-driven machine learning methods, the availability of large amounts of reliable data is essential for successful training.

[1] Presuming the relation between phonology and symbolic surface structure is known.

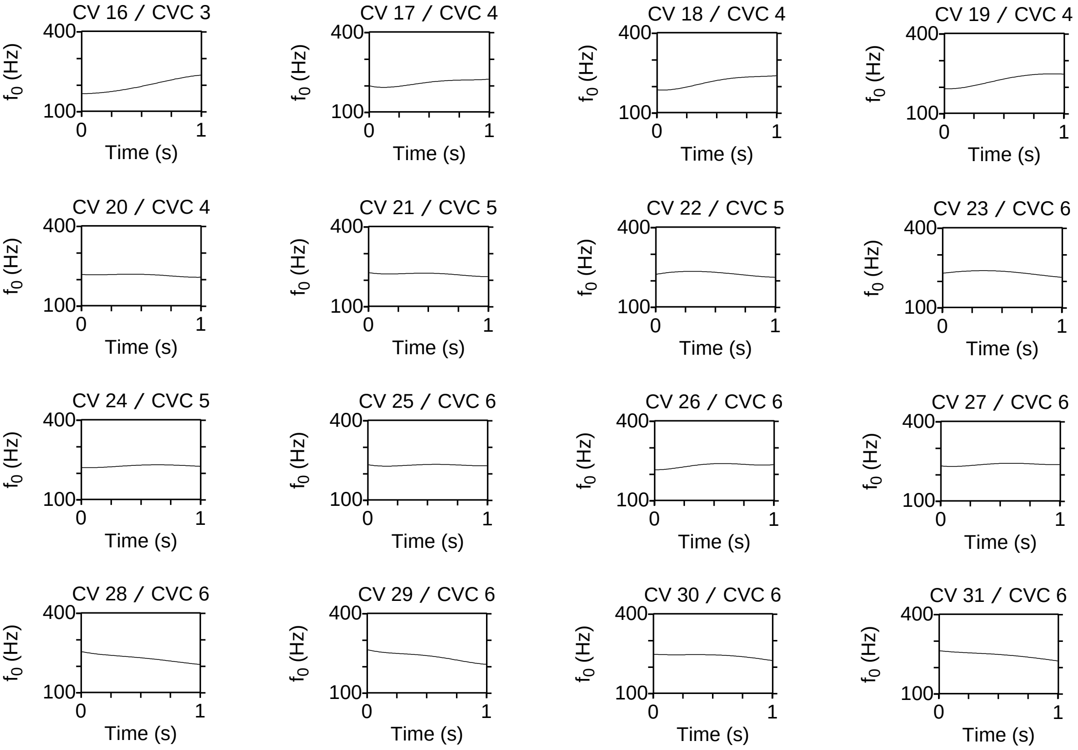


Figure 2: *Vector quantization: Codebook excerpt for codevectors 16. . .31 out of 64*

Order	Syllables	Size C_1	Distortion	SNR	Size C_2	Distortion	SNR
3	2333	64	230.31	-23.62	8	568.68	-27.55
4	2322		224.03	-23.50		531.99	-27.26
5	2067		240.54	-23.81		5656.15	-37.52
3	2333		230.31	-23.62	16	314.46	-24.98
4	**2322**		**224.03**	**-23.50**		**290.29**	**-24.63**
5	2067		240.54	-23.81		4836.02	-36.84
3	2333	128	156.82	-21.95		447.28	-26.51
4	2322		156.35	-21.94		389.75	-25.91
5	2067		170.50	-22.32		3762.66	-35.75
3	2333		156.82	-21.95	32	309.64	-24.91
4	2322		156.35	-21.94		258.54	-24.13
5	2067		170.50	-22.32		2744.57	-34.38

Table 2: *Vector quantization: Comparison of different codebook sizes and polynomial orders, smoothing of all voiceless syllable parts. Values given are overall distortion in data fitting and signal-to-noise ratio (SNR: $-10\log_{10} dist$). The best codebook size combination is printed in bold letters.*

A typical training set of 84 sentences for CART constructed from our Ibibio corpus would leave only about 4'30" of speech, not counting pauses. This is the amount left after removing ten sentences for testing purposes, which is clearly not enough to demonstrate the full potential of our approach. The results presented here should thus be seen as a preliminary estimation. Applying the CART for tone contour prediction to the test set rendered results ranging from 38.55 - 59.04 %, the large interval between the outcomes already indicating a sparsity problem. Duration prediction results were slightly better, but still unsatisfactory for the same reason. Table 3 lists the five most important parameters in codevector and duration prediction trees, taken from sample training set no. 7. The parameters *sylphrase wordphrase* represent the position of the respective unit in the phrase, *sylsphrase* and *wordsphrase* the number of these units it is comprised of. *Sylstruc* encodes the syllable type[2]. The left and right tonal context of each syllable was captured by *ltone4...ltone1* and *rtone1...rtone4*. Parameter *d* is the number of preceding downsteps in the phrase, and *firstcons* stands for the first consonant of the syllable. The distances to left and right phrase boundaries are given by *bodil* and *bodir*. Other features used for training are *r* and *f* for the number of preceding L-H and H-L tone shifts, respectively. For numbered features we also added categorical versions with the possible values initial, medial, final and single. The features used for prediction were collected from recommendations in the literature; an explanation of all features is given in [4]. After training, the decision structure of the tree was analyzed. Especially the role of the number of downsteps and of downdrift was inspected, but the impact of downsteps predicted in the literature was not transparent in the CART. Since *sylphrase* was the dominant decision feature in most trained trees, we would rather assume a declination component for the current data. More data has to be segmented and annotated for further investigation of the role of *r*, *f* and *d*. With respect to the different tonal shapes, only one valley shape was found in the codebook. Thus, a more restricted parametrical representation might also have worked.

Contour classification		**Duration regression**	
Feature	**% correct**	**Feature**	**% correct**
sylphrase	62.3	sylstruc	80.7
wordphrase	64.5	rtone	85.7
sylstruc	66.4	sylphrase	87.4
rtone3	67.4	firstcons	88.3
d	68.3	bodir	88.7

Table 3: *Most important five prediction features for tone template and duration CARTs and their cumulative prediction accuracy.*

2.5. Unit selection

BOSS employs a stepwise reduction of unit search criteria called preselection to reduce the number of database lookups. Thus, if no perfect fitting unit can be found — judging from the symbolic description only — the context is widened and other possible, but less narrowly defined, units come into selection focus.

We introduced two new cost functions to the Ibibio module: To compare the syllable tone contours, the data points from the codebook and those found in the corpus units are compared via RMSE. On the phone level, a categorical measure for the position inside the syllable was introduced with initial, medial, final and single as possible values. For mean syllable F_0 unit and transition costs, the standard BOSS approach is used.

Determining the weighting (or cost) factors for the different unit selection cost functions is a non-trivial problem. In our approach, we normalized all cost functions by their corpus mean value and weighted them in same parts.

A critical problem was the small corpus size: Even after widening the search focus maximally, for some test sentences no fitting syllables or even single phones were found in the database. This calls strongly for a bigger corpus. Additionally, the Ibibio module was originally designed only for syllable-based synthesis, so that phone synthesis represents an unsatisfactory solution. This stems from the fact that we predict syllable tones, and therefore it is hard to tell if a phone fits a given syllable contour. The forementioned phone cost term is one method to remedy this.

2.6. Signal manipulation

Until now, no signal manipulation has been implemented. There are two reasons: Corpus synthesis should in principle work without manipulation (and it does) and development time was restricted to six months in [4]. In principle, BOSS supports PSOLA manipulation, but the modules expect F0 contours as input which would have required an additional transformation function for codevectors. While the general algorithm for recreating a polynomial shape from the codevectors can be found in the BOSS-IBB documentation [15], it was not implemented in this first version of the Ibibio package modules.

3. Discussion

We have shown a syllable-based tone contour codebook synthesis with CART ML to be feasible. We believe that our model should be applicable to other tone languages and our prototypical implementation for Ibibio

[2] The symbols C, V and N were used to represent consonants, vowels and nasals respectively. The latter were included to account for the special importance of nasals in Ibibio.

could serve as a template for the creation of other language adaptations. So far, we have presented some evaluation results on the accuracy of polynomial fitting and vector quantization. With only the small amount of Ibibio data at hand, meaningful subjective listening tests with native Ibibio speakers could not be conducted. Data sparsity affected not only the reliability of the CART trees but also the number of units to choose from for synthesis. Thus, the next step will have to be the creation of a much larger corpus to synthesize from and retraining of the CART and CBs, as well as testing the method on other languages. Criteria to examine in listening tests based on the new data could pleasantness, naturalness, intelligibility and overall intonation.

Some of the technical work under way is the creation of an independent reference module as a starting point for other language modules. This is planned to be done for BOSS-IBB V 0.2. Other language adaptions waiting for realization are Yoruba and Chinese. To test the applicability to accent languages, the method shall be evaluated for German as well.

Other future plans include the improvement of tone template classes and a closer examination of the phonological role of downsteps and downdrift in Ibibio.

4. Acknowledgements

We would like to thank all the people who supported us with their knowledge and by providing additional source code: Axel Bußmann (Osnabrück / Germany), Moses Ekpenyong (Uyo / Nigeria), Prof. Dafydd Gibbon (Bielefeld / Germany), Prof. Wolfgang Hess (Bonn / Germany), Nam Phamdo (Laurel / United States) and Prof. Eno-Abasi Urua (Uyo / Nigeria).

5. References

[1] [Online]. Available: http://www.ikp.uni-bonn.de/dt/forsch/phonetik/boss/

[2] P. Birkholz, I. Steiner, and S. Breuer, "Control Concepts for Articulatory Speech Synthesis," *This conference*, 2007.

[3] "MySQL 5.0 Reference Manual," 2006. [Online]. Available: http://www.mysql.org

[4] A. Bachmann, "Ein quantitatives Tonmodell für Ibibio. Entwicklung eines Prädiktionsmoduls für das BOSS-Sprachsynthesesystem," Master's thesis, University of Bonn, Aug 2006.

[5] E.-A. E. Urua, "The tone system of Ibibio," in *Typology of African Prosodic Systems Workshop*, may 2001.

[6] E.-A. E. Urua, *Ibibio*, ser. Journal of the international phonetic association. International phonetic association, 2004, no. 34/1, ch. Ibibio, pp. 105–109.

[7] B. Connell, "Downdrift, Downstep and Declination," in *Typology of African Prosodic Systems Workshop*. Bielefeld University, may 2001.

[8] [Online]. Available: http://computing.ee.ethz.ch/sepp/esps-5.3.1-vj/

[9] P. Boersma and D. Weenink, "Praat: doing phonetics by computer (Version 4.5.19) [Computer program]," 2005, retrieved April 15, 2007. [Online]. Available: http://www.praat.org

[10] K. Dusterhoff and A. W. Black, "Generating F0 contours for speech synthesis using the tilt intonation theory," in *Proceedings of the 1997 ESCA Workshop on intonation*, Athens, Greece, 1997.

[11] G. Möhler and A. Conkie, "Parametric modeling of intonation using vector quantization," 1998.

[12] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," in *IEEE transactions on communications*, vol. 28, jan 1980, pp. 84–95.

[13] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. New York: Chapman and Hall (Wadsworth, Inc.), 1984.

[14] P. Taylor *et al.*, *Edinburgh Speech Tools Library - System Documentation Edition 1.2, for 1.2.0*, June 15 1999.

[15] A. Bachmann, *BOSS-IBB. Speech Synthesis Module Documentation for the BOSS Ibibio module*, University of Bonn, Apr 2007, revision 4.

Unit-Selection Text-to-Speech Synthesis Using an Asynchronous Interpolation Model

Alexander Kain [1,2]   *Jan P. H. van Santen* [1,2]

[1] Center for Spoken Language Understanding (CSLU)
OGI School of Science & Engineering at OHSU
20000 NW Walker Road, Beaverton, OR 97006, USA

[2] BioSpeech, Inc.
940 Upper Devon Lane
Lake Oswego, OR 97034, USA

Abstract

We describe the Asynchronous Interpolation Model, which represents speech as a composition of several different types of feature streams that are computed using asynchronous interpolation of neighboring basis vectors, according to transition weights. When applied to the acoustic inventory of a concatenative Text-to-Speech synthesizer, the model eliminates concatenation errors and affords opportunities for high rates of compression and voice transformation. We propose a particular instance of the model that uses formant frequency values and formant-normalized complex spectra as two types of streams, in conjunction with a unit-selection synthesizer. During analysis, basis vectors and transition weights were estimated automatically, using three different labeling schemes and dynamic programming methods. An evaluation of the intelligibility and quality of the synthesized speech showed significant improvements over a standard, size-matched compression scheme. The proposed method was also able to convincingly transform speaker characteristics through replacement of basis vectors.

1. Introduction

Today's most natural sounding Text-to-Speech (TTS) synthesis systems are based on the *concatenative synthesis* approach, which uses a multitude of pre-recorded speech "chunks" (a contiguous section of natural speech) of a single speaker, stored in an *acoustic inventory*, to stitch together a new output signal. The quality of the resulting speech relates directly to the size of the database, because the larger the chunks, the fewer the number of concatenation points at which audible artifacts can occur. Moreover, when the prosodic space is not covered by the acoustic inventory, prosodic modification becomes necessary, further degrading the speech signal. The concatenative approach can be contrasted with the *formant synthesis* approach, which is compact in size, gives full prosodic and spectral control over the speech signal, and is highly intelligible, but which does not sound very natural.

Researchers have attempted to improve the problem of audible discontinuities in concatenative synthesis, by interpolating in the formant, waveform, or suitable linear predictive coding domains [1, 2, 3, 4]. However, these approaches commonly neither increase synthesis flexibility nor address the issue of compactness.

We propose a model that combines aspects of both the formant and the concatenative approaches, called the Asynchronous Interpolation Model (AIM). Its features are:

- Elimination of concatenation errors, because speech units of the acoustic inventory have identical representations at concatenation points.
- Opportunity for compression. Even though memory is continuing to decline in price and increase in capacity, it is attractive to control the size/quality trade-off, and thus enable large acoustic inventories on (extremely) storage-limited devices such as cellphones. AIM can take advantage of the special properties of an acoustic inventory, which are that the inventory consists of a single speaker, is acoustically constant and noise-free, non-real-time encoding is possible, all data is known beforehand, and additional information such as phonetic content is available.
- Increased spectral flexibility. For example, changing the duration of a segment of speech changes its spectral properties in complex ways. The increased flexibility of AIM allows non-linear, independent changes of different aspects of the speech signal.
- Voice transformation with a small number of required samples from the target speaker, making it possible to easily produce additional voices from an existing acoustic inventory, as opposed to recording an entire new inventory for the voice, which is time-consuming, tedious, and expensive. Example applications include systems for persons with voice disorders who use TTS synthesizers to communicate. Many such people can, with great effort, produce clear speech intermittently, which can then be used as training samples, ultimately rendering the output of their TTS system with their own voice.

In previous work, we have applied AIM to a diphone synthesizer, reducing the size of a 6.5 MB inventory to 57 kB (1:114 compression) at 8 dB spectral distortion, while eliminating concatenation errors [5]. In this paper, we extend our work to a unit-selection TTS synthesizer, which leads to new approaches during analysis and synthesis. Section 2 introduces the core ideas, as well as the general and implementation-specific forms of AIM. Sections 3 and 4 describe the analysis and synthesis of speech under the model. Section 5 evaluates the TTS system with respect to its intelligibility, quality, and speaker recognizability. Section 6 concludes the paper and discusses future directions.

2. The Asynchronous Interpolation Model

The core idea of AIM is to represent a short region (on the order of 5–10 ms) of speech as a *composition* of several types of features called *streams*. Each stream is computed by asynchronous interpolation of neighboring *basis vector* features. Each basis vector is associated (labeled) with a particular phoneme, allophone, or more specialized unit and may contain additional information about phonetic and prosodic context. Thus, the speech region is described by the varying degrees of influence of several types of preceding and following acoustic features. In this section, we extend and improve upon the notation reported previously [6].

Representing speech as an interpolation between vectors has been researched before; for example, the temporal decomposition approach [7, 8] decomposes speech into arbitrary event targets that describe successive events. Our method stands apart in that the phonemic identities of the basis vectors are known, and asynchronous interpolations are carried out on several streams consisting of different types of features.

2.1. General Form

Given a speech waveform, let the complex spectrum $\mathbf{X}$ at frame m be equal to a composition operation $\mathcal{C}$ on the values of N streams $\mathbf{s}$ at that frame

$$\mathbf{X}[m] = \mathcal{C}(\mathbf{s}_1[m], \ldots, \mathbf{s}_N[m]) \tag{1}$$

where different streams represent different types of feature trajectories. An individual stream is calculated by the interpolation

$$\mathbf{s}_n[m] = \sum_{k=1}^{K} w_n^{U_k}[m] \cdot \mathbf{b}_n^{u_k} \tag{2}$$

where $\mathbf{b}_n^{u_k}$ are the *basis vectors* associated with stream n and acoustic event u_k, and $w_n^{U_k}[m]$ are the *transition weights* at frame m that are associated with stream n and context

$$U_k = u_{k-l}, u_{k-l+1}, \ldots, u_{k-1}, u_k, u_{k+1}, \ldots, u_{k+r-1}, u_{k+r}$$

that includes the *l* previous and *r* following acoustic events. The summation is performed over K acoustic events. In addition, for a given frame m and stream n, transition weights are constrained by

$$\sum_{k=1}^{K} w_n^{U_k}[m] = 1 \tag{3}$$

to ensure a convex operation. When choosing speech features, care must be taken that they are "interpolatable" so that stream values are valid in a physical sense at all times; for example, formant parameters are interpolatable, but polynomial filter coefficients are not.

2.2. Implementation

In our specific implementation, we reduced phonetic and prosodic context by constraining the summation of Equation 2 to only depend on the previous and the next unit; in other words, the influence of a basis vector never extends beyond its neighbor. We chose two types of features, namely formant frequency locations and the formant-normalized complex spectrum. The latter is the result of modifying the complex spectrum so that formants appear at constant neutral values, allowing the interpolation of spectra without adding extraneous formants. Therefore, Equation 1 becomes

$$\mathbf{X}[m] = \mathcal{C}(\mathbf{s}_s[m], \mathbf{s}_f[m]) \tag{4}$$

where the subscripts refer to the association with spectral and formant information, respectively. The composition operator $\mathcal{C}$ was implemented as a non-linear warping of the formant-normalized spectral feature stream to obtain a spectrum with formants at the locations specified by the formant stream (more on this in Section 2.2.2).

The reduced context allows combining Equations 2 and 3, resulting in

$$\begin{aligned} \mathbf{s}_s[m] &= w_s^{u_l \to u_r}[m] \cdot \mathbf{b}_s^{u_l} + (1 - w_s^{u_l \to u_r}[m]) \cdot \mathbf{b}_s^{u_r} \\ \mathbf{s}_f[m] &= w_f^{u_l \to u_r}[m] \cdot \mathbf{b}_f^{u_l} + (1 - w_f^{u_l \to u_r}[m]) \cdot \mathbf{b}_f^{u_r} \end{aligned} \tag{5}$$

where u_l and u_r are acoustic events left and right of frame m, and m varies from the frame associated with event u_l to the frame associated with u_r.

Our choice of features was guided by the observation that in transitions between most phonemes, formant frequencies and the overall spectral shape change asynchronously (although this instance of the model makes the simplifying assumption that the formants themselves are synchronous). For example, a transition from /i:/ to /v/, as in the word "leave", shows a change in formants that starts well before the onset of frication. Another view is to regard the resulting system as an equivalent to image morphing, where salient features are used to mark important regions of two still images, and transitions are created by smoothly moving the salient features while modifying the underlying still images appropriately. In our case we used formants as salient features to render a good approximation of the transition between two sounds, which could not be achieved by a simple cross-fade.

2.2.1. Basis Vector Labeling

We selected basis vector label names similar to the Worldbet [9] phonetic labels for American English. Since basis vectors represent single acoustic events, some phonemes needed to contain several basis vectors. Specifically, diphthongs contained two separate basis vectors for the two different targets (/aI/: "aI1", "aI2"), voiced plosives contained two basis vectors for closure and burst (/b/: "bc", "b"), and unvoiced plosives contained three basis vectors for closure, burst, and aspiration (/t/: "tc", "tb", "th"). Finally, we represent affricates as a combination of other basis vectors (/tS/: "tc", "tb", "S").

Two different basis vector occurrences with the same label in the acoustic inventory can be treated as identical or distinct. This gave rise to the following three labeling schemes:

Global In the global labeling scheme all basis vectors with the same label were shared, resulting in typically less than 60 basis vectors. This lead to the smallest representation of the acoustic inventory and thus also gave the highest compression rate.

Local The opposite of the global scheme, the local scheme considers every basis vector in the inventory as unique. Special care must be taken during synthesis when concatenating two units with distinct basis vectors at the cutpoint to ensure smoothness (see Section 4). This scheme still provided a high rate of compression because the majority of frames are within transitions and are represented by transition weights only.

Automatic This scheme allowed the selection of an arbitrary size or quality criterion (as specified by an objective function) on the continuum spanned by the two previous schemes. This was implemented either by growing the global scheme and iteratively splitting and reassigning shared basis vectors, or by pruning the local scheme and iteratively merging two unique basis vectors.

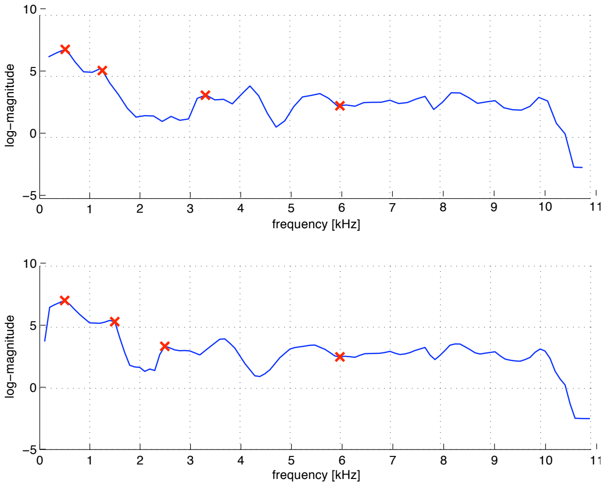


Figure 1: The effect of the composition operation. Given a log-magnitude spectrum of the phoneme /l/ with original frequency locations (top), the composition operation creates a non-uniformly resampled version to align with the desired frequency locations (bottom). Formant frequencies F1,F2, and F3, as well as the modification-cutoff frequency are located at the markers.

2.2.2. *Composition Operation*

The task of the composition operation is to receive a vector of stream values and to then render a short segment of speech. In our case the inputs are formant-normalized complex spectra and formant frequency values, and the composition consists of returning a modified complex spectrum with the neutral formant frequency locations changed to the specified ones.

Modifying formant frequencies in the natural spectrum has been previously researched [10, 11, 12]. Our implementation consists of non-uniformly resampling the original spectrum (see Figure 1). In addition to formant frequencies, we specify a modification-cutoff frequency at 6000 Hz to stop modification of the spectrum at and above that frequency. Conversely, the formant-normalized spectra themselves were initially created by modifying the original spectrum with associated original formant frequency locations to have formants at a constant neutral location.

3. Analysis

During analysis, synthesis, and evaluation, the system utilizes a small unit-selection database of a female speaker “AS” [13], which covers all diphones and specific triphones that are known to have a significant amount of coarticulation, but which does not have complete prosodic coverage.

3.1. Basis Vectors

In the proposed implementation, basis vectors contain information about both the complex spectrum and formant frequency locations. Therefore, the analysis process begins by making initial estimates of formant frequency trajectories F1, F2, and F3, using the ESPS get_formant algorithm [14].

The locations of basis vectors relative to phoneme boundaries are initialized as follows: When the phoneme contains just one basis vector, its location is set to that point which will, on average, result in the smallest concatenation error. For phonemes with two or more targets simple heuristics are employed, such as assigning the second basis vector at the 80% point of the total duration of a diphthong.

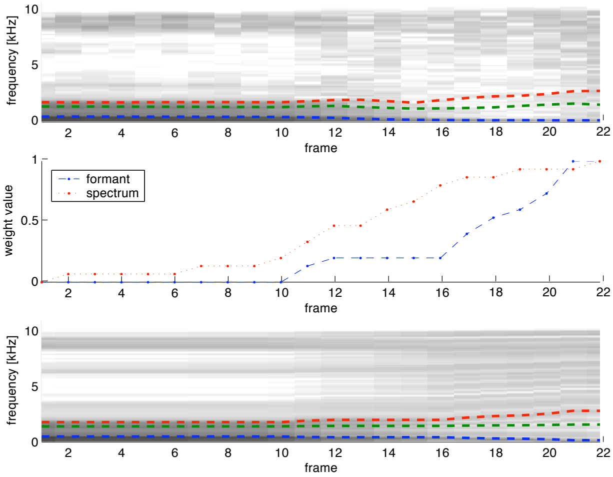


Figure 2: Transition weight analysis. The top panel shows the original log-magnitude spectrogram for the transition between /9r/ and /v/, with the original formant frequency trajectories superimposed. The middle panel shows the resulting weights after analysis. The asynchronous nature of the weights is easily observable. The bottom panel shows a resynthesis of the transition using the previously analyzed basis vectors and transition weights.

Both basis vector locations and formant frequency trajectories were manually corrected using a standard labeling tool in conjunction with a pen input device. This proved especially necessary in the following two cases: (1) to fine-tune the location of basis vectors during stops, affricates, and diphthongs, and (2) to create appropriate formant frequency locations in regions in which formants were not clearly visible, such as during a closure preceding a stop. In the latter case, formant frequencies were assigned in accordance with locus theory [15].

To extract complex spectra, we perform a pitch-synchronous sinusoidal analysis over two frames nearest to the basis vector location and store the magnitude and phase of each harmonic sinusoid, as well as fundamental frequency and voicing information of the analysis frame.

3.2. Transition Weights

For each transition, we fit the transition weights by first assuming a straight-line transition $w = 0, 1/Q, \ldots, (Q-1)/Q$, where Q represents the weight value resolution; for example, we use $Q = 16 = 2^4$ which allows weights to be stored in 4 bits. Then, the formant-normalized magnitude spectral stream and formant frequency stream are constructed using local basis vectors and the straight-line weights. (The phase spectrum is ignored during fitting.) Finally, the streams are separately aligned to the original formant and spectral transitions using a dynamic time warping (DTW) algorithm (see Figure 2). In cases where original formant trajectories are unavailable, a joint DTW can be used [6].

The DTW algorithm has local constraints that insure monotonically increasing transition weights. There are no global constraints and the local constraints allow for maximally discontinuous changes in the weights from one frame to the next. This is

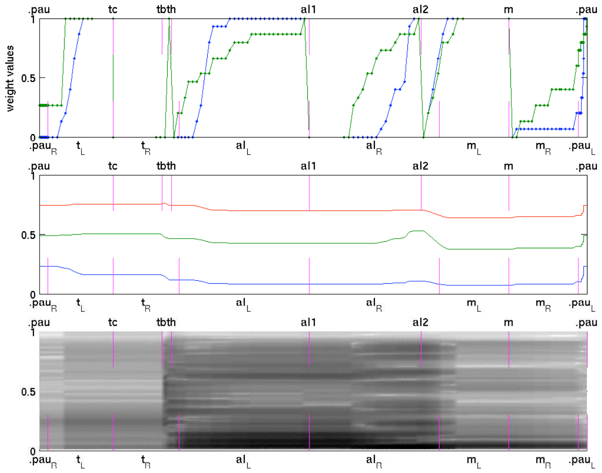


Figure 3: Basis vectors and transition weights used for synthetic utterance (top panel). Basis vectors "tc", "aI1", and "m" are surrounded by weight values of ones (on their left) or zeros (on their right), respectively, due to synthetic lengthening of the units. The formant and spectral streams are displayed as trajectories (middle panel) and the formant-normalized log-magnitude spectrogram (bottom panel). Lines at the tops of panels mark the position and identity of a basis vector, whereas lines at the bottoms of panels denote the diphone boundaries, their identities labeled at the center.

needed because many transitions are quite abrupt (for example, nasal to vowel transitions).

Transition weight trajectories could be further regularized by replacing them with parametric functions, for example a sigmoidal function. Moreover, weight trajectories of certain classes of similar transitions (for example vowel to nasal transitions) could be tied to a single model. Both of these optional steps would yield additional storage savings.

4. Synthesis

The compressed acoustic inventory consists of two 4-bit weights, $w_s^{u_l \rightarrow u_r}[m]$ and $w_f^{u_l \rightarrow u_r}[m]$, for each speech frame, and an associated basis vector time and identity list that references the collection of basis vectors $\mathbf{b}_s^u$ and $\mathbf{b}_f^u$, in addition to the traditional list of units that are used during the acoustic inventory search. During synthesis, we first construct basis vectors and transition weights for the synthetic utterance, by assigning weights, basis vector locations and basis vector identities according to the output of the unit search and the specified synthetic durations. When synthetic durations are shorter than the original units, the unit is shortened by compressing the times at which weights and basis vectors occur. When synthetic durations are longer than the original units, we leave the original weight trajectories unmodified, but instead shift the weights left when we are synthesizing the left side of a phoneme, and shift the weights right for the right side of a phoneme. The resulting effect is that phonemes are lengthened at their centers during stretching, but that the transitions themselves are at their original speeds (see Figure 3).

When two non-identical basis vectors fall onto the same point in time, we merge the basis vectors by taking their average; thus creating a new, temporary basis vector (by definition, streams are interpolatable). This situation occurs when the local or automatic labeling scheme is used and we are concatenating across two basis vectors associated with the same phoneme, but from two different contexts.

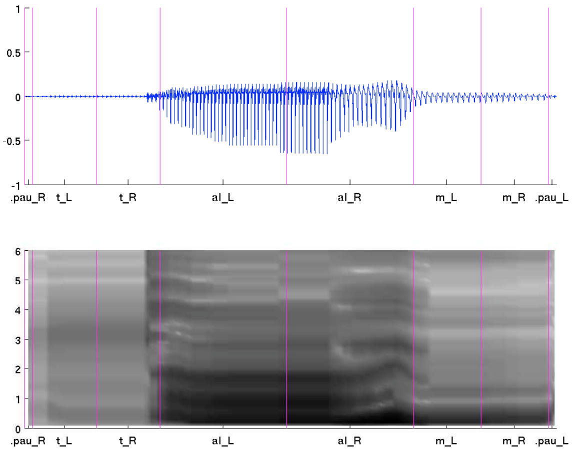


Figure 4: Synthetic waveform (top panel) and pitch-synchronous log-magnitude spectrogram (bottom panel) with diphone boundary lines.

After constructing basis vectors and transition weights, Equations 5 and then 4 are used to calculate the complex spectrogram of the synthetic utterance, which is finally rendered as a waveform by a pitch-synchronous sinusoidal synthesis algorithm (see Figure 4).

AIM also allows a new approach to the spectral aspect of voice transformation, by regarding basis vectors as speaker-dependent, but transition weights as speaker-independent. Using the global labeling scheme described in Section 2.2.1, we estimated a small number of basis vectors for several new target speakers. Then, transformed speech is produced by using the original speaker's transition weights with the desired target speaker's basis vectors.

5. Evaluation

5.1. Intelligibility and Quality

The following four conditions were compared: (1) the standard OGI TTS baseline system [13] at 352.8 kbps, (2) the baseline compressed with the Speex CELP coder [16] at 8.0 kbps, (3) the baseline compressed with the Speex CELP coder at 3.4 kbps, and (4) the BioSpeech AIM TTS system using the global labeling scheme at 3.4 kbps. The average bit rate for AIM was computed as follows: Given 54 basis vectors with an average dimension of 62, where each component is represented by 16 bits, yields 53,568 bits. Each of the 63,716 frames of the acoustic inventory contains an 8-bit number that marks the position of the frame; in addition, each frame contains two 4-bit transition weights, for a total of 1,019,456 bits. Finally, the 132,300-bit wave library is added, for a grand total of representing the database in 1,205,324 bits or 3,414 bps. Compared to the original representation of 124,530,928 bits, or 352.8 kbps, this represents a 103:1 compression rate.

The text material used in these experiments consisted of 48 sentences, randomly selected from the IEEE Harvard Psychoacoustic Sentences [17], containing five keywords each (e. g. "His <u>shirt</u> was <u>clean</u> but <u>one</u> <u>button</u> was <u>gone</u>"). Each sentence

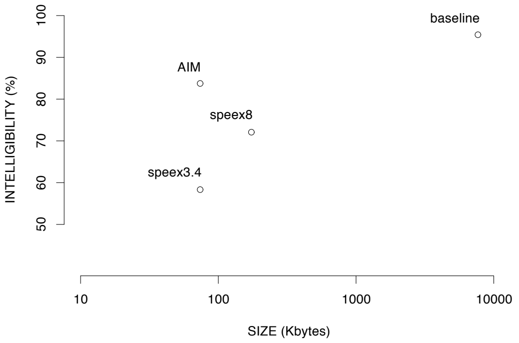


Figure 5: Word intelligibility, defined as the percentage of keywords correctly repeated per sentence.

was synthesized in each of the four conditions.

Six listeners aged 24–35 participated, all native speakers of English and unfamiliar with the goals of the study. Listeners heard an utterance exactly once, attempted to repeat the utterance, and then rated its speech quality on a 1–5 Mean Opinion Score (MOS) scale ("bad", "poor", "fair", "good", "excellent"). A test administrator scored the number of key words that were repeated correctly, while the rating was recorded automatically. The test was designed so that condition and presentation order were uncorrelated; therefore any effects due to condition cannot be attributed to some conditions being presented relatively late (or early) in the experiment.

Figures 5 and 6 show the results for word intelligibility (I_W), defined as the percentage of keywords correctly repeated per sentence, and sentence Intelligibility (I_S), defined as the percentage of sentences correctly repeated in their entirety. Figure 7 shows quality (Q) represented by the mean opinion score, averaged over all listeners and all sentences in that particular condition. Statistical tests (planned t-tests) indicated that AIM was significantly superior in intelligibility and quality to the size-matched 3.4 kbps coder condition (I_W: $p < 0.005$; I_S: $p < 0.015$; Q: $p < 0.001$). AIM was also superior in both ways to the larger 8 kbps coder condition, but this was significant only for quality (Q: $p < 0.001$).

5.2. Speaker Recognizability

In this test, a source speaker's basis vectors of an acoustic inventory were replaced with basis vectors from a target speaker's acoustic inventory, while leaving the transition weights unchanged. Prosody was kept exactly constant for all stimuli to ensure that speaker recognizability performance was measured based on spectral cues only, and not on prosodic cues.

The text material used in this experiment consisted of 40 sentences, randomly selected from the IEEE Harvard Psychoacoustic Sentences [17]. The sentences were synthesized using AIM with representations derived from the acoustic inventory of five male voices, aged 21–39, and whose native language was American English. The local labeling scheme was used for highest synthesis quality. For 20 of the sentences, the original basis vectors were replaced by basis vectors derived from exactly one of the other four voices.

A speaker recognizability test was chosen to evaluate voice transformation performance [18]. During testing, six listeners (the same as in Section 5.1) heard two utterances in sequence. One of them was the voice transformation condition described above, and the other was the normally synthesized version of a different sentence of either the same speaker or a different speaker. The task was to decide whether the two utterances were from the same speaker or from two different speakers. The response alternatives were "definitely different", "kind of different", "unsure", "kind of same", "definitely same", and were recorded automatically. Note the equal number (20) of correct "same" and "different" responses.

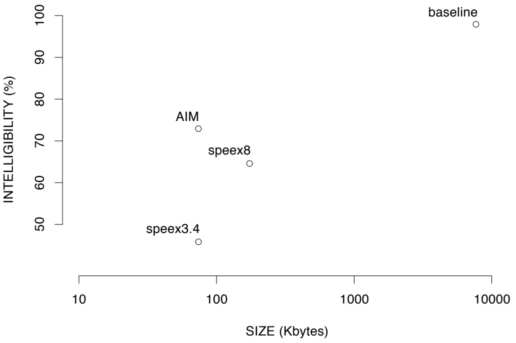


Figure 6: Sentence intelligibility, defined as the percentage of sentences correctly repeated in their entirety.

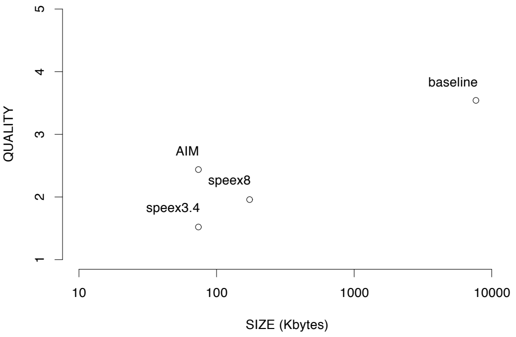


Figure 7: Mean Opinion Score (MOS) of speech quality, on a 1–5 scale ("bad", "poor", "fair", "good", "excellent").

All six listeners had higher percentages of matching speakers when they indicated "same" compared to "different", significant at $p < 0.025$ using a 1-tailed Sign test. Except for one listener, all speakers showed a completely monotonically decreasing response pattern, as shown in Figure 8. Four out of six listeners recognized speakers as being the same 100% correctly when they were certain of the speakers being the same. Conversely, three out of four speakers recognized speakers as different 100% correctly when they were certain of their choice. This indicates that, even when no prosodic cues are available and different sentences are presented, the AIM method preserves adequate speaker information to enable listeners to determine speaker identity.

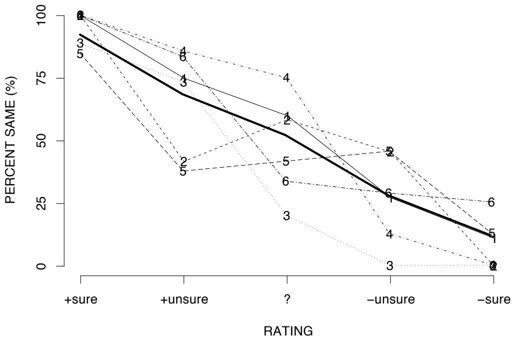


Figure 8: Speaker Recognizability, represented by percentages of items where the two speakers of the stimuli presented are the same, as a function of listener rating. "+" and "-" indicate same and different, respectively. For example, “-sure” refers to “definitely different”. Listener numbers are shown on the curves; the heavy curve represents the mean over all listeners.

6. Conclusion

We have described a speech synthesis system based on the Asynchronous Interpolation Model, which represents speech as a composition of several streams that are computed using asynchronous interpolation of neighboring basis vectors. Applied to a concatenative TTS system’s acoustic inventory, the model avoids concatenation errors during synthesis, and affords opportunities for variable compression and a new approach to voice transformation. During evaluation, AIM produced significantly higher quality and intelligibility than speech that has been compressed by traditional methods, using sizes equal to AIM or more than twice as large as AIM. The AIM compression ratio in this study was 103:1; this could easily be further increased by further parametrization of transition weights. Results also showed that AIM produces speech that can be reliably identified with a desired target speaker, using an extremely small set of training speech.

Further enhancements are necessary to increase intelligibility and quality scores. One of these would be a more sophisticated method of formant manipulation, which currently was implemented using a simple frequency warping. Another enhancement would be to model the deterministic and stochastic part of a speech frame separately, allowing for higher quality modeling of noise when a single spectral basis vector is repeated several times throughout a transition. Finally, we plan on investigating approaches that will automatically insert additional basis vectors, thus enabling a complete reconstruction of the original acoustic inventory in the limit.

7. Acknowledgments

Part of this research was funded by United States National Science Foundation STTR grant 0441125. Oregon Health & Science University, Dr. Kain, and Dr. van Santen have a significant financial interest in BioSpeech, Inc., a company that may have a commercial interest in the results of this research and technology. This potential conflict was reviewed and a management plan approved by the Conflict of Interest in Research Committee and the Integrity Program Oversight Council was implemented.

8. References

[1] H. Mizuno, M. Abe, and T. Hirowaka, “Waveform-based speech synthesis approach with a formant frequency modification,” in *ICASSP*, 1993, pp. 195–198.

[2] J. Wouters and M. Macon, “Control of spectral dynamics in concatenative speech synthesis,” *IEEE Trans. Speech and Audio Proc.*, vol. 9, no. 1, pp. 30–38, Jan. 2001.

[3] D. T. Chappell and J. H. L. Hansen, “A comparison of spectral smoothing methods for segment concatenation based speech synthesis,” *Speech Communication*, vol. 36, no. 3, pp. 343–373, 2002.

[4] P. H. Low, C. H. Ho, and S. Yaseghi, “Using estimated formant tracks for formant smoothing in text to speech synthesis,” in *ASRU*, 2003, pp. 688–693.

[5] A. Kain and J. van Santen, “Compression of acoustic inventories using asynchronous interpolation,” in *IEEE Workshop on Speech Synthesis*, 2002, pp. 83–86.

[6] A. Kain and J. van Santen, “A speech model of acoustic inventories based on asynchronous interpolation,” in *EUROSPEECH*, 2003, pp. 329–332.

[7] B. Atal, “Efficient coding for LPC parameters by temporal decomposition,” in *ICASSP*, 1983, pp. 81–84.

[8] S. Ghaemmaghami, M. Deriche, and B. Boashash, “Comparative study of different parameters for temporal decomposition based speech coding,” in *ICASSP*, 1997.

[9] J. Hieronymus, “ASCII phonetic symbols for the world’s languages: Worldbet,” Tech. Rep., Bell Labs, 1993.

[10] Y.-S. Hsiao and D.G. Childers, “A new approach to formant estimation and modification based on pole interaction,” in *Thirtieth asilomar conference on signals, systems and computers*, 1996, vol. 1, pp. 783–787.

[11] R. W. Morris and M. A. Clements, “Modification of formants in the line spectrum domain,” *IEEE Signal Processing Letters*, vol. 9, pp. 19–21, Jan. 2002.

[12] E. Turajlic, D. Rentzos, S. Vaseghi, and C.-H. Ho, “Evaluation of methods for parameteric formant transformation in voice conversion,” in *ICASSP*, 2003, pp. 724–727.

[13] M. Macon, A. Cronk, J. Wouters, and A. Kain, “OGIresLPC: Diphone synthesizer using residual-excited linear prediction,” Tech. Rep. CSE-97-007, Dept. of Computer Science, Oregon Graduate Institute of Science and Technology, Portland, OR, Sept. 1997.

[14] Entropic Research Laboratory, “Entropic Signal Processing System (ESPS) Waves+,” Software, Aug. 1993.

[15] D. Broad and F. Clermont, “A methodology for modeling vowel formant contours in CVC context,” *JASA*, vol. 81, no. 1, pp. 155–165, Jan. 1987.

[16] J.-M. Valin, “Speex: A Free Codec For Free Speech,” www.speex.org, 2006.

[17] E. H. Rothauser, W. D. Chapman, N. Guttman, K. S. Nordby, H. R. Silberger, G. E. Urbanek, and M. Weinstock, “IEEE Recommended practice for speech quality measurements,” *IEEE Trans. on Audio Electroacoustics*, vol. 17, pp. 227–246, 1969.

[18] A. Kain, *High Resolution Voice Transformation*, Ph.D. thesis, OGI School of Science & Engineering at Oregon Health & Science University, 2001.

Modelling Voiceless Speech Segments by Means of an Additive Procedure Based on the Computation of Formant Sinusoids

Ingo Hertrich, Hermann Ackermann

Department of General Neurology, Hertie Institute for Clinical Brain Research,
University of Tübingen, Germany

ingo.hertrich@uni-tuebingen.de

Abstract

A previously developed vowel synthesis algorithm implements formants as sinusoids, amplitude- and phase-modulated by the fundamental frequency (Hertrich and Ackermann, 1999, Journal of the Acoustical Society of America, 106, 2988-2990). The present study extends this approach to the modelling of the acoustic characteristics of aperiodic speech segments. To these ends, a voiceless signal component is generated by adding at each sample point a random parameter onto the formants' phase progression. Voiceless stop consonants then can be modelled, e.g., by combining a release burst, i.e., an interval in which the formant sinusoids abruptly increase and gradually decrease in amplitude, with formant-shaped noise components, representing inter-articulator frication, aspiration, and breathy vowel onset.

1. Introduction

Most speech synthesizers use a source-filter model in order to generate acoustic output signals. As a rule, the vocal tract filter characteristics are either derived from articulation-based parameters or specified in terms of a formant structure while the source, in case of voiced speech segments, approximates the laryngeal excitation signal [1]. A different approach is used in sinusoidal coding techniques [2, 3, 4, 5]: Each fundamental period, i.e., the time domain of a single laryngeal cycle, can be approximated by summing up a set of partial "formant-wave-functions" [3], corresponding to the eigenfrequencies of the vocal tract during this period. At least some of the remaining aspects of the speech signal such as formant bandwidth and voice quality can be implemented as modulations of the amplitude envelopes of these waveforms, characterized by, e.g., by an initial attack interval followed by a decay function within each fundamental period. Thus, formants can be modeled as sinusoids, implementing fundamental frequency as an amplitude- and phase modulation of the formants. In comparison to the natural mechanisms of speech production as well as most speech synthesizers, this approach reverses the source-filter hierarchy and, thus, might be considered an artificial construct. However, amplitude-, phase- and frequency-modulated sinusoids provide the opportunity for a more explicit control of formant structure at a high temporal resolution and at a high computational precision and, therefore, can be used to produce well-defined speech-like stimuli for the purpose of listening experiments. For example, a recent magnetoencephalographic study on the auditory processing of voiced stop consonants used this method to create stimuli that exclusively differed in the duration of syllable-initial formant transitions [6]. As a further example, an investigation of dichotic listening effects compared the perception of natural /ba/ and /da/ syllables to synthetic cognates that exclusively differed in their syllable-initial formant transitions [7]. Apart from psychoacoustic research, formant waveform synthesis may also contribute to an extension of speech synthesis applications with respect to some dynamic aspects of speech such as the continuously changing formant structure, in stop consonant release transients [8], characterized by a time-varying formant structure following a single excitation burst.

As compared to vowels, consonants may exhibit a more complex vocal tract resonance structure due to the engagement of different sound sources and a compartmentalization, more or less, of the vocal tract [9]. While resonance functions following impulse-like excitations can easily be created by sinusoidal formant wave functions, the formant-based generation of voiceless segments seems to be more difficult. One possibility of handling aperiodic segments is the use of multiple overlapping formant waves with irregularly-timed temporal onset [10]. Alternatively, voiceless formant waveforms may be derived from continuous narrowband-sinusoids lacking any decay function. This approach gives rise to unnaturally-sounding whistling-like sounds that, however, still can be perceived as speech ("sinusoidal speech") under some circumstances [11]. In order to simulate a noise-like source, these sinusoids can be manipulated by adding a random factor on their sample-to-sample phase progression [12], resulting in an increase of bandwidth. In fact, this procedure manipulates the instantaneous frequency of the formant waveform while the amplitude and the center frequency can be kept constant. In fact, fricative consonants such as /s/ are often characterized by high and sharp resonance frequencies due to the presence of small cavities near the sound source, giving rise to whistling-like phenomena [13], which can easily be modelled by this kind of synthesis.

The present study represents an extension of the formant-waveform-based speech synthesis algorithm introduced by Hertrich and Ackermann [5]. 'Voiceless' signal components are realized by random phase perturbation of the formant waves. If the perturbation is set to zero, the formants of the aperiodic signal component correspond to pure sine waves. In case of small perturbations, the formants remain visible in the spectrogram, and spectral density near the formant frequencies is relatively high. Increasing the random component ultimately results in broadband noise.

2. The Algorithm

The acoustic target characteristics of the signal to be synthesized are specified in an ASCII input file. Each line of this text file contains a set of 18 parameters referring to duration, intensity, voicing characteristics, fundamental frequency (F0), and five formant frequencies (F1 to F5) as well as the relative formant amplitudes.

Segment duration (L) represents a time interval of linear changes with respect to the remaining parameters from the current input line to the respective parameter values of the following line (the duration parameter of the final line is not evaluated). These parameters include: voiced (A_v) and voiceless (A_n) signal amplitude, relative amount of phase distortion per sample point for the voiceless formant sinusoids (P_n), relative duration of the rise (V_r) and the stationary phase (V_s) of the amplitude profile during one pitch period of the voiced signal component, fundamental frequency (F0) and its relative amplitude (a_{F0}), and five formant frequencies (F1 - F5) as well as their relative amplitudes in percent of total signal amplitude (a_{F1} - a_{F5}).

As a first step, as in [5], signal portions are also synthesized as sequences of pitch periods, voiced signal amplitude being set to a low value or zero, and the formant-related parameters are interpolated with respect to their values at the begin and the end of the respective pitch period. The second step performs period-by-period synthesis of the acoustic signal according to these specifications.

With respect to the voiced signal component, the algorithm introduced in [5] has been modified in order to provide the possibility to handle the amplitude profile A(t) of the formants within single pitch periods in a more flexible way. To these ends, the formants' amplitude envelope within each pitch period is subdivided into a (linear) rising interval, a steady-state portion, and a final decay phase toward zero at the end of the respective pitch period (t represents time from beginning of the current pitch period):

$$A(t) = A_v \cdot \frac{t}{To \cdot V_r} \qquad \text{for the rise phase,}$$

$$A(t) = A_v \qquad \text{for the steady-state phase, and}$$

$$A(t) = A_v \cdot \frac{To - t}{To \cdot (1 - V_r - V_s)} \qquad \text{for the decay phase.}$$

The voiced signal component of each pitch period, then, is the sum of all formant sinusoids modulated in the following way:

$$y_v(t) = A(t) \cdot \left[a_{F0} \cdot \sin\left(2\pi \cdot \frac{t}{To}\right) + \sum_{i=1}^{5} a_{Fi} \cdot \sin \varphi_{Fi}(t) \right],$$

where a_{F0} is the relative amplitude of the fundamental frequency, a_{Fi} is the relative amplitude of the i^{th} formant, and the phase angles

$$\varphi_{Fi}(t) = \varphi_{Fi}(t - dt) + 2\pi \cdot F_i(t) \cdot dt$$

are computed incrementally [$\varphi_{Fi}(0) = 0$] for each sample point using the instantaneous formant frequencies

$$F_i(t) = F_i(0) + \frac{t}{To} \cdot [F_i(To) - F_i(0)]$$

(dt is the duration between two successive signal samples).

In contrast to the voiced part, the voiceless signal component does neither exhibit pitch-induced amplitude modulation nor a phase reset at the beginning of each pitch period. Each formant is just represented by a sinusoid of a given amplitude

$$y_n(t) = A_n(t) \cdot \sum_{i=1}^{5} a_{Fi} \cdot \sin \varphi_{Fi}(t),$$

where the formants' phase angles are derived by, first, considering (as in the voiced signal component) the instantaneous formant frequencies $F_i(t)$ and, additionally, a random increment the magnitude of which may also vary in time:

$$\varphi_{Fi}(t) = \varphi_{Fi}(t - dt) + 2\pi \cdot F_i(t) \cdot dt + P_n(t) \cdot rnd,$$

rnd being a random number in the range +/- 2π and

$$P_n(t) = P_n(0) + \frac{t}{To} \cdot [P_n(To) - P_n(0)]$$

representing the local value of the phase distortion parameter. For each sample point, then, voiced and unvoiced signal amplitude are added:

$$y(t) = y_v(t) + y_n(t).$$

3. Examples and Comments

3.1. Single-formant Test Signal

In order to demonstrate the working principle of the algorithm, a single-formant test signal was generated, characterized by five phases (0.2 s each) with the following characteristics (Figure 1):

(1) The signal starts with a voiced interval, the formant moving from 500 to 1500 Hz, F0 changing from 250 to 100 Hz, and amplitude decreasing to zero at 0.2 s.

(2) A voiceless segment starts with gradually increasing amplitude, constant formant frequency at 1500 Hz, and

constant amount of phase distortion, resulting in spectral dispersion.
(3) The formant starts moving back to 500 Hz, phase distortion and amplitude being kept constant.
(4) The phase distortion decreases to zero, i.e., the formant approximates a pure sinusoid at 0.8 s.
(5) The pure formant sinusoid rises from 500 to 1500 Hz.

These five phases (segments of a duration of 80 ms each) are exemplified in Figure 2.

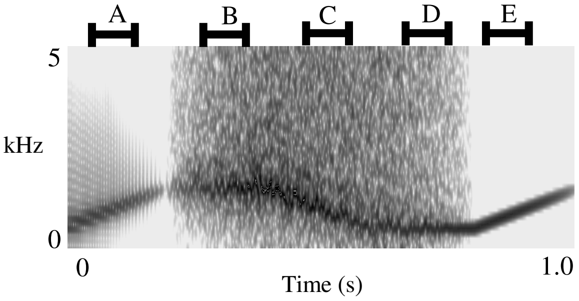


Figure 1. Spectrogram of a single-formant test signal, generated to demonstrate the working principles of the synthesis algorithm. The waveforms of the five 80-ms segments marked with capital letters are displayed in Figure 2.

3.2. Synthesis of a voiceless stop consonant-vowel syllable

In principle, aspirated stops may encompass five acoustic events, (1) a silent occlusion interval, (2) the initial plosion burst, (3) a short interval of inter-articulator frication, (4) aspiration noise, and (5) vowel onset. To some degree, these intervals may overlap, eventually giving rise to spirantization, multiple initial bursts, inter-articulator frication synchronous with aspiration noise, and/or a breathy or harsh voice quality during the initial part of the vowel. As an example, the syllable /ka/ (Figure 3) was synthesized using the parameter specifications listed in Table 1. The relevant phonetic characteristics of this signal were composed in the following way: The initial plosion and the amplitude decrease following the burst is represented in the first interval (20 ms, lines 1-2 in Table 1), modeled as a superimposition of phase-distorted (A_n, P_n) formant sinusoids and undistorted formant sinusoids declining in amplitude (A_v,V_r, V_s). The latter component represents an impulse-like event followed by a vocal tract filter response (see Repp and Lin, 1989) and, thus, can be realized using the algorithm for the synthesis of voiced portions of speech (F0 was set to 50 Hz to obtain a single resonance period within the 20 ms only). The second interval specified in Table 1 (30 ms) mainly models the aspiration phase, exhibiting aperiodic noise with a lower center of gravity than the stop burst. The first formant increases in amplitude (a_{F1}, lines 2-3) while the higher formants undergo attenuation. Note that the first three formant frequencies show a continuous transition typical for velar articulation preceding the vowel /a/ during the initial two intervals (rising F1 and F3, falling F2). The following part of Table 1 (lines 3-6) specifies the vowel part of the syllable, characterized by high voiced signal amplitude (A_v), pitch (120 Hz at vowel onset, declining to 90 Hz), formant frequencies of the vowel /a/, and the largest relative amplitude in the first formant. The initial part of the vowel (lines 3-4) is characterized by an increase of voiced (A_v) and a decrease of voiceless (A_n) amplitude, accounting for declining breathiness during vowel onset. Furthermore, the relative amplitude of the fundamental frequency decreases while the formants are amplified toward the center of the vowel. The offset of the vowel (lines 5-6) shows a drop in voiced intensity, a slight onset of voicelessness, a drop in amplitude of the higher formants, and a change toward a less skewed intensity profile (V_r) within each pitch period.

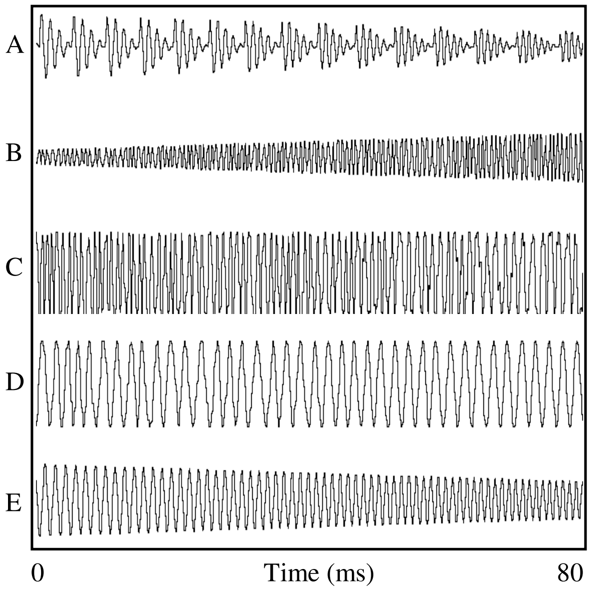


Figure 2. Oscillograms of five selected 80-ms intervals of the test signal displayed as a spectrogram in Figure 1 (capital letters on top of this figure).
A) This segment shows a combination of falling pitch (increase of the fundamental period from left to right), rising formant frequency (oscillations within each period), and decrease in amplitude.
B) Voiceless segment with increasing amplitude, constant effective formant frequency and irregularly-timed peaks indicating the random variation of the formant's instantaneous frequency.
C) Voiceless segment with constant amplitude and increasing period duration (i.e., falling formant frequency. Note that, in contrast to a pure sinusoid, the shape of the waveform and the spacing of peak-to-peak intervals is characterized by some irregularity.
D) During this segment, the random factor upon the formant's instantaneous frequency continuously decreases, resulting in increasingly regular peak-to-peak intervals from left to right.
E) Sinusoid with decreasing amplitude and rising frequency.

Table 1. Input parameters used for synthesis of the syllable /ka/ displayed in Figure 3

L (ms)	A_v	A_n	P_n	V_r	V_s	F0 (Hz)	a_{F0}	F1	a_{F1}	F2	a_{F2}	F3	a_{F3}	F4	a_{F4}	F5	a_{F5}
20	5000	2000	.1	.05	.05	50	0	300	.1	1800	.1	1800	.15	3800	.1	4500	.08
35	0	3000	.15	.1	.2	29	.1	500	.1	1600	.15	2000	.12	3800	.08	4500	.05
40	5000	2000	.05	.04	.2	120	.2	800	.15	1240	.1	2300	.05	3800	.05	4500	.03
80	20000	500	.05	.04	.2	110	.1	800	.25	1240	.15	2300	.15	3800	.10	4500	.05
25	20000	0	.05	.04	.2	100	.1	800	.25	1240	.15	2300	.15	3800	.10	4500	.05
0	10000	300	.05	.2	.2	90	.1	800	.25	1240	.15	2300	.05	3800	.05	4500	.03

Abbreviations: A_v = voiced amplitude, A_n = unvoiced amplitude, P_n = phase distortion, L = segment duration, V_r = relative rising time within a pitch period, V_s = relativ duration of the steady-state phase of a pitch period, F0 = fundamental frequency, a_{f0} = relative amplitude of the fundamental frequency, F1 to F5 = formant frequencies, a_{F1} to a_{F5} = relative formant amplitudes (see text for further discussion of the various parameters).

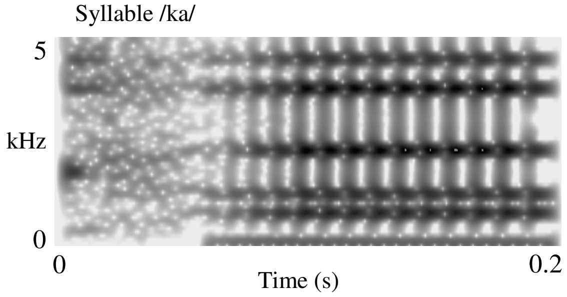


Figure 3. Spectrogram of the syllable /ka/ specified by the parameter settings given in Table 1 (see Text 3.2.). Note the formant transitions in the aspiration phase.

3.3. Comments

The current algorithm allows for the implementation of continuous changes across time of the following aspects of the acoustic signal: (1) voiced and voiceless signal amplitude, (2) formant frequencies and relative formant intensities, (3) magnitude of random phase distortion of the formants controlling spectral bandwidth of voiceless signal components, (4) and the amplitude profile within single pitch periods. So far, this approach does not provide different formant specifications for the voiced and voiceless components in case of mixed voiced/unvoiced signals. Furthermore, the parameter controlling phase distortion during voiceless segments has the same value across all formants. Although the algorithm at its current stage of elaboration seems to work quite well, a more detailed modelling of speech signals may require an increase in the number of input parameters. Considering the additive working principle of this procedure, such extensions can easily be implemented.

4. References

[1] Klatt, D.H., "Review of text-to-speech conversion for English." *J. Acoust. Soc. Amer., Vol. 82, 1987, pp 737-793.*

[2] McAulay, R.J., Quatieri, T.F., "Synthesis based on a sinusoidal representation." *IEEE-ASSP, Vol 34, 1986, pp. 744-754.*

[3] Rodet, X., "Time-domain formant-wavefunction synthesis". *Computer Music J., Vol. 8, 2007, pp. 9-14.*

[4] D'Alessandro, C., "Time-frequency speech transformation based on an elementary waveform representation." *Speech Comm., Vol. 9, 1990, pp. 419-431.*

[5] Hertrich, I., Ackermann, H., "A formant synthesizer based on formant sinusoids modulated by fundamental frequency." *J. Acoust. Soc. Amer., Vol. 106, 1999, pp. 2988-2990.*

[6] Hertrich, I., Mathiak, K., Lutzenberger, W., Ackermann, H., "Processing of dynamic aspects of speech and non-speech stimuli: a whole-head magnetoencephalography study". *Cogn. Brain Res., Vol. 17, 2003, pp. 130-139.*

[7] Hertrich, I., Mathiak, K., Lutzenberger, W., Ackermann, H., "Hemispheric lateralization of the processing of consonant-vowel syllables (formant transitions): effects of stimulus characteristics and attentional demands on evoked magnetic fields." *Neuropsychologia, Vol. 40, 2002, pp. 1902-1917.*

[8] Repp, B., Lin, H.-B., "Acoustic properties and perception of stop consonant release transients." *J. Acoust. Soc. Amer., Vol 85, 1989, pp. 379-396.*

[9] Holmes, J.N., "Formant synthesizers: cascade or parallel?" *Speech Comm., Vol. 2, 1983, pp. 251-273.*

[10] Richard, G., d'Alessandro, C., "Analysis/synthesis and modification of the speech aperiodic component." *Speech Comm., Vol. 19, 1996, pp. 221-244.*

[11] Remez, R.E., Rubin, P.E., Pisoni, D.B., Carrell, T.D., "Speech perception without traditional speech cues." *Sciene, Vol. 212, 1981, pp. 947-950.*

[12] Freed, A., "Spectral line broadening with transform domain additive synthesis." *Proc. Int. Computer Music Conf. Bejing, China, 1999*, http://cnmat.cnmat.berkeley.edu/ICMC99/papers/InverseNoise/InverseNoiseICMC.pdf.

[13] Shosted, R.K., "Just put your lips together and blow? Whistled fricatives in Southern Bantu." Yehia, H.C., Demolin, D., Laboissiere R. (Eds.), *Proceedings of ISSP 2006: 7th Int. Seminar on Speech Production*. CEFALA, Belo Horizonte, 2006, pp. 565-572.

Using Articulatory Position Data in Voice Transformation

Arthur R. Toth, Alan W Black

Language Technologies Institute, Carnegie Mellon University,
Pittsburgh, PA, USA

atoth@cs.cmu.edu, awb@cs.cmu.edu

Abstract

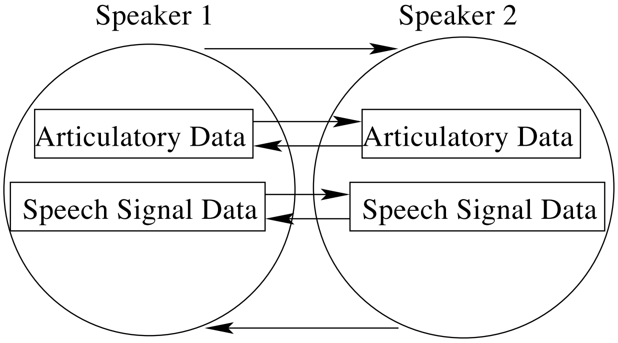
Articulatory position data is information about the location of various articulators in the vocal tract. One form of it has been made freely available in the MOCHA database [1]. This data is interesting in that it provides direct information on the production of speech, but there is the question of whether it actually provides information beyond what can be derived from the audio signal, which is much easier to collect. Although there has been some success in improving small-scale speech recognition and in demonstrating mappings between articulatory positions and spectral features of the audio signal, there are many problems to which this data has not been applied. This work investigates the possibility of using articulatory position data to improve voice transformation, which is the process of making speech from one person sound as if it had been spoken by another. After further investigation, it appears to be difficult to use articulatory position data to improve voice transformation using state-of-the-art voice transformation techniques as we only had a few positive results across a range of experiments. To achieve these results, it was necessary to modify our baseline voice transformation approach and/or consider features derived from the articulatory positions.

1. Introduction

Articulatory position data is information on the location of articulators during speech. The particular set investigated here is the freely available MOCHA database [1], which includes recordings of the 460-sentence British TIMIT corpus along with coordinates in the mid-sagittal plane for the upper and lower lip, the lower incisor, three points on the tongue, and the velum of each speaker. As this data provides direct information on the physical production of speech, there is hope that it can be used to improve models for speech. In many cases, current speech models are based on features derived from the audio signal through signal processing techniques such as LPC, cepstra, or mel-cepstral coefficients. Such features are arguably either more related to the perception of speech than the production of speech or represent an attempt to indirectly reconstruct information about production. Articulatory position data is exciting in that it gives direct information about production, but it is not without its limitations. One difficulty is that it may not fully represent the important parts of production. Seven points in a plane may not be sufficient to represent lateral effects, constrictions in the vocal tract, or the shape of the tongue. Information about pitch and power will not be directly represented. However, there may still be usable information even though the information is not complete, and there is evidence, at least for speech recognition, that it can help [2].

Another difficulty is that articulatory position data is hard to collect and this makes it fairly sparse. In most cases, it will probably not be collected during audio recordings. Thus, there is the additional question of whether this data can be useful in cases when it is available for a different speaker than the one who was recorded. There has been some work in this area as well [3]. In this context, it is natural to ask whether using articulatory position data can provide useful modeling information beyond what is available from the audio signal and for what tasks is it helpful.

Figure 1: *Transformation of Articulatory and Speech Data*



This paper attempts to extend the use of articulatory position data to voice transformation. Voice transformation is the process of making speech from one speaker sound as if it came from another. It is an important topic in speech synthesis, because successful voice transformation could greatly reduce the difficulty in producing synthetic voices with new identities and styles. Creating a concatenative speech synthesizer typically requires recording more than a thousand sentences for reasonable coverage of phonetic events. Coverage of different styles may require even more recordings. These recordings must be created for each speaker. Voice transformation has a much smaller incremental cost. After the first speaker is recorded, it is typical to record only an additional 20-30 sentences to create a new synthetic voice.

Researchers have investigated voice transformation for over 20 years and have explored many different techniques. The experiments in this paper are based on Gaussian Mixture Model (GMM) mapping techniques. These models were used at least as early as the mid-1990s [4], have been refined since then [5] [6] [7] [8], and are still considered state-of-the-art. Furthermore, scripts for implementing this type of voice transformation, based on the work of Tomoki Toda, are freely available from the FestVox website [9]. We modified these scripts to allow the use of additional features in the GMM mappings.

A high-level view of the approach taken in this paper can be seen in Figure 1. The general idea is that, in addition to mapping features derived from the speech signal data from one speaker to another, we can also map features derived from articulatory data from one speaker to another. In this paper we focus on comparing joint mappings of the speech signal and articulatory features from one speaker to another and how they compare to mappings that use only speech signal features.

2. MOCHA Database

For each of its speakers, the MOCHA database supplies audio files, Electro-Magneto Articulograph (EMA) files, laryngograph files, and electroglottograph files for the 460 sentences in the British TIMIT corpus [1]. There are two speakers for whom full data has been released. They are labeled msak0 and fsew0. The msak0 speaker is male and has a northern English accent. The fsew0 speaker is female and has a southern English accent.

The following experiments are based on features derived from the audio files and the EMA files. The audio files contain 16 bit samples at a rate of 16kHz. The EMA files contain samples at a rate of 500Hz of the x and y coordinates in the mid-sagittal plane of the positions of 7 different articulators, for a total of 14 values per sample. These 7 articulators include the upper and lower lip, the lower incisor, three points on the tongue, and the velum. The EMA files also contain additional coordinates for the bridge of the nose and the upper incisor, but they are only used for calibrating the positions of the other articulators and are not used as features in the following experiments.

3. Voice Transformation with GMM mapping

The basic idea behind GMM mapping techniques is that the probability of a joint feature vector, x, composed of features from both a source and target speaker, can be modeled by a GMM, which has the following probability density function:

$$p(x) = \sum_{i=1}^{M} \alpha_i \mathcal{N}(x; \mu_i, \Sigma_i)$$

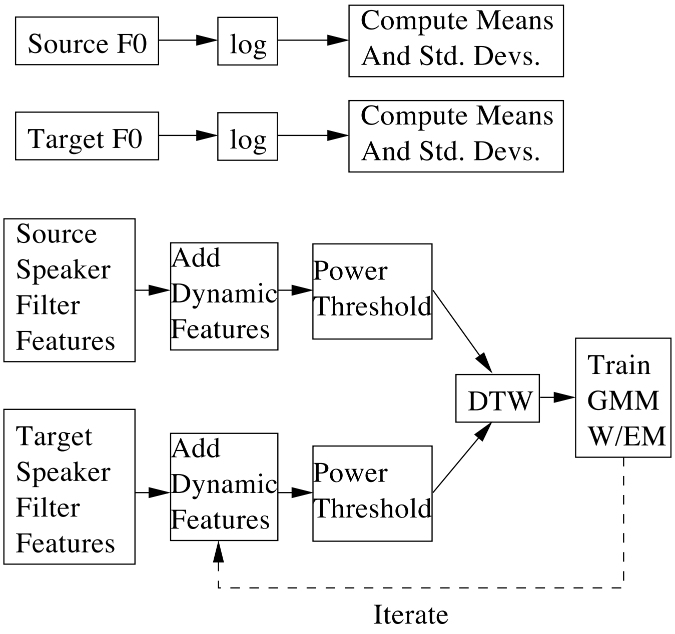
where M is the number of Gaussian components, $\mathcal{N}$ is a Gaussian distribution, μ_i and Σ_i are the mean and covariance of the ith Gaussian distribution, and the α_is are weights that are non-negative and sum to 1. In the following experiments, the default settings of the voice transformation scripts in FestVox are used to specify the form of the covariance matrix, which is diagonal in each quarter.

3.1. Training

The voice transformation training process is illustrated in Figure 2. It is based on recordings of the source and target speakers reading the same text. Fundamental frequency estimates are made for both speakers every 5ms, and mean and standard deviation statistics for their log values are calculated and recorded.

There is a separate part of the process that involves training a GMM based on filter features. The filter features used in the baseline system are the defaults used by the scripts from FestVox. 24 frequency-warped cepstral coefficients, called MCEPs, are extracted every 5ms from the recordings of the source and target speakers reading the same sentences. MCEPs approximate mel-cepstral coefficients and can be used with pitch estimates as inputs to the Mel Log Spectral Approximation (MLSA) filter [10], which is used to synthesize the transformed utterances. Dynamic features are also produced for the MCEP vectors using a weighted window centered on the current MCEP vector with values $[-0.5, 1, 0.5]$. At this point, there are now twice as many features for each speaker per frame. Frames below a certain power threshold are removed to reduce the chance of including background noise in the data. Because the durations of the parallel utterances will probably differ, dynamic time warping is used to align MCEP vectors between the two speakers to produce joint vectors with lengths of 4 times the original feature vectors (the original source speaker features plus the source speaker dynamic features plus the original target speaker features plus the target speaker dynamic features). The joint vectors are the ones that are modeled by the GMM, which is trained using EM. A couple iterations are performed where the trained GMM parameters are used to produce predictions from the source speech, which are then used to refine the DTW.

Figure 2: *Voice Transformation Training*



3.2. Transformation

Transformation is performed by the following process, which is illustrated in Figure 3:

1. Extract power, F_0, filter features (MCEP and possibly additional EMA values), and dynamic features from the utterance to be transformed.
2. Use a z-score mapping in the log domain to transform the source speaker's F_0 estimates to the target speaker's F_0 predictions.
3. Use the GMM to map the source speaker's features to the target speaker's by fixing the source speaker values and producing maximum likelihood estimates for the target speaker's features.
4. Use Maximum Likelihood Parameter Generation (MLPG) with global variance to predict final values based on filter features and dynamic features [11].
5. Use the power from the source speaker's utterance along with the F_0 and MCEP predictions as inputs to the MLSA filter to synthesize the transformed utterance.

Figure 3: *Voice Transformation*

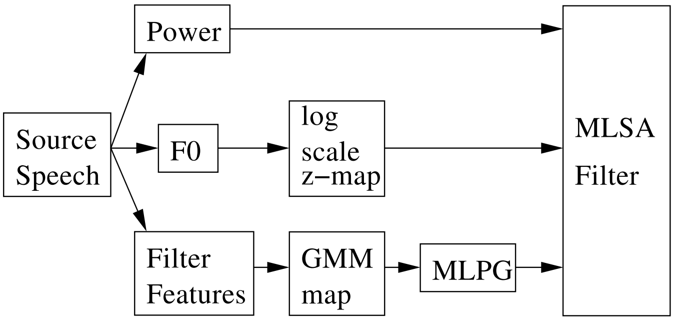


Table 1: *MCEP vs. EMAMCEP MCD Means (Std. Devs.)*

	msak0 to fsew0		fsew0 to msak0	
M	MCEP	EMAMCEP	MCEP	EMAMCEP
1	6.33(1.62)	6.88(1.61)	5.59(1.59)	5.95(1.68)
2	5.84(1.95)	6.34(1.97)	5.51(1.59)	5.79(1.71)
4	5.67(1.94)	6.25(2.06)	5.57(1.42)	5.81(1.64)
8	5.74(1.78)	6.60(1.65)	5.31(1.55)	5.95(1.62)
16	**5.58(1.79)**	6.09(1.89)	5.20(1.58)	5.46(1.62)
32	5.74(1.79)	N/A	5.06(1.62)	5.66(1.50)
64	5.74(1.70)	N/A	**5.01(1.63)**	N/A
128	N/A	N/A	N/A	N/A

3.3. Error Measure

Mel-Cepstral Distortion (MCD) is an objective error measure that is used in the following experiments to compare transformed utterances to reference utterances recorded by the target speaker. MCD has some correlation with results from subjective listening evaluations and has been used to measure the quality of voice transformation results in other work [7]. MCD is essentially a weighted Euclidean distance, that is defined by

$$MCD = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{24}(m_d^{(t)} - m_d^{(r)})^2}$$

where $m_d^{(t)}$ is the dth MCEP of a frame of transformed speech, and $m_d^{(r)}$ is the dth MCEP of the corresponding frame in the reference utterance recorded by the target speaker. Again, because the utterances will probably differ in length, Dynamic Time Warping is used to align them before computing the MCDs.

MCD is more related to filter characteristics of the vocal tract. Although characteristics such as power and fundamental frequency are also important to the quality of voice transformation output, the use of MCD for these experiments seems appropriate as the articulatory positions are expected to be most closely related to the filter characteristics of the vocal tract.

For the following results, no power thresholding was performed on frames before calculating MCDs, and the transformed MCEPs were used, as opposed to MCEPs rederived after synthesizing waveforms.

4. Adding Articulatory Position Data

Numerous experiments were conducted which added articulatory position data to the baseline MCEP features within the same general framework. The scripts were modified to allow the use of articulatory position features instead of or in addition to MCEP features. The rest of the processing continued in the same basic manner, with the exception that the error measure for the combination of articulatory position data and MCEPs was based solely on the MCEP subset. In the following descriptions, EMA will be used to refer to the articulatory position data, because it is the abbreviation for Electro-Magneto Articulograph, which is the specific type of articulatory position data that we used. Similarly, EMAMCEP will be used to refer to the combined use of EMA and MCEP data.

The EMA data from the MOCHA data had to be processed before combination with the MCEPs because it was sampled every 2ms instead of every 5ms, and the durations of the EMA files did not always match the durations of the audio files. Resampling was performed with the ch_track program from the Edinburgh Speech Tools [9], and EMA or MCEP features were truncated when the lengths didn't match.

Recordings from two speakers, msak0 and fsew0, were available from the MOCHA database. The experiments include transformations from each speaker to the other. The data was split into a training set of 414 utterances and a test set of 46 utterances. Most of the experiments were trained on a subset of 50 utterances due to the amount of time necessary to train the entire training set and the similarity of the results in some preliminary experiments.

Finally, there were some additional considerations that allowed the training of the GMM to work. The original EMA values were measured in thousandths of centimeters, and in some cases exceeded 5,000. Using these original values led to overflow errors with the training program, so we z-scored the EMA values to put them in a manageable range. Also, the number of Gaussian components in the GMM could affect whether training succeeded. In some cases the training program was unable to estimate parameters for the GMM and returned an error message suggesting that fewer Gaussian components should be used. In the following tables, the results for these trials will be marked as N/A (Not Applicable).

We tried to use multiple values to determine a range of success and also to track where increasing the number of components improved performance. After the initial trials, our basic choices were 16, 32, 64, or 128 components. These generally appeared to capture the range where results first improved and then worsened, presumably due to overtraining, or training even failing.

5. Experiments

5.1. Baseline Experiments

The first experiment was a comparison of only using MCEP features with using a combination of MCEP and EMA features. The only change made to the GMM mapping procedure for the initial trials including EMA was to include the EMA values in the feature vectors as well as the MCEP values. The results are in Table 1.

Adding all the EMA features directly as z-scored x and y coordinates in the mid-sagittal plane did not help in any of the trials, so it was necessary to investigate the data and the learning process more closely.

5.2. Attempts to Remove Noise from the Data

One possibility was that there was noise in the EMA data. Some potential causes were:

- The electrical apparatus originally used to collect the

Table 2: *Drift Correction MCD Means (Std. Devs.)*

	msak0 to fsew0	fsew0 to msak0
M	EMAMCEP	EMAMCEP
16	6.09(1.73)	5.58(1.59)
32	N/A	5.31(1.78)
64	N/A	N/A
128	N/A	N/A

Table 3: *First EMA Repeated MCD Means (Std. Devs.)*

	msak0 to fsew0		fsew0 to msak0	
M	MCEP	EMAMCEP	MCEP	EMAMCEP
16	5.54(1.80)	6.16(1.84)	5.18(1.59)	5.49(1.62)
32	5.67(1.82)	N/A	5.04(1.61)	5.45(1.71)
64	5.80(1.90)	N/A	5.02(1.61)	N/A
128	N/A	N/A	N/A	N/A

data

- The alignment of the MCEP with the EMA
- The resampling of the EMA data to match the default MCEP sampling rate

It has been noted by others [2] that there appears to be line noise at 50Hz in the MOCHA data. For that reason and also assuming that the motions of the articulators would be slow enough at our sampling rate, we tried applying low-pass filters with cut-offs of 45Hz and 10Hz to the MOCHA data using the sigfilter program from the Edinburgh Speech Tools [9]. Adding this low-pass filtered EMA data to the MCEP data failed to reduce the MCD error when compared to only using the MCEP data for voice transformation.

Another possible problem with the MOCHA data is that the means of the feature positions appear to vary over time more than what would be expected based on the differing phonetic contexts alone, according to other researchers [12] [13]. Although these sources were not certain whether this "drift" came from the Electo-Magneto Articulograph or the adjustment of speakers to the probes used to measure them, they found for their tasks that it was useful to try to compensate for it. We tried applying the "drift correction" strategy from the latter reference to the EMA data. This consisted of treating the mean values per utterance of the EMA features as signals, low-pass filtering these signals forward and backward with a FIR filter of length 100 and cut-off of 0.04π, and subtracting the resulting per-utterance "drift" values from the corresponding EMA features in the corresponding utterances. Adding the resulting drift-corrected data to the MCEP data failed to reduce the MCD error when compared to using the MCEP data alone for voice transformation, as can be seen in Table 2.

Another possible problem was that the EMA data was not aligned with the MCEP data. We experimented by shifting the EMA data one frame by repeating the first EMA frame. The results of this experiment are in Table 3.

This only made a minor change to the results and demonstrated that shifting the EMA by repeating the first EMA frame did not help. A companion experiment was performed where the first EMA frame was removed from each utterance. Shifting the EMA frames in that direction did not lead to an improvement in the results for trials using EMA data either. The results for this experiment are in Table 4. In both of these experiments, due to differences in the truncation of the feature files after alignment, there are small differences in the results for the trials which only used MCEP data.

Table 4: *First EMA Deleted MCD Means (Std. Devs.)*

	msak0 to fsew0		fsew0 to msak0	
M	MCEP	EMAMCEP	MCEP	EMAMCEP
16	5.54(1.80)	6.15(1.79)	5.18(1.59)	5.47(1.59)
32	**5.47(1.76)**	N/A	5.06(1.59)	5.69(1.67)
64	5.65(1.61)	N/A	**4.99(1.61)**	N/A
128	5.81(1.78)	N/A	N/A	N/A

Table 5: *DTW Based only on MCEPs MCD Means (Std. Devs.)*

	msak0 to fsew0	fsew0 to msak0
M	EMAMCEP	EMAMCEP
16	**5.84(1.81)**	5.35(1.73)
32	5.90(1.76)	**5.31(1.77)**
64	N/A	N/A
128	N/A	N/A

5.3. Attempts to Refine the Transformation Process

The baseline script that was used to perform voice transformation was based on techniques that were refined over time to handle MCEP data. It was unclear whether parts of this process were still appropriate when adding EMA data to the MCEP vectors. We investigated the following areas more closely:

- Dynamic Time Warping (DTW) used for alignment of the two speakers
- Use of the Maximum Likelihood Parameter Generation (MLPG) algorithm
- Use of multiple iterations of DTW during training

In the baseline voice transformation system, DTW was performed over all features and their derived dynamic features to align feature vectors between speakers. The distance measure used in the DTW was Euclidean. Because the MCEP and z-scored EMA values were not of the same scale, this did not seem appropriate. For this reason, we ran experiments that only considered the MCEP values during DTW when additional EMA features were used. The results are in Table 5. As can be seen through comparison with Table 1, this approach did not give better results than using MCEP data alone for the entire process. However, it did improve the results of the trials that included EMA data in comparison to previous trials that used EMA data, so it was used in later experiments.

One other thing to note is that basing the DTW only on MCEP features in the trials that also include EMA data leads to the same source speaker and target speaker frames being aligned across the different trials. This is not guaranteed when the DTW in the trials using EMA data also uses EMA values.

In the baseline voice transformation system, a program called MLPG is used to take the GMM estimates of the target speaker's MCEP and MCEP dynamic feature means and covariances to try to estimate final MCEP values that form a good path. It was unclear whether including EMA features in this process was appropriate. We ran another set of experiments where we used the means of the MCEP features for predictions and did not use MLPG (in addition to using the abovementioned strategy of only considering MCEP and MCEP dynamic feature

Table 6: *No MLPG and MCEP DTW MCD Means (Std. Devs.)*

	msak0 to fsew0		fsew0 to msak0	
M	MCEP	EMAMCEP	MCEP	EMAMCEP
16	**5.39(1.78)**	5.49(1.86)	4.95(1.57)	4.97(1.86)
32	**5.60(1.78)**	**5.50(1.81)**	4.91(1.59)	4.97(1.83)
64	5.76(1.84)	N/A	5.10(1.69)	N/A
128	N/A	N/A	N/A	N/A

Table 7: *Lip Distance MCD Means (Std. Devs.)*

	msak0 to fsew0	fsew0 to msak0
M	EMAMCEP	EMAMCEP
16	5.64(1.96)	5.40(1.78)
32	**5.55(2.00)**	5.25(1.80)
64	6.07(2.08)	5.19(1.81)
128	6.01(2.11)	5.19(1.89)

values during DTW). The results of these experiments are in Table 6. Adding EMA data helped in the trial that used 32 Gaussian components for the transformation from msak0 to fsew0. However, this was not a global best result for this transformation direction as the 16 Gaussian trial using only MCEP data still had better results.

5.4. Representation of EMA Features

Another possibility was that the x and y coordinates in the EMA data were a poor match for voice transformation in general or even the GMM mapping technique in particular. Perhaps there is more relevant information in features that are derived from these coordinates. After all, the x and y coordinate values are related to each other, both in terms of pairs being related to the same articulators, and in the sense that the positions of some articulators can pose constraints on the positions of others. Furthermore, the positions of some articulators relative to others provide information on constrictions in the vocal tract, which influence the filter characteristics. We investigated the following types of derived EMA features:

- Distances between the lips
- 1st order differences
- Projections onto lines fit to the articulator data

One type of vocal tract constriction that seemed reasonable to measure from the 7 articulators available in the MOCHA database was the distance between the lips. The two-dimensional Euclidean distance between the lips was used as a derived feature. The results for this experiment are in Table 7. In comparison with Table 1, it can be seen that adding lip distance improved the MCD when transforming from the msak0 voice to the fsew0 voice with 32 Gaussian components in the GMM.

Another thought was that capturing information about the motion of the articulators in two-dimensional space might supply more information. We ran experiments where the two-dimensional Euclidean distances were calculated between (x, y) coordinate pairs from frame to frame. This constructed 7 EMA derived features that could be added to the MCEP data. In this case, the dynamic features for the EMA are akin to second order differences. These trials were performed using only the MCEP and MCEP dynamic features for DTW and did not use MLPG. The results of these experiments are in Table 8. As can be seen by comparison with Table 6, adding these EMA derived distance features helped in the case of using 64 Gaussian components for the transformation from msak0 to fsew0. However, this was not a global positive result for the msak0 to fsew0 transformation as it did not perform as well as the 16 and 32 Gaussian component trials which only used MCEP data.

Table 8: *2-D EMA Distances MCD Means (Std. Devs.)*

	msak0 to fsew0	fsew0 to msak0
M	EMAMCEP	EMAMCEP
16	5.47(1.99)	5.21(1.73)
32	5.62(2.01)	5.14(1.80)
64	**5.56(2.02)**	N/A
128	N/A	N/A

Table 9: *EMA Projection MCD Means (Std. Devs.)*

	msak0 to fsew0	fsew0 to msak0
M	EMAMCEP	EMAMCEP
16	5.60(1.78)	5.01(1.85)
32	**5.36(1.97)**	5.00(1.86)
64	N/A	N/A
128	N/A	N/A

One problem with using 2-dimensional distances as features is that it does not include any notion of directionality, which seems like it should be important. There is a question of how to include this directionality in a meaningful way in the vectors used in the GMM mapping strategy. Although the articulator positions were measured in two dimensions, in many cases it appeared that individual articulators moved more along certain lines than others. For example, the lower incisor data showed more motion along the y-dimension than the x-dimension. In an attempt to capture some of this information, we derived features from the EMA by running linear regression on the (x, y) coordinate pairs in the training set for individual articulators to create best-fit lines, projecting the EMA (x, y) pairs onto these lines, and determining how far along these lines the articulators were. The results of using these projected EMA features are in Table 9. Again, in these trials, only the MCEP features were used for DTW and MLPG was not used. By comparison with Table 6, it can be seen that not only does adding these features improve the trial using 32 Gaussians for the transformation from msak0 to fsew0, but that this is a global positive result as it is better than all the other trials for transforming msak0 to fsew0, including the ones that only use MCEP data.

A different approach to investigating the possibility of the data being a mismatch for the model is to switch the model instead of changing the features. To this end, we tried using wagon, the Classification And Regression Tree (CART) program from the Edinburgh Speech Tools [9], instead of GMM mapping to learn the mapping between speakers. Using a step size of 100, CART predicted MCEPs from MCEPs in the fsew0 to msak0 direction with a MCD mean of 4.71 and standard deviation of 1.71. Using the combination of EMA data with MCEPs from the fsew0 speaker to predict MCEPs for the msak0 speaker gave a MCD mean of 5.22 and standard deviation of 1.90. Even with a different learning algorithm, adding EMA data failed to help improve voice transformation in terms of MCD. Although the numbers for the individual trials were better than for the GMM mapping baseline, there was the same general trend of the MCEP-only trial performing better than a trial that adds EMA x and y coordinates directly.

6. Conclusions

A number of strategies were applied to the problem of trying to use EMA data to improve a fairly standard GMM mapping based voice transformation technique in terms of Mel-Cepstral Distortion. For the most straightforward extension of the baseline voice transformation technique, none of the experimental trials that used additional EMA data directly as x and y coordinates improved the Mel-Cepstral Distortion. We made a number of attempts to use the EMA data to improve results. These attempts focused on the following three areas:

1. Removing noise from the data
2. Modifying parts of the voice transformation process that no longer appeared appropriate when using a combination of EMA and MCEP data
3. Finding a better way of representing EMA information in the model

In the first case, attempts to remove noise through filtering and realigning the EMA data, among other things, did not appear to help. In the second case, changing the way DTW was performed and not using MLPG led to results for the trials that used EMA to improve to the point where there was a trial where adding the EMA data led to better performance than using MCEP data alone. However, this was still not a global positive result as there was an MCEP trial with a different number of Gaussian components that outperformed it. In the third case, there was another positive result that came from using the distance between the lips, and finally, the first global positive result appeared in the case of using features derived from EMA by projecting the coordinates onto lines fit to the data through linear regression. In this case, the strategies of basing the DTW only on the MCEP data and not using MLPG were also followed.

It appears that the use of EMA data to improve voice transformation is not very straightforward. One additional thing to note is that all of the positive results occurred while transforming from msak0 to fsew0. There were none in the other direction. This appears to be another case of asymmetry in voice transformation. Asymmetric results have also been noted in identity perception for voice transformation [14].

There are numerous areas for further investigation. Maybe the Mel-Cepstral Distortion metric is not good enough for this task, even though it shows some correlation to subjective listening tests. Perhaps the information necessary for voice transformation is already present in MCEPs and EMA provides nothing additional. It is also possible that EMA features need to be combined or represented in a different space before they will be useful. Further experimentation will be necessary to tell.

7. Acknowledgments

This work was supported by the US National Science Foundation under grant number 00414675 "TRANSFORM: flexible voice synthesis through articulatory voice transformation". Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. References

[1] A. Wrench, "The MOCHA-TIMIT articulatory database," 1999, queen Margaret University College, http://www.cstr.ed.ac.uk/artic/mocha.html.

[2] E. Uraga and T. Hain, "Automatic speech recognition experiments with articulatory data," in *Interspeech 2006*, 2006.

[3] A. Toth, "Cross-speaker articulatory position data for phonetic feature prediction," in *Interspeech2005*, Lisboa, Portugal, 2005.

[4] Y. Stylianou, O. Cappé, and E. Moulines, "Statistical methods for voice quality transformation," in *Proc. EUROSPEECH95*, Madrid, Spain, 1995, pp. 447–450.

[5] A. Kain, "High resolution voice transformation," Ph.D. dissertation, OGI School of Science and Engineering, Oregon Health and Science University, 2001.

[6] T. Toda, "High-quality and flexible speech synthesis with segment selection and voice conversion," Ph.D. dissertation, Nara Institute of Science and Technology, 2003.

[7] T. Toda, A. Black, and K. Tokuda, "Mapping from articulatory movements to vocal tract spectrum with gaussian mixture model for articulatory speech synthesis," in *5th ISCA Speech Synthesis Workshop*, June 2004.

[8] ——, "Acoustic-to-articulatory inversion mapping with gaussian mixture model," in *Proc. ICSLP2004*, Oct. 2004, pp. 1129–1132.

[9] A. Black and K. Lenzo, "Building voices in the Festival speech synthesis system," 2000, http://festvox.org/bsv/.

[10] S. Imai, "Cepstral analysis synthesis on the mel frequency scale," in *Proceedings of ICASSP 83*, Boston, MA, 1983, pp. 93–96.

[11] T. Toda, A. Black, and K. Tokuda, "Spectral conversion based on maximum likelihood estimation considering global variance of converted parameter," in *Proc. ICASSP2005*, vol. 1, Philadelphia, PA, USA, Mar. 2006, pp. 9–12.

[12] K. Richmond, "Estimating articulatory parameters from the acoustic speech signal," Ph.D. dissertation, CSTR, University of Edinburgh, 2001.

[13] Y. Shiga, "Precise estimation of vocal tract and voice source characteristics," Ph.D. dissertation, CSTR, University of Edinburgh, 2005.

[14] A. Toth and A. Black, "Visual evaluation of voice transformation based on knowledge of speaker," in *ICASSP*, Toulouse, France, 2006.

Text Processing for Text-to-Speech Systems in Indian Languages

Anand Arokia Raj [1], *Tanuja Sarkar* [1], *Satish Chandra Pammi* [1],
Santhosh Yuvaraj [1], *Mohit Bansal* [2], *Kishore Prahallad* [1] [3], *Alan W Black* [3]

[1] International Institute of Information Technology, Hyderabad, India.
[2] Indian Institute of Technology, Kanpur, India.
[3] Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA.

skishore@cs.cmu.edu, awb@cs.cmu.edu

Abstract

To build a natural sounding speech synthesis system, it is essential that the text processing component produce an appropriate sequence of phonemic units corresponding to an arbitrary input text. In this paper we discuss our efforts in addressing the issues of Font-to-Akshara mapping, pronunciation rules for Aksharas, text normalization in the context of building text-to-speech systems in Indian languages.

1. Introduction

The objective of a text to speech system is to convert an arbitrary given text into a corresponding spoken waveform. Text processing and speech generation are two main components of a text to speech system. The objective of the text processing component is to process the given input text and produce appropriate sequence of phonemic units. These phonemic units are realized by the speech generation component either by synthesis from parameters or by selection of a unit from a large speech corpus. For natural sounding speech synthesis, it is essential that the text processing component produce an appropriate sequence of phonemic units corresponding to an arbitrary input text.

One of the question often asked by end-users is why we don't have TTS systems for all or many of the 23 official Indian languages. What are the complexities: Is it because the synthesis technology isn't matured enough to be able to build for any language or is it because of the non-existence of speech databases in Indian languages?. Unfortunately, for a decade the core speech generation technology i.e., generation of speech from a phonemic sequence has largely been automated due to unit selection techniques [1]. With the introduction of statistical parametric speech synthesis techniques, it is much easier to build a voice in a language with fewer sentences and a smaller speech corpus [2] [3].

It is difficult to convince an end-user that the input to a TTS system is not a phonemic sequence but rather the raw text as available in news websites, blogs, documents etc which contain the required text in font-encodings, native scripts and non-standard words such as addresses, numbers, currency etc. The majority of the issues are associated in building a TTS for a new language is associated with handling of real-world text [4]. Current state-of-art TTS system in English and other well-researched languages use such rich set of linguistic resources such as word-sense disambiguation, morphological analyzer, Part-of-Speech tagging, letter-to-sound rules, syllabification, stress-patterns in one form or the other to build a text processing component of a TTS system. However for minority languages (which are not well researched or do not have enough linguistic resources), it involves several complexities starting from accumulation of text corpora in digital and processable format. Linguistic components are not available in such rich fashion for all languages of the world. In practical world, minority languages including some of the Indian languages do not have that luxury of assuming some or any of the linguistic components.

The purpose of this paper is to describe our efforts at IIIT Hyderabad to build a generic framework for build text processing modules and linguistic resources which could be extended to all of the Indian languages with minimal efforts and time. Our approach is to make use of minimal language information (i.e., information available with an average educated native speakers), take the aid of acoustic data and machine learning techniques [5]. In this paper we summarize some of our efforts in this direction but mainly for font identification, Font-to-Akshara conversion, pronunciation rules for Aksharas and text normalization.

2. Nature of Indian Language Scripts

The scripts in Indian languages have originated from the ancient Brahmi script. The basic units of the writing system are referred to as *Aksharas*. The properties of Aksharas are as follows: (1) An Akshara is an orthographic representation of a speech sound in an Indian language; (2) Aksharas are syllabic in nature; (3) The typical forms of Akshara are V, CV, CCV and CCCV, thus have a generalized form of C*V.

The shape of an Akshara depends on its composition of consonants and the vowel, and sequence of the consonants. In defining the shape of an Akshara, one of the consonant symbols acts as pivotal symbol (referred to as semi-full form). Depending on the context, an Akshara can have a complex shape with other consonant and vowel symbols being placed on top, below, before, after or sometimes surrounding the pivotal symbol (referred to as half-form).

Thus to render an Akshara, a set of semi-full or half-forms have to be rendered, which in turn are rendered using a set of basic shapes referred to as *glyphs*. Often a semi-full form or half-form is rendered using two or more glyphs, thus there is no one-to-one correspondence between glyphs of a font and semi-full or half-forms [6].

2.1. Convergence and Divergence

There are 23 official languages of India, and all of them except English and Urdu share a common phonetic base, i.e., they share a common set of speech sounds. While all of these languages share a common phonetic base, some of the languages such as Hindi, Marathi and Nepali also share a common script known as Devanagari. But languages such as Telugu, Kannada and Tamil have their own scripts.

The property that makes these languages separate can be attributed to the phonotactics in each of these languages rather than the scripts and speech sounds. phonotactics is the permissible combinations of phones that can co-occur in a language.

2.2. Digital Storage of Indian Language Scripts

There is a chaos as far as the text in Indian languages in electronic form is concerned. Neither can one exchange the notes in Indian languages as conveniently as in English language, nor can one perform search easily on texts in Indian languages available over the web. This is because the texts are being stored in ASCII font dependent glyph codes as opposed to Unicode.

The glyph coding schemes are typically different for different languages and within a language there could exists several font-types with their own glyph codes (as many as major news-portals in a language). To view the websites hosting the content in a particular font-type, these fonts have to be installed on local machine. As this was the technology existed before the era of Unicode and hence a lot of electronic data in Indian languages were made and available in that form [7].

2.3. Need for Handling Font-Data

The text available in a font-encoding (or font-type) is referred to as *font-data*. While Unicode based news-portals and web-pages are increasing, there are two main reasons to deal with ASCII based font-data: 1) Given that there are 23 official Indian languages, and the amount of data available in ASCII based font-encodings is much larger than the text content available in Unicode format, 2) If a TTS system has to read the text from a ASCII font based website then the TTS system should automatically identify the font-type and process the font-data to generate speech.

2.4. A Phonetic Transliteration Scheme for Digital storage of Indian Language Scripts

To handle diversified storage formats of scripts of Indian languages such as ASCII based fonts, ISCII (Indian Standard code for Information Interchange) and Unicode etc, it is useful and becomes necessary to use a meta-storage format.

A transliteration scheme maps the Aksharas of Indian languages onto English alphabets and it could serve as meta-storage format for text-data. Since Aksharas in Indian languages are orthographic represent of speech sound, and they have a common phonetic base, it is suggested to have a phonetic transliteration scheme such as IT3 [8] [6]. Thus when the font-data is converted into IT3, it essentially turns the whole effort into font-to-Akshara conversion.

3. Identification of Font-Type

Given a document we often need to identify the font-type, and sometimes a document can contain the data encoded in different font-types. Then the task would boil down to identifying the font-type for each line or for each word. In this paper, we propose the use of TF-IDF approach for identification of font-type. The term frequency - inverse document frequency (TF-IDF) approach is used to weigh each glyph-sequence in the font-data according to how unique it is. In other words, the TF-IDF approach captures the relevancy among glyph-sequence and font-type. In this approach, the term refers to a 'glyph' and the document refers to the font-data of a particular 'font-type'. Here the glyph-sequence could mean a single glyph or 'current and next' glyph or 'previous, current and next' glyph etc.

To build a document for each font-type, a web-site for each font-type was manually identified and around 0.12 million unique words were crawled for each of the font-type. The set of unique words for each font-type are referred to as a document representing the particular font-type. Thus given N documents (each representing a font-type), we considered three different terms namely, a single glyph or *current and next* glyph or *previous, current and next* glyph. For each term a TF-IDF weight was obtained as follows: (i) Calculate the term frequency for the glyph-sequence: The number of times that glyph-sequence occurred divided by the total number of glyph-sequences in that specific document. (ii) Calculate document frequency: In how many different documents (font-types) that specific glyph-sequence has occurred. (iii) Calculate inverse document frequency of the term and take logarithm of inverse document frequency.

To identify the font-type of a given test font-data, the steps involved are as follows: 1) Generate the terms (glyph-sequences) of the test font-data 2) Compute the relevancy scores of the terms and for each of the document (font-type) using the corresponding TF-IDF weights of the terms 3) The test font-data belongs to the document (font-type) which produces a maximum relevancy score.

The performance of TF-IDF approach for identification of font-type was evaluated on 1000 unique sentences and words per font-type. We have added English data as also one of the testing set, and is referred to as English-text. The performance of font-type identification system using different terms *single* glyph, *current and next* glyphs, *previous, current and next* glyphs are shown in Table 1, Table 2 and Table 3 respectively and it could be observed that the use of *previous, current and next* glyphs as a term provided an accuracy of 100% in identification of font-type even at the word level.

Table 1: *Performance of Single glyph based font models*

Font Name	Sentence-Level	Word-Level
Amarujala (Hindi)	100%	100%
Jagran (Hindi)	100%	100%
Webdunia (Hindi)	100%	0.1%
SHREE-TEL (Telugu)	100%	7.3%
Eenadu (Telugu)	0%	0.2%
Vaarttha (Telugu)	100%	29.1%
Elango_Panchali (Tamil)	100%	93%
Amudham (Tamil)	100%	100%
SHREE-TAM (Tamil)	100%	3.7%
English-text	0%	0%

4. Font-to-Akshara Mapping

Font-data conversion can be defined as converting the font encoded data into Aksharas represented using phonetic transliteration scheme such as IT3. As we already mentioned that Aksharas are split into glyphs of a font, and hence a conversion from font-data has essentially to deal with glyphs and model how a sequence of glyphs are merged to form an Akshara. As there exist many fonts in Indian languages, we have designed a generic framework has been designed for the conversion of font-data. It

Table 2: *Performance of current and next glyph based font models*

Font Name	Sentence-Level	Word-Level
Amarujala (Hindi)	100%	100%
Jagran (Hindi)	100%	100%
Webdunia (Hindi)	100%	100%
SHREE-TEL (Telugu)	100%	100%
Eenadu (Telugu)	100%	100%
Vaarttha (Telugu)	100%	100%
Elango_Panchali (Tamil)	100%	100%
Amudham (Tamil)	100%	100%
SHREE-TAM (Tamil)	100%	100%
English-text	100%	96.3%

Table 3: *Performance of previous, current and next based font models*

Font Name	Sentence-Level	Word-Level
Amarujala (Hindi)	100%	100%
Jagran (Hindi)	100%	100%
Webdunia (Hindi)	100%	100%
SHREE-TEL (Telugu)	100%	100%
Eenadu (Telugu)	100%	100%
Vaarttha (Telugu)	100%	100%
Elango_Panchali (Tamil)	100%	100%
Amudham (Tamil)	100%	100%
SHREE-TAM (Tamil)	100%	100%
English-text	100%	100%

has two phases, in the first phase we are building the base-map table for a given font-type and in the second phase forming and ordering the assimilation rules for a specific language.

4.1. Building a Base-Map Table for a Font-type

The base-map table provides the mapping basic between the glyphs of the font-type to the Aksharas represented in IT3 transliteration scheme. The novelty in our mapping was that the shape of a glyph was also included in building this mapping table. The shape of a glyph is dictated by whether it is rendered as pivotal consonant, or on top, bottom, left or right of the pivotal consonant. Thus the pivotal glyphs were appended with 0 (for full characters such as e, ka) or 1 (for half consonants such as k1, p1), '2' for glyphs occur at left hand side of a basic character (ex: i2, r2), '3' for glyphs occur at right hand side of a basic character (ex: au3, y3), '4' for glyphs occur at top of a basic character (ex: ai4, r4) and '5' for glyphs occur at bottom of a basic character (ex: u5, t5).

4.2. Forming Assimilation Rules

In the conversion process the above explained basic-mapping table will be used as the seed. A well defined and ordered set of assimilation rules have to be formed for each and every language. Assimilation is the process of merging two or more glyphs and generating a valid single character. This assimilation happens at different levels and our observation across many languages was that the firing of following assimilation rules were universally applicable. The rules are:(i) Modifier Modification, (ii) Language Preprocessing, (iii) Consonant Assimilation, (iv) Maatra Assimilation, (v) Consonant-Vowel Assimilation, (vi) Vowel-Maatra Assimilation, (vii) Consonants Clustering and (viii) Schwa Deletion.

The Modifier Modification is the process where the characters get modified because of the language modifiers like virama and nukta (ka + virama = k1). The Language Preprocessing step deals with some language specific processing like (aa3 + i3 = ri in Tamil) and (r4 moves in front of the previous first full consonant in Hindi). The Consonant Assimilation is known as getting merged two or more consonant glyphs and forms a valid single consonant like (d1 + h5 = dh1 in Telugu). The Maatra Assimilation is known as getting merged two or more maatra glyphs and forms a valid single maatra like (aa3 + e4 = o3 in Hindi). The Consonant-Vowel Assimilation is known as getting merged two or more consonant and vowel glyphs and forms a valid single consonant like (e + a4 + u5 = pu in Telugu). The Vowel-Maatra Assimilation is known as getting merged two or more vowel and maatra glyphs and forms a valid single vowel like (a + aa3 = aa in Hindi). The Consonant Clustering in known as merging the half consonant which usually occurs at the bottom of a full consonant to that full consonant like (la + l5 = lla in Hindi). The Schwa Deletion is deleting the inherent vowel 'a' from a full consonant in necessary places like (ka + ii3 = kii).

4.3. Testing and Evaluation

The evaluation on these font converters is carried out in two phases. We picked up three different font-types for training or forming the assimilation rules and one new font-type for testing per language. In the first phase for the selected three font-types the assimilation rules are formed and refined. In the second phase we chose a new font-type and built the base-map table only and used the existing converter without any modifications. We have taken 500 unique words per font-type and generated the conversion output. The evaluation results in Table 4 show that the font converter performs consistently even for a new font-type. So it is only sufficient to provide the base-map table for a new font-type to get a good conversion results. The issue of Font-to-Akshara mapping has been attempted in [7] and [9] but we believe that our framework is a generic one which could easily be extended to a new font-type with $> 99\%$ conversion accuracy.

5. Building Pronunciation Models For Aksharas

Having converted the font-data into Aksharas, the next step is to obtain appropriate pronunciation for each of the Aksharas. As noted earlier, Aksharas are orthographic representation of speech sounds and it is commonly believed or quoted that there is direct correspondence between what is written and what is spoken in Indian languages, however, there is no one-to-one correspondence between what is written and what is spoken. Often some of the sounds are deleted such as Schwa deletion in Hindi. Schwa is the default short vowel /a/which is associated with a consonant, and often it is deleted to aid in faster pronunciation of a word. Similarly there exists exceptions for Bengali and Tamil. There are attempts to model these exceptions in the form of the rules, however, they are often met with limited success or they use linguistic resources such as Morph analyzer. Such linguistic resources may not always be available for minority languages. Thus we had built a framework based on machine learning techniques where pronunciation of Aksharas could be modeled using machine learning techniques and using a small set of supervised training data.

Table 4: *Performance results for font conversion in Indian languages*

Language	Font Name	Training/Testing	Accuracy
Hindi	Amarujala	Training	99.2%
	Jagran	Training	99.4%
	Naidunia	Training	98.8%
	Webdunia	Training	99.4%
	Chanakya	Testing	99.8%
Marathi	Shree Pudhari	Training	100%
	Shree Dev	Training	99.8%
	TTYogesh	Training	99.6%
	Shusha	Testing	99.6%
Telugu	Eenadu	Training	93%
	Vaartha	Training	92%
	Hemalatha	Training	93%
	TeluguFont	Testing	94%
Tamil	Elango Valluvan	Training	100%
	Shree Tam	Training	99.6%
	Elango Panchali	Training	99.8%
	Tboomis	Testing	100%
Kannada	Shree Kan	Training	99.8%
	TTNandi	Training	99.4%
	BRH Kannada	Training	99.6%
	BRH Vijay	Testing	99.6%
Malayalam	Revathi	Training	100%
	Karthika	Training	99.4%
	Thoolika	Training	99.8%
	ShreeMal	Testing	99.6%
Gujarati	Krishna	Training	99.6%
	Krishnaweb	Training	99.4%
	Gopika	Training	99.2%
	Divya	Testing	99.4%

5.1. Creation of Data-set

Given the input word list with the corresponding pronunciations in terms of phones, feature vectors were extracted for training the pronunciation model at the phone level. About 12200 sentences in IT3 format were used to collect the training data, for building the pronunciation model in Hindi. These sentences had about 26000 unique words, which were used to extract around 32800 feature vectors. Different sets of feature vectors to experiment on the selection of features. As for Bengali and Tamil, 5000 words with corresponding pronunciations were used for obtaining about 9000 feature vectors.

5.2. Use of Contextual Features

Contextual features refers to the neighbor phones in a definite window-size/level. Using the contextual features, experiments were performed for various Contextual Levels (CL). A decision forest was built for each phone to model its pronunciation. A decision forest is a set of decision trees built using overlapping but different sub-sets of the training data and it employs a majority voting scheme on individual prediction of different trees to predict the pronunciation of a phone. Table 5 shows the results of pronunciation model for Hindi, Bengali and Tamil using various level of contextual features. We found that that a context level of 4 (i.e., 4 phones to the left and 4 phones to the right) was sufficient to model the pronunciation and moving beyond the level of 4, the performance was degraded.

Table 5: *Pronunciation Model with Contextual features*

Languages	Context Level			
	2	3	4	6
Hindi	90.24%	91.44%	**91.78%**	91.61%
Bengali	82.77%	84.48%	**84.56%**	83.56%
Tamil	98.16%	**98.24%**	98.10%	98.05%

5.3. Acoustic-Phonetic and Syllabic Features

Acoustic phonetic features lists the articulatory properties of the consonants and the vowels. Typically vowels are characterized by the front, back, mid position of the tongue while consonants are characterized by manner and place of articulation and voicing and nasalization features. Syllabic features indicate where a particular syllable is of type CV or CCV, or CVC etc. The performance of the pronunciation model for Hindi, Tamil and Bengali using syllabic and acoustic-phonetic features of the current and neighboring phones are shown in Table 6 and Table 7 respectively. We found that the use of syllabic or acoustic-phonetic features didn't show any significant improvement than that of contextual features for Hindi, Tamil and Bengali.

A rule based algorithm for Hindi LTS is given in [10]. To compare our results with the rule-based algorithm, we have used the same algorithm with out morphological analyzer on our test data set. We found that the performance of pronunciation model using rule-based technique was 88.17%. while the decision forest model in Table 6 was providing an accuracy of 92.29%.

Table 6: *Pronunciation Model with Syllabic features*

Feature Sets	Languages		
	Hindi	Bengali	Tamil
Syl_Struct. of Cur. Phone	**92.29%**	**82.41%**	**98.31%**
Syl_Struct. of all Phones	91.61%	67.56%	98.27%

Table 7: *Pronunciation Model with Acoustic-Phonetic features*

Feature Sets	Languages		
	Hindi	Bengali	Tamil
Acoustic_Phonetic	89.73%	**84.78%**	**98.18%**
+ Syl_Struct. of Curr. Phone	89.73%	81.21%	98.17%
+ Syl_Struct. of all Phones	**91.09%**	69.33%	**98.13%**

6. Normalizing of Non-Standard Words

Unrestricted texts include Standard Words (common words and Proper Names) and Non-Standard Words (NSWs). Standard Words have a specific pronunciation that can be phonetically described either in a lexicon, using a disambiguation processing to some extent, or by letter-to-sound rules. In the context of TTS the problem is to decide how an automatic system should pronounce a token; even before the pronunciation of a token, it

Table 8: *Taxonomy of NSWs with examples*

Category	Description	Examples
Addr	Address (house/streetno.)	12/451 Janapath Road
Curr	Currency	Rs. 7635.42
Count	Count of items	10 computers, 500 people
Date	Date(to be expanded)	1/1/05, 1997-99
PhoneNo	As sequence of digits	040 2300675
Pin	As sequence of digits	208023
Score	Cricket, tennis scores	India 123/4, sets 3-5 3-4 5-6
Time	Time (to be expanded)	1.30, 10:45-12:30, 11.12.05, 1930 hrs
Units	As decimal or number	10.5 kms, 98 %, 13.67 acres
NUM	Default category	

is important to identify the NSW-Category of a token. A typical set of NSW-category and their examples are shown in Table 8.

6.1. Creation of Supervised Training Data

To build a training dataset, it typically requires a large manual effort to annotate an example with the appropriate NSW-category. For example, given a word corpus $> 3M$ words in Telugu, Tamil and Hindi, we extracted 150-500K sentences containing an NSW. Annotating such huge set of examples needs lots of time and effort. To minimize such effort, we used a novel frequency based approach to create a representative example set.

NSW techniques uses context information for disambiguation with various window sizes, context information contains a set of word like units which occurs in left and right side of a NSW, and this information is to be considered as a features characterizing a NSW. However, not of all context would be useful, so we used a window size of 2 (left and right) as a default and given to the pattern generator module. The pattern generator takes the four tokens (two to left and two to the right of a NSW) and generates 15 patterns using all possible combinations of 4(like examples, 0001, 0010, 0011, 0100, ., 111) where 1 represent presence of a token and 0 represent deletion of the token. Given such 15 patterns for each example, these patterns were sorted in the descending order of their frequency and based on a threshold a set of patterns were choosen and given to a native speaker to annotate the NSW category. The user interface was built such that if the native speaker couldn't annotate the NSW with the given pattern, then an extended context was presented to him at varying levels. Using the frequency based approach, we could reduce the training examples to around 1000-1500 which a native could annotate within a couple of hours. Having got the annotation done, we looked at level of context the native speaker has used to annotate a NSW. We found less than 10% of time the user has looked into a context information more than a window size of two.

6.2. Performance of Base-line System

Using word level units and decision tree, we built a base-line system to predict the category of a NSW. We have tested the performance of the system on a separate manually prepared data obtained from a different source (web) referred to as Test-Set-1 (TS1). The results of prediction of NSW category on TS1 is shown in Table 9.

Table 9: *Performance of prediction of NSW-category Using Word Level Features*

Language	% accuracy on Training set	% accuracy on TS1
Telugu	99.57%	63.52%
Hindi	99.80%	66.99%
Tamil	99.01%	55.42%

Table 10: *Performance of prediction of NSW-category Using Syllable level Features*

Language	% accuracy on Training set	% accuracy on TS1	Diff with base-line
Telugu	99.57%	91.00%	27.48%
Hindi	99.80%	82.80%	15.81%
Tamil	99.01%	87.20%	31.78%

The performance of the base-line system on TS1 is around 60%. After analyzing the errors made by the system, we found that the errors are primarily due to new words found in the context of NSW, and Indian languages being rich in inflectional and derivative morphology, the roots of many of these words were present in the training data. It suggests that we should use roots of the context as the features to predict NSW-category, however, such approach needs morphological analyzers. Many of the Indian languages fall into category of minority languages where linguistic resources are scarce. Thus we wanted to investigate sub-word units such as syllables and their combinations as features for prediction of NSW-category.

Our experiments on POS-tagging on Hindi, Bengali and Telugu using syllable-level units further provided evidence that syllable level features could be used as alternative and a first-order approximation of root of a word [11]. After initial set of experiments to explore different possibilities of using syllable-level features, we confined to a set of following three syllable level features. They are: 1) F1: previous ten and next ten syllables of a NSW, 2) F2: previous ten and next ten syllables and onset of each syllables and 3) F3: Onset, vowel and coda of previous ten and next ten syllables.

Using decision forest, the final prediction of NSW-category is chosen based on voting on the outputs of the three decision trees built using F1, F2 and F3. This strategy gets the results of each decision tree and performs a majority voting to predict the NSW-category. The performance of the decision forest based system using syllable level features is shown in Table 10. We found that the results of using syllable-level features for text normalization performed significantly better than that of using word-level features. This significant improvement in the performance is primarily due to syllables acting a first-order approximation of roots of the context words and thus minimizing the problem of unseen context. The final performance of the text normalization system is further improved after using expander module from 91.00%, 82.80% and 87.20% to 96.60%, 96.65% and 93.38% for languages Telugu, Hindi and Tamil respectively.

7. Conclusions

This paper explained the nature and difficulties associated with building text processing components of TTS systems in Indian languages. We have discussed the relevancy of font-identification and font-to-Akshara conversion and proposed a TF-IDF based approach for font-identification. A novel approach of conversion from font-to-Akshara using the shapes of the glyphs and the assimilation rules was explained. We have also studied the performance of pronunciation models for different features including contextual, syllabic and acoustic-phonetic features. Finally we have shown that syllable-level features could be used to build a text normalization system whose performance is significantly better than the word-level features.

8. References

[1] Hunt A.J. and Black A.W., "Unit selection in a concatenative speech synthesis system for a large speech database," in *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, 1996, pp. 373–376.

[2] Black A.W., Zen H., and Tokuda K., "Statistical parametric speech synthesis," in *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, Honolulu, USA, 2007.

[3] Zen H., Nose T., Yamagishi J., Sako S., Masuko T., Black A.W., and Tokuda K., "The hmm-based speech synthesis system version 2.0," in *Proc. of ISCA SSW6*, Bonn, Germany, 2007.

[4] Sproat R., Black A.W., Chen S., Kumar S., Ostendorf M., and Richards C., "Normalization of non-standard words," *Computer Speech and Language*, pp. 287–333, 2001.

[5] HaileMariam S. and Prahallad K., "Extraction of linguistic information with the aid of acoustic data to build speech systems," in *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, Honolulu, USA, 2007.

[6] Prahallad L., Prahallad K., and Ganapathiraju M., "A simple approach for building transliteration editors for indian languages," *Journal of Zhejiang University Science*, vol. 6A, no. 11, pp. 1354–1361, 2005.

[7] Garg H., *Overcoming the Font and Script Barriers Among Indian Languages*, MS dissertation, International Institute of Information Technology, Hyderabad, India, 2004.

[8] Ganapathiraju M., Balakrishnan M., Balakrishnan N., and Reddy R., "Om: One tool for many (Indian) languages," *Journal of Zhejiang University Science*, vol. 6A, no. 11, pp. 1348–1353, 2005.

[9] Khudanpur S. and Schafer C., "http://www.cs.jhu.edu / cschafer/ jhu_devanagari_cvt_ver2.tar.gz," 2003.

[10] Choudhury M., "Rule-based grapheme to phoneme mapping for hindi speech synthesis," in *90th Indian Science Congress of the International Speech Communication Association (ISCA)*, Bangalore, India, 2003.

[11] S. Chandra Pammi and Prahallad K., "POS tagging and chunking using decision forests," in *Proceedings of Workshop on Shallow Parsing in South Asian Languages, IJCAI*, Hyderabad, India, 2007.

Flexible Harmonic/Stochastic Speech Synthesis

Daniel Erro, Asunción Moreno, Antonio Bonafonte

TALP Research Center
Department of Signal Theory and Communications
Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
{derro,asuncion,antonio}@gps.tsc.upc.edu

Abstract

In this paper, our flexible harmonic/stochastic waveform generator for a speech synthesis system is presented. The speech is modeled as the superposition of two components: a harmonic component and a stochastic or aperiodic component. The purpose of this representation is to provide a framework with maximum flexibility for all kind of speech transformations. In contrast to other similar systems found in the literature, like HNM, our system can operate using constant frame rate instead of a pitch-synchronous scheme. Thus, the analysis process is simplified, while the phase coherence is guaranteed by the new prosodic modification and concatenation procedures that have been designed for this scheme. As the system was created for voice conversion applications, in this work, as a previous step, we validate its performance in a speech synthesis context by comparing it to the well-known TD-PSOLA technique, using four different voices and different synthesis database sizes. The opinions of the listeners indicate that the methods and algorithms described are preferred rather than PSOLA, and thus are suitable for high-quality speech synthesis and for further voice transformations.

1. Introduction

In concatenative speech synthesis, a set of recorded speech units are selected from a database and are concatenated to create synthetic utterances. The prosodic characteristics of the units are adapted to the desired prosodic contour and the discontinuities between the different units are minimized at the boundaries. The performance of the speech synthesis systems strongly depends on the techniques and algorithms used for all these tasks. Furthermore, voice conversion methods are usually integrated into speech synthesis systems as a complement used to modify the physical attributes of the output voice to be perceived by the listeners like a different voice. This fact makes desirable the choice of flexible signal models capable of providing a high degree of flexibility without causing artifacts.

In [1] we presented a new simple method for prosodic modification of speech and for concatenation of speech units. The harmonic plus stochastic model of speech (HSM) was used to implement the waveform generation block of a text-to-speech synthesis system (TTS), due to the flexibility and capacity of manipulation provided by the model, as well as its interesting properties for embedded systems. Unlike in Stylianou's HNM [2] and other similar methods, the prosodic modifications were not based on pitch-synchronous overlap-add (PSOLA) techniques [3]. The main advantage of the system was that, although neither pitch marks nor accurate separation of signal periods were necessary, the inter-frame phase coherence and the speech waveform shape invariance were successfully maintained by means of new phase manipulation algorithms. Therefore, the analysis of speech was simplified using a constant frame rate, whereas the usage of onset times, source-filter separation techniques and cross-correlation-based phase corrections was also avoided, in contrast to other previous non-pitch-synchronous sinusoidal systems [4, 5, 6]. Instead, the modification algorithms designed for the new method were conceptually simple and straightforward.

At present, successful voice conversion methods compatible with HSM have been designed and tested on natural speech in a public evaluation campaign in both, intra-lingual and cross-lingual applications, achieving excellent results [13]. In this paper we describe the full HSM-based waveform generation block, which has been integrated into a TTS system. New improved phase manipulation algorithms, related to the prosodic modification and concatenation of speech units, are explained in detail. The purpose of the comparative experiments conducted in this paper is to validate the suitability of our waveform generator for high-quality speech synthesis, prior to using it for converting synthetic voices. A brief explanation about our voice conversion method is also included, although it does not take part in the discussion here.

The paper is structured as follows. Section 2 shows how the speech signals are analyzed and reconstructed from the measured parameters. Section 3 describes the algorithms for modifying the signal parameters in order to change the pitch or duration of speech. Section 4 deals with the artifact-free concatenation of speech units. The system is extended with the WTW voice conversion method in section 5. In section 6 the performance of the speech synthesis system is evaluated by comparing it to the UPC TTS system Ogmios [7]. The main conclusions of this work are listed in section 7.

2. Analysis and Reconstruction of Signals

The harmonic plus stochastic model (HSM) [2] assumes that the speech signal can be represented as a sum of a number of harmonically related sinusoids with time-varying parameters and a noise-like component. The harmonic component is present only in the voiced fragments of speech. It can be represented at each analysis frame by the fundamental frequency and the amplitudes and phases of the harmonics. The stochastic component tries to model all the non-sinusoidal signal components, caused by the frication, breathing noise, etc. It can be represented at each frame by the coefficients of an all-pole filter.

2.1. Analysis

The signals are analyzed using a constant frame rate of 100 frames per second. Given a speech frame to be analyzed, frame number *k*, the fundamental frequency $f_0^{(k)}$ has to be estimated and a binary voicing decision is taken. If the frame is voiced, the amplitudes $\{A_j^{(k)}\}$ and phases $\{\varphi_j^{(k)}\}$ of all the harmonics below a cutoff frequency of 5 KHz are detected. The choice of a fixed cutoff frequency is adequate for voice conversion purposes, because the spectral envelopes are extracted from the harmonic component. The amplitudes and phases are obtained by means of a least squares optimization in the spectral domain, using the algorithm of Depalle et al. [8] particularized to harmonic sinusoids:

$$S^{(k)}(f)=\sum_{j=1}^{J^{(k)}}\tfrac{1}{2}A_j^{(k)}\left[e^{i\varphi_j^{(k)}}W(f-jf_0)+e^{-i\varphi_j^{(k)}}W(f+jf_0)\right] \quad (1)$$

$S^{(k)}(f)$ is the STFT of the k^{th} frame and $W(f)$ denotes the Fourier transform of the analysis window, whose length is two pitch periods. $J^{(k)}$ is the highest integer that satisfies $J^{(k)}\cdot f_0^{(k)}<5$ KHz. As the optimization is performed in the spectral domain, the relative position of the analysis window within the pitch period is not important. This is adequate for a pitch-asynchronous analysis framework.

Once the frequencies, amplitudes and phases of the harmonics are known, the sinusoidal component of the signal is regenerated by interpolating between the measured values. For each time instant, the instantaneous amplitudes are obtained by means of a linear interpolation, and the 3rd order polynomial proposed by McAulay and Quatieri [9] is used to interpolate the instantaneous frequencies and phases of each harmonic. The regenerated harmonic component is subtracted from the original signal, and the remaining part of the signal, which corresponds to the stochastic component, is LPC-analyzed at each frame.

2.2. Reconstruction

The signal is reconstructed by overlapping and adding 2*N*-length frames, where *N* is the distance between the analysis frame centres, measured in samples. Each synthetic frame contains the sum of the measured harmonics with constant amplitudes, frequencies and phases, and the stochastic contribution, generated by filtering white gaussian noise with the measured LPC-filters. A triangular window is used to overlap-add the frames in order to obtain the time-varying synthetic signal. Let *k* be the frame number and *j* the harmonic number. The following expressions are used to reconstruct the signal.

$$s^{(k)}[n]=\sum_{j=1}^{J^{(k)}}A_j^{(k)}\cos\left(2\pi jf_0^{(k)}n/f_s+\varphi_j^{(k)}\right)+\sigma[n]*h_{LPC}^{(k)}[n] \quad (2a)$$

$$s[kN+m]=\left(\tfrac{N-m}{N}\right)\cdot s^{(k)}[m]+\left(\tfrac{m}{N}\right)\cdot s^{(k+1)}[m-N] \quad (2b)$$

where *m* is in the range [0, *N*–1]. The speech signal resynthesized from the measured parameters is almost indistinguishable from the original.

3. Prosodic Modifications

As a pitch-asynchronous scheme is being used, the prosodic modification of the signal implies the challenge of modifying the phases of the harmonics without altering the phase coherence between frames or causing artifacts. For this purpose, we have developed new strategies to manipulate the phases. We consider that the phases $\varphi_j^{(k)}$ measured at a certain analysis frame *k* are the sum of two components: a linear-in-frequency term given by the parameter $\alpha^{(k)}$, and the phase contribution of the time-varying vocal tract, $\theta_j^{(k)}$.

$$\varphi_j^{(k)}=j\alpha^{(k)}+\theta_j^{(k)} \quad (3)$$

The estimation of $\alpha^{(k)}$ is discussed in section 3.3.

3.1. Duration Modification

The duration modification can be carried out by increasing or decreasing the distance *N* between the synthesis points in equation (2b), so that the amplitude and fundamental frequency variations get adapted to the new time scale. On the other hand, if the phases were kept unmodified, fixed at the center of the frames, the waveform coherence between consecutive points would be lost, causing artifacts and noisy pitch variations. Therefore, the change in *N* needs to be compensated with a phase manipulation in a way that the waveform and pitch of the duration-modified signal are similar to the original. This manipulation should affect only to the linear-in-frequency phase term. Assuming that the fundamental frequency varies linearly from frame *k*–1 to *k*, we define the function Ψ which represents the expected phase increment of the first harmonic between those points, affecting only the linear-in-frequency term:

$$\alpha^{(k)}-\alpha^{(k-1)}\cong\Psi\left(f_0^{(k-1)},f_0^{(k)},N\right)=\pi N\left(f_0^{(k-1)}+f_0^{(k)}\right)/f_s \quad (4)$$

If *N* is substituted by *N'*, the following phase correction is applied:

$$\Delta\varphi_1^{(k)}=\Psi\left(f_0^{(k-1)},f_0^{(k)},N'\right)-\Psi\left(f_0^{(k-1)},f_0^{(k)},N\right) \quad (5a)$$

$$\varphi_j'^{(k)}=\varphi_j^{(k)}+j\sum_{\kappa=2}^{k}\Delta\varphi_1^{(\kappa)} \qquad j=1\ldots J^{(k)} \quad \forall k>1 \quad (5b)$$

This correction compensates the modification of *N* without affecting the small local variations in the vocal tract phase response. The stochastic coefficients are not modified. Note that the modification factor can be time-varying.

3.2. Pitch Modification

For the pitch modifications, the amplitudes of the new harmonics $A_j'^{(k)}$ are obtained by a simple linear interpolation between the measured log-amplitudes in order to maintain the formant structure unaltered. A constant multiplicative factor is used to keep constant the energy of the harmonic component despite the variation of the number of sinusoids. The vocal tract contribution to the phases of the new harmonics, $\theta_j'^{(k)}$, can be obtained by means of a linear interpolation of the real and imaginary parts of the complex amplitudes $A_j^{(k)}\exp(i\cdot\theta_j^{(k)})$. The values of $\theta_j^{(k)}$ are calculated from the original phases $\varphi_j^{(k)}$ by subtracting the linear-in-frequency phase term given by $\alpha^{(k)}$.

$$\theta_j^{(k)}=\varphi_j^{(k)}-j\alpha^{(k)} \quad (6)$$

Finally, the relative position of the synthesis point within the new pitch period is now different and the linear term has to be corrected to compensate the modification of the periodicity. The phase correction to be performed is given by (5b) with

$$\Delta\varphi_1^{(k)} = \Psi\left(f_0'^{(k-1)}, f_0'^{(k)}, N\right) - \Psi\left(f_0^{(k-1)}, f_0^{(k)}, N\right) \qquad (7)$$

The stochastic coefficients are not modified. Time-varying modification factors can be used following this method, and the simultaneous duration + pitch modification of the signal is also possible.

3.3. Linear Phase Term Estimation

The estimation of the parameter $\alpha^{(k)}$ is a crucial point for obtaining high quality synthetic speech. In pitch synchronous systems the linear phase term is zero at the frame centres, but in exchange a set of pitch marks need to be stored and synchronized with the waveform. Even in such systems, some problems appear when there are linear phase mismatches between different signal periods [10]. We propose to estimate $\alpha^{(k)}$ using the following formula.

$$\alpha^{(k)} = \arg\max_{\alpha} \sum_{j=1}^{J^{(k)}} A_j^{(k)} \cos\left(\varphi_j^{(k)} - j\alpha\right) \qquad (8)$$

Thus, the linear phase term is considered to be zero near the maximum of the waveform defined by the measured harmonics. Note that this strategy is similar to the one followed in some pitch-synchronous systems in which the two-period-length frames are separated using the signal maxima as reference. The underlying assumption is that the waveform reaches its maximum when the phases of the harmonics are maximally close to zero. The maximum of the summation is one of the zeros of its derivative:

$$\sum_{j=1}^{J^{(k)}} \left[jA_j^{(k)} \sin\varphi_j^{(k)} \cos(j\alpha) - jA_j^{(k)} \cos\varphi_j^{(k)} \sin(j\alpha) \right] = 0 \qquad (9)$$

The resulting equation is nonlinear, but it can be simplified using the substitution x=cosα and the Tsebyshev polynomials, defined recursively as:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \qquad (10a)$$

$$U_0(x) = 1, \quad U_1(x) = 2x, \quad U_n(x) = xU_{n-1}(x) + T_n(x) \qquad (10b)$$

These polynomials verify the following conditions:

$$\cos j\alpha = T_j(x) \qquad (11a)$$

$$\sin j\alpha = \sin\alpha \cdot U_{j-1}(x) = \pm\sqrt{1-x^2} \cdot U_{j-1}(x) \qquad (11b)$$

so equation (9) can be transformed into

$$P(x) \pm \sqrt{1-x^2}\,Q(x) = 0 \qquad (12)$$

where P and Q contain the weighted sum of T-type and U-type polynomials, respectively. The solutions of (12) are also solutions of

$$P(x)^2 - \left(1-x^2\right)Q(x)^2 = 0 \qquad (13)$$

Among all the solutions of (13), which are easily located by any typical root finding method between x=–1 and x=1, the one that maximizes (8) is chosen and its corresponding α is calculated.

In practice, not all harmonics need to be used for the calculation of the linear phase term. Only the most powerful harmonics are relevant for this task, so the complexity of the problem can be reduced by selecting only those harmonics.

It must be taken into account that the polarity of the signals is not always the same. This algorithm is designed for signals in which the positive peaks are greater than the negative peaks. In the other case, equation (8) should be minimized instead of maximized.

4. Concatenation of Units

In concatenative speech synthesis, the synthetic utterances are built by concatenating different speech units selected from a recorded database. The prosodic contour of the units is adapted to the desired specifications, given by a prosody generation block whose input is the text to be pronounced by the system. The algorithm for concatenation of units recorded in different phonetic contexts has to minimize the waveform discontinuities and the spectral mismatches at the boundaries.

In order to develop a waveform generation block using the HSM, the whole database has to be analyzed and parameterized according to the model. Once the prosody of the selected units is modified, the waveform discontinuities are avoided by correcting the linear phase term of the incoming unit to be coherent with the previously concatenated units. Let k^A be the last frame of the last unit concatenated A, and k^B the first frame of the incoming unit B. The phase correction is given by the following expressions:

$$\Delta\varphi_1^{AB} = \alpha^{(k_A)} - \alpha^{(k_B)} + \Psi\left(f_0^{(k_A)}, f_0^{(k_B)}, N\right) \qquad (14a)$$

$$\varphi_j'^{(k)} = \varphi_j^{(k)} + j\Delta\varphi_1^{AB}, \quad k \geq k_B \qquad (14b)$$

where α is calculated using equation (8). It must be emphasized that in the case of concatenative synthesis the calculation of the linear phase terms is performed only once, when building the synthesis database, so that α is stored together with the rest of signal parameters.

On the other hand, a smoothing technique is applied to the amplitude envelopes of the frames near the unit boundaries, so that the spectral discontinuities are also minimized.

5. Voice Conversion by WFW

In general, voice conversion systems apply a previously trained transformation function to the input signal. In our case, the input signals are synthetic utterances obtained by concatenation of selected units. Thus, the TTS system acts as source speaker. In our system, the Weighted Frequency Warping method (WTW), recently proposed by the author [11], is used for voice conversion. This method has been already tested with natural speech, and the results show that a good balance between quality and conversion degree is obtained. In the framework of the TC-STAR project, our voice conversion system was evaluated in both intra-lingual and cross-lingual contexts, and excellent results were obtained [13]. Although the WFW method is not discussed or evaluated in this paper, the voice conversion algorithm is described in this section in order to offer complete information about the waveform generation process.

5.1. Prosodic Conversion

During the training phase, the mean μ and standard deviation σ of the logf_0 are determined for the source and target speakers. During the conversion phase, given a synthetic utterance generated by the TTS system, the pitch contour is modified to match the specifications of the target speaker according to the following expression:

$$\log f_0^{(\text{converted})} = \mu^{(\text{target})} + \frac{\sigma^{(\text{target})}}{\sigma^{(\text{source})}} \left(\log f_0^{(\text{source})} - \mu^{(\text{source})} \right) \quad (15)$$

5.2. Spectral Conversion

The spectral transformation concerns the amplitudes and phases of the harmonics and the LPC coefficients of the stochastic component. During the training phase, a gaussian mixture model (GMM) of m gaussian components is trained from a set of phonetically aligned source-target acoustic vector pairs $\{[x^{\mathrm{T}}\ y^{\mathrm{T}}]^{\mathrm{T}}\}$ [12]. The joint source-target GMM is represented by the weights $\{\alpha_i\}$, the mean vectors $\{\mu_i\}$ and the covariance matrices $\{\Sigma_i\}$ of each of the gaussian components. In this work, the training vectors $\{x\}$ and $\{y\}$ contain the line spectral frequencies (LSF) that represent the all-pole filter that better fits the amplitudes of the harmonics. Once the GMM has been trained, given a source LSF vector x, the probability that x belongs to the i^{th} gaussian component of the model, $p_i(x)$, is given by

$$p_i(x) = \frac{\alpha_i N\left(x, \mu_i^x, \Sigma_i^{xx}\right)}{\sum_{j=1}^{m} \alpha_j N\left(x, \mu_j^x, \Sigma_j^{xx}\right)} \qquad (16)$$

where μ_i^x and Σ_i^{xx} can be extracted from μ_i and Σ_i, respectively.

$$\mu_i = \begin{bmatrix} \mu_i^x \\ \mu_i^y \end{bmatrix} \quad \Sigma_i = \begin{bmatrix} \Sigma_i^{xx} & \Sigma_i^{xy} \\ \Sigma_i^{yx} & \Sigma_i^{yy} \end{bmatrix} \qquad (17a, b)$$

Using the information provided by the GMM, a different frequency warping function $W_i(f)$ is calculated for each gaussian component i between μ_i^x and μ_i^y. As they both are LSF vectors, the formants given by their corresponding all-pole filters are used as reference points for a piecewise linear frequency warping function. Finally, to conclude with the training procedure, a new function is designed to predict the stochastic component of the target speaker from the LSF representation of its harmonic component. The stochastic LPC coefficients associated with each of the training LSF vectors $\{y\}$ are also translated into LSF vectors $\{y_{st}\}$, and matrices $\{\Gamma_i\}$ and vectors $\{v_i\}$ are found so that the following prediction function is optimized:

$$y_{st} = \sum_{i=1}^{m} p_i(y) \cdot \left[\eta_i + \Gamma_i \left(\Sigma_i^{yy} \right)^{-1} \left(y - \mu_i^y \right) \right] \qquad (18)$$

where μ_i^y and Σ_i^{yy} are used in equation (16) to obtain $p_i(y)$. At the end of the training phase, the GMM parameters, the frequency warping functions and the stochastic prediction function have been calculated.

In the conversion phase, given a source frame to be converted, the associated LSF vector x is extracted from the amplitudes of the harmonics, and the m probabilities $p_i(x)$ are calculated using expression (16). The individual warping function of the current frame is obtained as a linear combination between the m trained basis functions $W_i(f)$.

$$W(f) = \sum_{i=1}^{m} p_i(x) \cdot W_i(f) \qquad (19)$$

We assume that phonemes with similar formant structures, which are linked to the same gaussian component of the GMM, should be associated with similar frequency warping trajectories. Thus, the probabilities $p_i(x)$ are used as weights for the linear combination of the m different warping trajectories. The magnitude envelope $A(f)$ of the current frame is estimated by means of a linear interpolation between the measured harmonic log-amplitudes. The phase envelope $\theta(f)$ is estimated by linearly interpolating the real and imaginary parts of the complex amplitudes $A_j^{(k)}\exp(i \cdot \theta_j^{(k)})$, as in section 3.2. Warped envelopes $A'(f)$ and $\theta'(f)$ are calculated, and the target amplitudes $\{A_j'^{(k)}\}$ and vocal tract phases $\{\theta_j'^{(k)}\}$ are calculated by resampling them at the positions of the harmonics.

$$A'(f) = A\left(W^{-1}(f)\right), \quad \theta'(f) = \theta\left(W^{-1}(f)\right) \qquad (20a, b)$$

This step does not completely transform the source voice into the target speaker's voice because the formants are only reallocated while their amplitude remains unmodified. Therefore, the energy distribution is corrected using the converted LSF vector $F(x)$, which is obtained by means of the typical GMM-based transformation function:

$$F(x) = \sum_{i=1}^{m} p_i(x) \cdot \left[\mu_i^y + \Sigma_i^{yx} \left(\Sigma_i^{xx} \right)^{-1} \left(x - \mu_i^x \right) \right] \qquad (21)$$

The energy of the envelope given by $F(x)$ is measured at the bands 100-300Hz, 300-800Hz, 800-2500Hz, 2500-3500Hz and 3500-5000Hz, which are likely to contain different formants. Multiplicative factors are used inside each band to correct the energy of the frequency-warped harmonics. Finally, the stochastic component of the converted frame is predicted using expression (18), in which y is substituted by the converted LSF vector $F(x)$ (21). The stochastic component of the unvoiced frames is left unmodified, because its conversion does not lead to any important improvement and it can cause a small loss of quality.

6. Experiments and Discussion

A preference test was carried out in order to determine if the proposed algorithms were suitable for the development of a high quality speech synthesis system. Ogmios is the speech synthesis system that has been created at the UPC [7]. It is based on unit selection, and it includes a waveform generation block based on the TD-PSOLA technique, which cannot be used for voice conversion but is almost standard for synthesis. For the preference test, the text processing, prosody generation and unit selection blocks of Ogmios were used to obtain the sequence of units and prosodic specifications of the different synthetic utterances, and the audio samples were generated using both Ogmios and a new waveform generation block based on the HSM and the algorithms described in the previous sections.

In order to emphasize the effectiveness of both methods in speech modification and concatenation, the system was forced to modify the prosody of all the selected units to match the specifications provided by the prosody generation block of Ogmios. Under these conditions, the artifacts introduced by both methods were more visible for the comparison, although the quality of the sentences was lower.

The 18 listeners that participated in the test, 6 speech synthesis experts and 12 volunteers, were asked to listen to 17 pairs of synthetic utterances in Spanish. All the listeners were native Spanish speakers. Four different voices were used in this experiment. Two of them, one male and one female, were built from a database consisting of more than 10 hours of recorded speech. The databases of the two remaining voices, male and female, contained less than an hour of recorded speech. 10 of the sentence pairs in the test were generated from the large-database voices, and 7 pairs were built from the small-database voices. For each sentence pair, whose components were played in random order, the listeners were asked to choose between the following options: "I prefer the first", "I prefer the second" or "I can't decide". The results of the preference test are shown in figure 1.

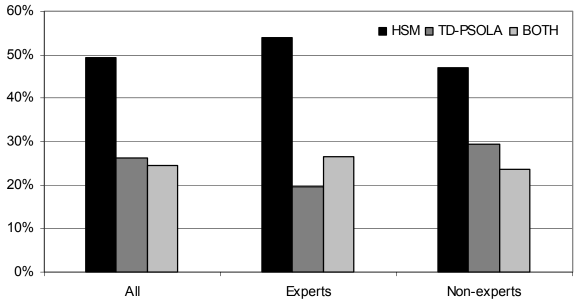


Figure 1: results of the preference test.

Figure 2 shows separately the results for large synthesis databases (a) and for small synthesis databases (b). In figure 3 individual results for female voices (a) and for male voices (b) are displayed separately.

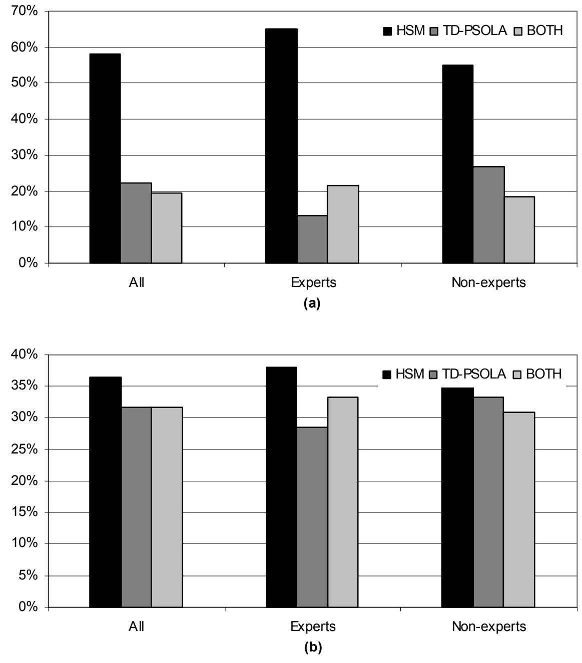


Figure 2: results for large (a) and small databases (b).

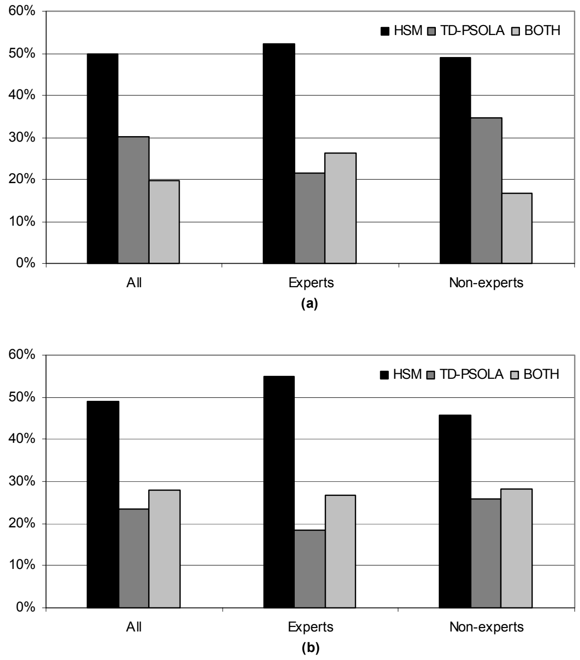


Figure 3: results for female (a) and male (b) voices.

As it can be seen, in the conditions of this experiment the new HSM waveform generation block clearly outperforms the one based on TD-PSOLA. This assertion holds for both expert and non-expert listeners, but the new method is slightly better scored by experts. Concerning figure 2, it can be observed that when the synthesis databases are small, the uncertainty increases and the scores are closer to each other. This fact can be a consequence of the different noise sources in each case. When the databases are large, all the phonemes are represented by a high number of instances. Thus, the prosodic modification factors needed are lower and the associated noise is less important than the artifacts coming from the concatenation of units. The concatenations obtained by means of the HSM algorithms are smoother because the spectral envelopes can be manipulated. On the contrary, when the synthesis database is small, the loss of quality caused by the prosodic modifications and by severe concatenation artifacts affects both methods in a more similar way. Figure 3 shows that the scores reached by the HSM waveform generator are similar in both genders.

The experiment described shows that the HSM method and the algorithms presented in this paper, which have been successfully used for voice conversion purposes, are also suitable for high-quality speech synthesis without voice conversion. The listeners' choices seem to be more influenced by the concatenation properties than by the quality of the prosodic modification. However, the results may be different for other implementations of the unit selection procedure that assign a lower weight to the prosodic aspects of the units and a higher weight to the spectral aspects. In addition, it must be taken into account that in a standard synthesis application not all the units are prosodically modified, and in this situation the TD-PSOLA approach can be expected to reach higher scores because it works directly with the recorded speech samples.

7. Conclusions

In this paper we have presented our improved waveform generation system for speech synthesis based on the harmonic plus stochastic model. The new algorithms for prosodic modification, concatenation and conversion of speech, which in contrast to other methods do not require pitch-synchronism, have been described in detail. The experiments carried out in this paper show that the listeners prefer this new approach to a more standard TD-PSOLA approach. It can be concluded that the algorithms and methods described, which were successfully used for voice conversion applications, are also suitable for high-quality speech synthesis.

In future works voice conversion constraints will be included in the cost function of the unit selection block of a TTS system. It is expected that the performance of the synthesis + conversion system will be improved if the units that are easier to convert are assigned a higher probability to be selected for synthesis.

8. Acknowledgements

This work was partially supported by TC-STAR (Technology and Corpora for Speech-to-Speech Translation, FP6-506738) and AVIVAVOZ (TEC2006-13694-C03).

9. References

[1] Erro, D., Moreno, A., "A pitch-asynchronous simple method for speech synthesis by diphone concatenation using the deterministic plus stochastic model", *Proc. 10th Int. Conf. on Speech and Computer*, pp.321-324, 2005.

[2] Stylianou, Y., "Harmonic plus Noise Models for Speech, combined with Statistical Methods, for Speech and Speaker Modification", PhD thesis, École Nationale Supérieure des Télécommunications, 1996.

[3] Moulines, E., Charpentier, F., "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones", *Speech Communication*, vol.9 no.5-6 pp.453-467, 1990.

[4] Quatieri, T.F., McAulay, R.J., "Shape invariant time-scale and pitch modification of speech", *IEEE Transactions on Signal Processing*, 1992.

[5] O'Brien, D., Monaghan, A.I.C., "Concatenative synthesis based on a harmonic model", *IEEE Transactions on Speech and Audio Processing*, 2001.

[6] Chazan, D., Hoory, R., Sagi, A., Shechtman, S., Sorin, A., Shuang, Z.W., Bakis, R., "High quality sinusoidal modeling of wideband speech for the purposes of speech synthesis and modification", *ICASSP*, 2006.

[7] Bonafonte, A., Agüero, P.D., Adell, J., Pérez, J., Moreno, A., "OGMIOS: The UPC text-to-speech synthesis system for spoken translation", *TC-Star Workshop on Speech to Speech Translation*, 2006.

[8] Depalle, Ph., Hélie, T., "Extraction of spectral peak parameters using a STFT modeling and no sidelobe windows", *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1997.

[9] McAulay, R.J., Quatieri, T.F., "Speech analysis/synthesis based on a sinusoidal representation", *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1986.

[10] Stylianou, Y., "Removing linear phase mismatches in concatenative speech synthesis", *IEEE Transactions on Speech and Audio Processing*, 2001.

[11] Erro, D., Moreno, A., "Weighted Frequency Warping for Voice Conversion", *InterSpeech*, 2007.

[12] Kain, A., "High resolution voice transformation", PhD thesis, OGI School of Science and Engineering, 2001.

[13] Choukry, K., et al. Evaluation Report. Deliverable D30 of the EU funded project TC-STAR. March 2007.

Prosody Modelling in Czech Text-to-Speech Synthesis

Jan Romportl, Jiří Kala

Department of Cybernetics, Faculty of Applied Sciences
University of West Bohemia, Pilsen, Czech Republic
rompi@kky.zcu.cz, jkala@kky.zcu.cz

Abstract

This paper describes data-driven modelling of all three basic prosodic features – fundamental frequency, intensity and segmental duration – in the Czech text-to-speech system ARTIC. The fundamental frequency is generated by a model based on concatenation of automatically acquired intonational patterns. Intensity of synthesised speech is modelled by experimentally created rules which are in conformity with phonetics studies. Phoneme duration modelling has not been previously solved in ARTIC and this paper presents the first solution to this problem using a CART-based approach.

1. Introduction

Concatenative text-to-speech (TTS) synthesis of the Czech language has been researched, elaborated and implemented already for a significant period of time. During this period various prosody models have been proposed, yet at least to our knowledge there has not been implemented and practically applied any complex *data-driven* (in the sense of automatic training using very large real speech databases) prosody model of all three basic prosodic characteristics (i.e. fundamental frequency (F0), intensity and segmental duration altogether).

This paper tries to present such a prosody model implemented in the TTS system ARTIC, developed at the Department of Cybernetics, University of West Bohemia [1]. The model is formally based on a linguistically motivated structural prosody description framework, which explicitly separates prosodic function from its form. The fundamental frequency generation part of the model is based on our data-driven intonation model previously introduced for example in [4], whereas intensity modelling is rule based. The most recent advance presented in this paper consists in incorporating a CART-based duration model trained on a large speech corpus.

2. Prosody description framework

The prosody model used in TTS system ARTIC is based on explicit distinction between prosodic form and function. The importance of such a form of linguistic stratification has already been frequently discussed (let us at random mention for instance [2]).

2.1. Prosodic form and function

In our conception each input sentence is represented in form of a prosodic structure. The prosodic structure is a result of parsing a sentence using a specific set of linguistically motivated transformation rules collectively called *prosodic grammar*. The prosodic structure of a sentence formally corresponds to a prosodic function while a prosodic form (i.e. how prosody is eventually realized by acoustic means – "surface" prosody) is then derived from it (i.e. the allowed prosodic forms depend purely on the prosodic function together with phonotactics restrictions, not on the text or sentence itself).

In other words – the prosodic structure determines a parameterisation of input text and this parameterisation is then used in a system for prosodic form assignment (i.e. a classifier, knowledge base, unit selection algorithm, etc.). It is not a goal of this paper to fully describe the prosodic structures and grammar – the discussion on this topic can be rather found in [3]. The following paragraphs just briefly summarise some information necessary as a background for our TTS prosody model.

2.2. Prosodic grammar

The prosodic grammar tries to capture structuring of a sentence relevant for prosody functioning. Using generative-based rules it decomposes a sentence into its immediate constituents (terminals and non-terminals) and mutual relations between these constituents formalise the prosodic function. The grammar (or rather its equivalent Chomsky's normal form) is designed to be implemented in a stochastic grammar parser, which is now being developed and tested. We distinguish the following language units serving as the grammar terminal and non-terminal constituents (parenthesised symbols are used in the respective grammar rules):

Prosodic sentence (PS)

Prosodic sentence is a prosodic manifestation of a sentence as a syntactically consistent unit, yet it can also be unfinished or grammatically incorrect.

Prosodic clause (PC)

Prosodic clause is such a linear unit of a prosodic sentence which is delimited by pauses. A prosodic sentence generally consists of more prosodic clauses.

Prosodic phrase (PP)

Prosodic phrase is such a segment of speech where a certain intonation scheme is realized continuously. A prosodic clause generally consists of more prosodic phrases.

Prosodeme (P0), (Px)

Prosodeme is an abstract unit established in a certain communication function within the language system. We have postulated that any single prosodic phrase consists of two prosodemes: so called "null prosodeme" and "functionally involved prosodeme" (where (Px) stands for a type of the prosodeme chosen from the list shown below), depending on the communication function the speaker intends the sentence to have. In the present research we distinguish the following prosodemes (for the Czech language; other languages may need some modifications):

- P0 – null prosodeme
- P1 – prosodeme terminating satisfactorily (a reply is not expected)
  - P1-1 unmarked
  - P1-2 marked directive
  - P1-3 marked expressive
  - P1-4 specific
- P2 – prosodeme terminating unsatisfactorily (a reply is expected)
  - P2-1 unmarked (supplementary, “wh-questions”)
  - P2-2 marked declaratory (“yes/no questions”)
  - P2-3 marked disjunctive (questions with disjunctive “or”)
  - P2-4 specific
- P3 – prosodeme nonterminating
  - P3-1 unmarked
  - P3-2 marked bound (involved in a function primarily held by P1 or P2)
  - P3-3 specific

Prosodic word (PW)

Prosodic word (sometimes also called phonemic word) is a group of words subordinated to one word accent (stress). Languages with a non-fixed stress position would need a stress position indicator too.

Semantic accent (SA)

By this term we call such a prosodic word attribute, which indicates the word is emphasised (using acoustic means) by a speaker.

There are two more terminal symbols used (“$” and “#”) standing for pauses differing in their placement (inter- and intra-sentential). The terminal symbol *(w_i)* stands for a concrete prosodic word from a lexicon and ∅ means an empty terminal symbol. Note that *Px* is only an “abbreviation” for each prosodeme (i.e. P1-1, etc.). The rules should be understood this way: “(PC) → (PP) {1+} # {1}” means that the symbol *(PC)* (prosodic clause) generates one or more *(PP)* symbols (prosodic phrases) followed by one # symbol (pause).

(PS) → (PC) {1+} $ {1}

(PC) → (PP) {1+} # {1}

(PP) → (P0) {1} (Px) {1}

(P0) → ∅

(P0) → (PW) {1+}

(Px) → (PW) {1}

(Px) → (SA) (PW) {1+}

(PW) → w_i {1}

Figures 1 and 2 show two possible prosodic structures of the Czech sentence: “It is not a singular transformation of a long vowel into a diphthong.” However, the second variant bears a semantic accent on the word “singular” so as to bring forward the contrastive focus as the opposite of e.g. “frequent”.

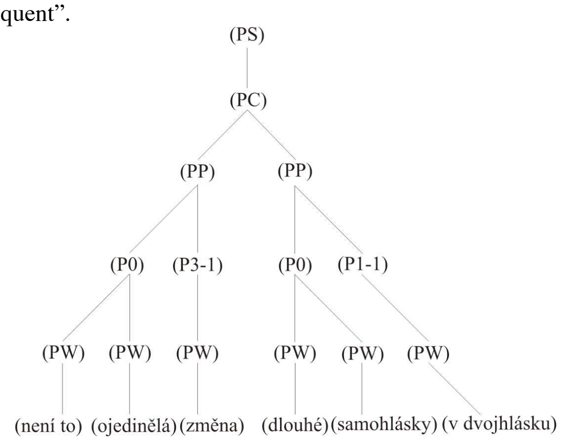


Figure 1: Czech sentence prosodic structure in a neutral form.

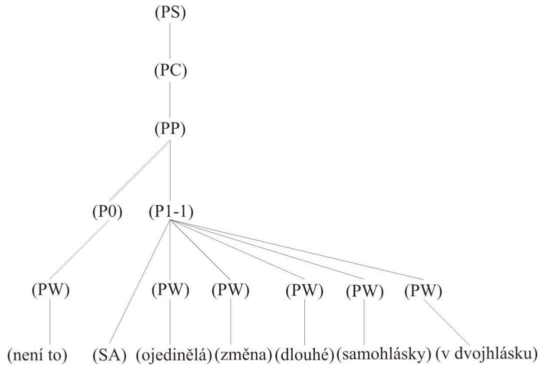


Figure 2: Czech sentence prosodic structure with a semantic accent.

It is not a simple task to infer the full prosodic structure from the surface form of a sentence. This can be done using a probabilistic grammar parser similar to a parser used for syntax analysis – on one hand the prosodic parser is simpler due to far less complex grammar, but on the other hand the relations among prosodic constituents are not as clear and straightforward as among syntactic constituents (in case of prosody many phenomena are facultative, singular or even random). Hence the goal of the prosodic parser is not to create couple of “definitely correct” prosodic structures of a given sentence; rather it should delimit a class of prosodic structures acceptable in a given context.

Because of such peculiarities we have not yet implemented fully working automatically trained parser into ARTIC and the task of prosodic structure parsing is carried out by a set of heuristic rules. These rules are obviously far from performing optimally (for example they are very inaccurate in prosodic phrase detection and semantic accents have to be omitted at all) but they are treated as a temporary solution.

3. F0 modelling

It is beyond the scope of this paper to fully describe the data-driven model of F0 implemented in ARTIC – more information on this (including the model evaluation) can be found in [4]. However, the basic idea is in conformity with the aforementioned considerations about duality of prosodic form and function.

From the formal point of view all information about prosodic function of each word is encoded in the prosodic structure itself and hence the position of the word within the structure. Therefore the prosodic form realised by means of F0 behaviour depends purely on positions of the prosodic words within the prosodic structure of a given sentence.

The position of a prosodic word ("position" not in the exact meaning – rather we would use it in the sense of mutual configuration between prosodic words and their parent prosodic constituents) is described by a set of features (we refer to it as description array – DA) which include for instance: index of the prosodic word within its neighbours with the same parent node, type of its parent node and its index (and this recursively up to the root node), and also various quantitative features concerning syllabic, stress and phoneme structure of the word. More details on DA can be found in [4].

The relation between prosodic function (formulated through DA) and its form is represented by a function in the mathematical sense, which we refer to as *realization function* (because it realizes the function through the form). The realization function is created from a suitable speech corpus (ideally the same one used for a particular speech segment database creation) with transcribed utterances, prosodic structure tags (i.e. the transcribed sentences are prosodically parsed) and F0 contours (e.g. acquired by electroglottograph measuring). Speech must be segmented at least on the level of prosodic words (i.e. time intervals of prosodic words must be known).

The F0 contours are segmented according to the prosodic words – this way the F0 contour of each prosodic word token is acquired (let us call such a segment a *sub-contour*). The corpus used in ARTIC consists of 5,000 sentences involving 55,655 sub-contours which are then clustered into so called *cadences* (abstract intonational patterns – as will be described further in the text).

3.1. Realization function

The realization function is defined as

$$R: DA \rightarrow I \times pot(C)$$

where $I = \{i_1, \ldots, i_l\}$ is a set of initial conditions, $C = \{c_1, \ldots, c_m\}$ is a set of cadences and $pot(C)$ is a power set of C. A cadence is an intonational pattern which fits into an interval of a single prosodic word. The set C can also be called a cadence inventory. Initial conditions say where on the frequency scale a cadence chosen for a prosodic word starts.

Fujisaki shows [5] that F0 can be modelled in a logarithmic space as a sum of outputs of two linear systems. In the linear space this summation corresponds to a multiplication of values, therefore each sub-contour (as a segment of a whole F0 trajectory) acquired from the corpus can be decomposed into two components: (a) the initial F0 value of the sub-contour; (b) the rest of the sub-contour relatively to the initial value (in its multiples).

The realization function also consists of two components. The first one is constructed from the corpus by linking each DA occurring in the corpus with the initial F0 value of the respective sub-contour occurring with this DA in the corpus. Since a particular DA is often assigned to several prosodic word tokens in the corpus, there are usually more possible initial value links. In such cases the first sub-contour with a given DA occurring in the corpus (supposing indeed arbitrary, yet constant sentence numbering) is considered – this ensures the synthesised prosodemes to be intonationally "consistent" as for the prosodic word initial conditions because the initial F0 values of the prosodic words within a particular synthesised prosodeme are all selected from the same sentence (otherwise it could happen that each initial condition in the synthesised prosodeme is selected from a different sentence, although with the same DA).

The set $C = \{c_1, \ldots, c_m\}$ (the cadence inventory) is created by a clustering algorithm based on repeated bisections and cosine similarity function, applied on all F0 sub-contours from the corpus. Prior to this, the sub-contours are represented by vectors with the dimension x (i.e. by approximating each sub-contour with x equidistant points relatively to its initial value – this ensures sub-contour normalisation over time intervals and F0 values). The elements of C (i.e. cadences) are constructed as either centroids of the clusters, or there is one (or more) vector chosen from each cluster as its representative (using various methods, such as elimination of outliers according to Mahalanobis distance).

We have experimented with various values of m (the number of cadences) ranging from 3 to 200. Good results are achieved for example with the number of clusters m=30. In this case the smallest cluster consists of 911 vectors (sub-contours) and the largest of 3571. The cadence inventory is created from the cluster centroids.

We say a cadence *belongs* to a particular DA provided that the sub-contour occurring in the corpus with this DA is an element of the cluster represented by the given cadence. The second component of the realization function is constructed from the corpus by linking each DA occurring in the corpus with the set of all cadences belonging to this DA. Thus if we have a prosodic word w_j, then

$$R(DA(w_j)) = <i_j, C_j>$$

where $i_j \in I$ is the assigned initial condition and $C_j \subseteq C$, $C_j = \{c_{j,1}, c_{j,2}, \ldots, c_{j,lj}\}$ is a set of the assigned cadences. Now let the synthesised sentence S be given as:

$$S{:}\;\; w_1\, w_2 \ldots w_p$$

The resulting generated F0 contour of the sentence S is then constructed from the initial conditions and cadences given by the realization function for each prosodic word w_1, … w_p – the initial conditions are F0 values at the beginnings of the prosodic words and the cadences actually fill the gaps between neighbouring initial conditions by F0 values calculated as multiples of the initial conditions. As it can be seen from the definition of the realization function, the set of several suitable cadences is given for each prosodic word – only one of them must be chosen at a time. This is done by a criterion function, minimised over all combinations of proposed cadencies. One of the choices for the criterion function is for example a sum of differences of F0 values on the boundaries of the prosodic words – to avoid or at least minimise F0 discontinuities in junctures where one cadence ends and the next one (based on a different initial condition) starts. This process of cadence concatenation is described together with the criterion function in more detail in [4].

3.2. Prosodic homonymy

One can easily see no corpus can offer all possible DAs and therefore it is impossible to construct the realization function ideally. Hence the crucial importance for the realization function has the *relation of indistinguishableness* [4]. Two description arrays are in the relation of indistinguishableness provided that their different deep prosodic-semantic functions can be realized by the same functor (i.e. same surface prosodic means) – two different DAs are homonymous in terms of their surface realization and thus mutually interchangeable. Informally: the realization function is defined also for those possible DAs not occurring in the corpus; namely if a set of appropriate cadences is to be determined for a DA not occurring in the corpus, another DA which occurs in the corpus and is homonymous according to the aforementioned relation, is taken instead and the set of cadences and initial conditions is determined for the new DA.

A question is how to determine the relation of indistinguishableness. The best method is probably an automatic analysis of heldout corpus data – this presupposes that the heldout data include DAs not occurring in the training data (i.e. factually unobserved) and the relation of indistinguishableness can be determined by a feasible generalisation of the mutual relation between the training and heldout data. This generalisation can be formalised for instance by a specific DA space metrics which allows to find a homonymous DA in terms of the minimum vector distance.

However, research in this field has not been successfully finished yet and thus our TTS system ARTIC must now settle for a workaround in the form of performing a number of limited perturbations of the least significant (heuristically and experimentally determined) components of an unobserved DA (e.g. exact length of a prosodic word in phonemes, exact number of prosodic clauses in a sentence, etc.) which eventually transform the unobserved DA into such a DA that occurs in the corpus and is very likely to be still homonymous.

4. Intensity modelling

It has been often discussed in Czech phonetics literature that intensity (or loudness – as a psychological correlate of intensity) is of far less importance than fundamental frequency with respect to suprasegmental features of speech, therefore our prosody model pays significantly less attention to it.

Moreover, we have undertaken theoretical considerations of modelling intensity analogically to fundamental frequency, i.e. by "intensity cadencies". However, since intensity is much more interconnected with segmental qualities of speech, the application of such a model is not as straightforward as in the case of fundamental frequency (intensity can be treated as sort of a distinguishing feature of a phoneme, unlike F0 which is basically present at voiced phonemes and not present at unvoiced phonemes).

Considering the aforementioned, our prosody model currently incorporates only a simple rule for intensity modelling. Czech phonetics studies usually mention some increase of intensity (or perceived loudness) on stressed syllables. We have experimentally revealed that linear increase of speech signal amplitude by 1.3 on stressed syllables is well assessed by listeners evaluating the resulting synthesised speech. This is in conformity with [6] stating that stressed syllables usually feature increase of intensity level by 1 – 3 dB.

5. Segmental duration modelling

All previous versions of our prosody model did not comprise any explicit duration modelling techniques and have been using only average lengths of phonemes from segmented speech corpus. However, in our recent research we have incorporated and implemented a Classification and Regression Tree (CART) approach for segmental duration modelling, mainly because of possibility of its straightforward application and rich experience of other research teams. Our experiments are similar to [7], [8] but there is one important difference – we do not use only one regression tree for all phonemes, rather we have trained an independent tree for each phoneme (experiments with a single universal tree have reached worse score for us).

5.1. Training data

Training data for tree construction consists of 5,000 indicative sentences recorded by a female voice talent (the same data have been used also for the acoustic unit inventory creation and for fundamental frequency modelling). These recordings have been automatically segmented by a statistical approach (HMM-based). Resulting inventory counts over 400,000 phonemes where each of them has been represented by 172 features (as it is described further).

5.2. Phoneme features

For the sake of the CART-based classification each phoneme token (i.e. occurrence of a phoneme) is represented (or described) by a set of 172 features which can be methodologically divided into five groups. Since an independent tree is built for each phoneme type (the word "type" is used here in the sense of commonly understood duality "token/type" – "type" is the phoneme itself and "token" its textual occurrence), the phoneme type itself is not included among the features.

5.2.1. Basic feature groups

These groups of features are derived from phoneme types of neighbouring phonemes and their categorisation into phoneme classes such as vowel, consonant, fricative, plosive, etc.

Features defined by neighbour type form the first group:

- **previous_type/next_type** – the type of the previous/next phoneme. If the phoneme stands as the first/last one in a sentence, the symbol "_" (underscore) is used as a value of this feature.
- **previous2_type/next2_type –** the type of a phoneme which stands over one phoneme before/after. Identically as in the previous case the underscore symbol is used in case the type of the phoneme cannot be obtained.

The second group is based on membership of a phoneme type into specified phoneme classes. The classes are distinguished by various articulatory and phonational criteria (e.g. vowel quantity, sonority, articulation place and manner, etc.). Values of the features are either true or false – depending on whether a phoneme type is or is not a member of the given class.

5.2.2. Feature groups based on prosodic grammar

The next feature groups describing phonemes are based on the prosodic grammar described in Section 2 of this paper (although not all grammar attributes are used). Every sentence is thus structured hierarchically into the constituents resulting from the prosodic grammar, i.e. prosodic sentence, prosodic clause, prosodic phrase, prosodeme, prosodic words – and in addition to them – syllables and phonemes.

The constituents are hierarchically sorted from the parent ones down to their children. Each of them contains one or more child elements. For example every phoneme stands somewhere in a syllable and each syllable contains one or more phonemes; a syllable stands in a prosodic word and each prosodic word contains one or more syllables.

Features in the third group have their values derived from the "length" of a prosodic sentence constituent in the phoneme token context. This length is determined for each constituent by the number of its child constituents (the number of phonemes in a syllable, syllables in a prosodic word, etc.).

The fourth group consists of features which indicate the position of a child constituent within its parent constituent in the phoneme token context – from the beginning and from the end of the parent constituent (the numeric representation is used). Again, not just the position of the constituent within its immediate parent is used, but the positions in the whole parent hierarchy are taken into account as well.

The last group of features is similar to the previous one with the difference that the values are not represented by numbers, but positions are categorised into these possibilities:

- FIRST/LAST – the child is positioned within its parent as the first/last one (from beginning)
- MIDDLE – in other cases

5.3. Training process

The duration model training has been carried out using the *wagon* CART building program, a part of the Edinburgh Speech Tools Library. Root mean squared error (RMSE) and correlation coefficient (CORRC) values, presented in the evaluation further in this paper, have been therefore computed by *wagon*.

Since our segmented speech data contain more than 400,000 phoneme tokens, there are enough occurrences of each phoneme type and thus we have decided to train individual regression tree for each phoneme type.

The first 80 percent of sentences from the whole corpus have formed a training set and the rest of the data then has been used for testing.

5.4. Experiments

Several training and evaluation experiments have been carried out. The very first training experiments used only some of the features from the groups described in Section 5.2. However, due to poor results the feature set has then been extended to the final number of 172 features.

As described in the text above, an independent tree for each phoneme type is used, therefore the phoneme duration estimator is built as a composition of all individual regression trees where the root (i.e. first) questions is about the phoneme type. After that the algorithm continues in a standard way.

In one of the training experiments the features based on phoneme classes were excluded. However, this way we have reached too high values of RMSE and CORRC (see Table 1) and thus the approach had to be improved. The next couple of experiments were characterised by leaving out the features based on the position and then also on the categorised position because of our hypothesis these features are strongly correlated. The results of these two experiments were very similar and – most importantly – worse than without excluding any features.

The next step consisted in adding the features based on neighbour phoneme type and because this way we have achieved better results, we have expanded the feature set to the full form described hereinbefore. The results achieved by such classifier and feature configurations eventually reached the applicable level and are comparable to results presented by other reports [9], [10], [11].

Since our speech corpus segmentation is based on a statistical approach (HMM) and not conducted by human experts, it sometimes can happen that segment boundaries are placed relatively far from the position where they should be. To prevent these errors from negatively influencing segmental duration estimation we have tried to eliminate them from the training data by excluding phoneme tokens with statistically improbable duration. We have experimentally set this statistical relevance so that only phoneme tokens with duration between 5 and 95 percent fractile (computed for each phoneme type independently) have been included into the training data (sort of a "fractile pruning"). This way we have achieved the best results in terms of the values of RMSE and CORRC.

We have also performed calculation of RMSE and CORRC for a "dummy" duration estimator previously used in our system which gives each phoneme token the length equal to the average length of the respective phoneme type computed from the training data (i.e. actually no estimator because each occurrence of a certain phoneme type has the same length). The results of this experiment are quite important and illustrative since they give an idea of the theoretically lowest acceptable classifier performance. They are presented in the Table 1 as well.

5.5. Evaluation

The first aspect of evaluation of the phoneme duration estimator is mathematical (or rather quantitative). RMSE and correlation coefficient values of the previously described approaches are presented in the following table.

Approach	RMSE	CORRC
“dummy” estimator	24,47	0,85
excl. neighbour token classes	28,39	0,77
all features	22,56	0,75
all features – fractile pruning	18,89	0,92

Table 1: Duration model performance assessment

In comparison with results reported by other studies based on CART (see the Table 2), our experiments have come out slightly better (as for RMSE and CORRC). One cannot judge (concerning current research and evaluation methodology and techniques) whether this is a language or even speaker dependent phenomenon, or our set of features performs really better (the influence of the language is indubitable – e.g. more conservative duration behaviour in the Czech language in comparison with English). However, our model is still not in its final version and we will continue to analyse the results in more detail.

Language [source]	RMSE	CORRC
German [9]	22,71	0,83
English [10] (voice *lja*)	21,00	0,78
English [10] (voice *rjs*)	20,00	0,80
English [10] (voice *erm*)	24,00	0,82
Korean [11]	26,48	0,73
Czech [7]	20,30	0,79
Czech – this paper	18,89	0,92

Table 2: Results comparison with other studies

The second, for our work actually more important aspect of the evaluation is overall quality of produced synthetic speech. We have not yet carried out formal inter-subjective listening tests which quantitatively represent perceptional difference between the baseline “dummy” estimator and the evaluated one. However, according to informal judgement based on listening to synthesised sentences our CART estimator with all features and fractile pruning performs same or better than the baseline technique.

6. Conclusion

The research concerning F0 modelling is currently focusing mainly on the issues connected with prosodic homonymy. We have been able to prove that the current version of synthesised intonation is very well assessed and we expect that further improvement of prosodic structure parsing brings in more naturalness, especially in the field of semantic coherence of the synthetic speech. The presented approach in duration estimation has also performed well in our case and future work in this area will involve mainly more precise perceptual evaluation and also accuracy improving.

7. Acknowledgements

This research is supported by the Ministry of Education of the Czech Republic, project LC 536, Grant Agency of the Czech Republic, project 102/05/0278, and 6th FP IST-034434 COMPANIONS.

8. References

[1] Matoušek, J., Tihelka, D. and Romportl, J., “Current state of Czech text-to-speech system ARTIC”, *LNAI Vol. 4188*, Springer, Berlin, 2006, p. 439 – 446.

[2] Hirst, D. J., “Form and function in the representation of speech prosody”, *Speech Communication Vol. 46,* 2005, p. 334 – 347.

[3] Romportl, J., Matoušek, J., “Formal prosodic structures and their application in NLP”, *LNAI Vol. 3658*, Springer, Berlin, 2005, p. 371 – 378.

[4] Romportl, J., “Structural data-driven prosody model for TTS synthesis”, *Proceedings of Speech Prosody 2006*, *Studientexte zur Sprachkommunikation, Vol. 40*, Dresden, 2006, p. 549 – 552.

[5] Fujisaki, H., “A note on the physiological and physical basis for the phrase and accent components in the voice fundamental frequency contour”, New York, 1988.

[6] Ptáček, M., “Akustika řeči”, Praha, 1996 (“Speech acoustics”, in Czech).

[7] Batůšek, R., “A duration model for Czech text-to-speech synthesis”, *Proceedings of Speech Prosody 2002*, Aix-en-Provence, p.167 – 170.

[8] Öztürk, Ö., Çiloğlu, T., “Segmental duration modeling in Turkish”, *LNAI Vol. 4188*, Springer, Berlin, 2006, p. 669 – 676.

[9] Reidi, M. P., “Controlling Segmental Duration in Speech Synthesis Systems”, dissertation thesis, Zurich, 1998.

[10] Goubanova, O., King, S., “Predicting consonant duration with Bayesian belief networks", *Proceedings of InterSpeech 2005*, 2005, p.1941 – 1944.

[11] Chung, H., Huckvale, A. M., “Linguistic factors affecting timing in Korean with application to speech synthesis", *Proceedings of Eurospeech 2001*, 2001.

Measuring Attribute Dissimilarity with HMM KL-Divergence for Speech Synthesis

Yong ZHAO[1]*, Chengsuo ZHANG*[2]*, Frank K. SOONG*[1]*, Min CHU*[1] *and Xi XIAO*[2]

[1] Speech Group, Microsoft Research Asia, China
[2] Department of Electronic Engineering, Tsinghua University, China
{yzhao, frankkps, mchu}@microsoft.com

Abstract

This paper proposes to use KLD between context-dependent HMMs as target cost in unit selection TTS systems. We train context-dependent HMMs to characterize the contextual attributes of units, and calculate Kullback-Leibler Divergence (KLD) between the corresponding models. We demonstrate that the KLD measure provides a statistically meaningful way to analyze the underlining relations among elements of attributes. With the aid of multidimensional scaling, a set of attributes, including phonetic, prosodic and numerical contexts, are examined by graphically representing elements of the attribute as points on a low dimensional space, where the distances among points agree with the KLDs among the elements. The KLD between multi-space probability distribution HMMs is derived. A perceptual experiment shows that the TTT system defined with the KLD-based target cost sounds slightly better than one with the manually-tuned.

Index Terms: speech synthesis, unit selection, target cost, Kullback-Leibler divergence, HMM, multi-space probability distribution, multidimensional scaling

1. Introduction

Text-to-Speech (TTS) systems based on unit selection feature advantages in synthesizing highly natural and intelligible speech, and have become dominant in commercial applications. These systems rely on a very large database of segmental samples, where the best segment sequence is retrieved for generating speech output with the criterion to minimize a cost function. The cost function is a summation of two sub-cost functions: a concatenation cost, which reflects how well two segments concatenate, and a target cost, as is our interest in this paper, which describes the difference between target and candidate segments.

In the literature, various techniques have been proposed to define the target cost function. A number of approaches presented to minimize the generation error of synthesized speech [1][2], where costs are tuned toward minimizing the distortion of synthetic utterances from their natural counterparts as a reference. Other approaches were based on agreement with human perception [3][4][5], where synthesized utterances are scored subjectively, and costs producing a maximum correlation with subjective scores are regarded as objectively optimal.

Anyhow, in the above approaches, cost functions are optimized by means of synthesizing speech and comparing with sort of criteria. Though these approaches typically lead to a high performance in synthesis by considering all factors, including the process of the synthesis, we lack the ability to reveal the intrinsic proprieties of the target cost.

It is essential that the target cost reflect the difference between units just as human perceives [6]. In this paper, we exploit Kullback-Leibler Divergence (KLD) to estimate the target cost, where we train context-dependent Hidden Markov Models (HMM) to characterize the contextual attributes of units, and calculate KLD between these corresponding models as the distance between units.

One main advantage of the KLD measure is that it allows analyzing the underlining relations among elements of an attribute. It offers a statistically sound way to study the acoustic characteristics of the attributes from varied categories. These categories may involve phonetic, prosodic, linguistic and even paralinguistic.

In this paper we attempt to gain insight of the relations among elements of attributes with the aid of Multidimensional Scaling (MDS). A set of attributes, including phonetic, prosodic and numerical contexts, are examined by graphically representing elements of the attribute as points on a plane or line, where the distances among points agree with the KLDs among elements.

The KLD for a variety of statistical models is presented, including Multi-Space Probability Distribution (MSD) HMM. A subjective evaluation showed the system with KLD-based target cost sounds slightly better than one with manually-tuned.

This paper is organized as follows: Section 2 introduces the concepts of the KLD and its expressions for several statistical models. Section 3 describes how to exploit the KLD as a distance measure of attributes, how to evaluate its effectiveness, and its application as a target cost in unit selection systems. Experiments and discussions are given in Sectioin4 and 5 respectively.

2. Kullback-Leibler Divergence

The KLD between two N-dimensional probability distributions M and $\tilde{M}$ [7] is defined as:

$$D(M \| \tilde{M}) = \int_{R^N} p(X \mid M) \log \frac{p(X \mid M)}{p(X \mid \tilde{M})} dx \qquad (1)$$

KLD describes how far a “true” model M is from an arbitrary model $\tilde{M}$. Note that KLD is asymmetric. If we are not sure which model is correct, we can sum up the integrals in both directions to obtain a symmetrical version of KLD:

$$D_s(M \| \tilde{M}) = D(M \| \tilde{M}) + D(\tilde{M} \| M) \qquad (2)$$

When M and $\tilde{M}$ are Gaussian distribution, $M \sim N(\mu, \Sigma)$ and $\tilde{M} \sim N(\tilde{\mu}, \tilde{\Sigma})$, a closed form KLD is:

$$\begin{aligned} D(M \| \tilde{M}) = \frac{1}{2}[(\mu - \tilde{\mu})^T \tilde{\Sigma}^{-1} (\mu - \tilde{\mu}) + \\ tr(\Sigma\tilde{\Sigma}^{-1}) - \log\left|\Sigma\tilde{\Sigma}^{-1}\right| - N] \end{aligned} \qquad (3)$$

2.1. KLD between HMMs

HMMs are statistical models widely used in speech recognition. In [8], we derived an algorithm to assess the KLD between two general left-to-right HMMs. Given two HMMs H and $\tilde{\mathrm{H}}$ with parameter sets of $\{\pi, A, B\}$ and $\{\tilde{\pi}, \tilde{A}, \tilde{B}\}$ respectively, we approximate the upper bound of a symmetric KLD between two equal-length left-to-right HMMs:

$$D_s\left(\mathrm{H}\middle\|\tilde{\mathrm{H}}\right) \le \sum_{i=1}^{J-1}\Big\{ l_i\left[D\left(b_i\middle\|\tilde{b}_i\right)+\log\left(a_{ii}/\tilde{a}_{ii}\right)\right] + \tilde{l}_i\left[D\left(\tilde{b}_i\middle\|b_i\right)+\log\left(\tilde{a}_{ii}/a_{ii}\right)\right]\Big\} \quad (4)$$

Where $D(b_i \| \tilde{b}_i)$ is the KLD between the observation distributions at state *i*, $\log\left(a_{ii}/\tilde{a}_{ii}\right)$ is the log-likelihood ratio of the transition probability, and $l_i = 1/(1-a_{ii})$ is the expected duration of the i^{th} state in H .

The meaning which equation (4) suggests conforms to our intuition with it: Sum up KLDs over all states and meanwhile, the KLD between states take into account the difference of state transition and observation probabilities, both of which are weighted by the expected state duration.

For KLD between two GMMs $D(b_i \| \tilde{b}_i)$, we use unscented transform to approximate it [9].

2.2. KLD between multi-space probability distributions

Multi-Space Probability Distribution (MSD) was proposed by Tokuda et.al. [10][11]. The HMMs based on MSD are especially useful to model the characteristics of fundamental frequency (F0) in speech, where the voiced part is modeled in a continuous space, and the unvoiced part, a discrete symbol, is looked upon as from zero-dimensional space.

MSD assumes that the observation space Ω is composed of G sub-spaces. Each sub-spaces Ω_g is of n_g dimension and has a prior probability w_g, satisfying $\sum_{g=1}^{G} w_g = 1$. The observation is represented by a random vector o , which consists of two parts, the set of sub-space indices $S(o)$ and a n-dimensional random variable $V(o)$ that is distributed in all sub-spaces specified by $S(o)$. The observation probability of o is defined as:

$$b(o) = \sum_{g \in S(o)} w_g N_g(V(o)) \quad (5)$$

Where $N_g(V(o))$ denotes the probability density of observation $V(o)$ for the g^{th} sub-space.

Consider two MSDs consist of the same sub-spaces and have one-to-one correspondence between sub-spaces, $n_g = \tilde{n}_g$. If all $S(o)$ specify one sub-space, i.e. $|S(o)| \equiv 1$, by calculating KLD in each individual space, we get KLD between two MSDs:

$$D(b \,\|\, \tilde{b}) = D(\mathbf{w} \,\|\, \tilde{\mathbf{w}}) + \sum_{g=1}^{G} w_g D(N_g \,\|\, \tilde{N}_g) \quad (6)$$

where $D(\mathbf{w} \,\|\, \tilde{\mathbf{w}}) = \sum_{g=1}^{G} w_g \log(w_g / \tilde{w}_g)$ denotes KLD between two mixture weight vectors.

If $|S(o)| \ge 1$, i.e. some sub-spaces share their space, they literally form GMMs. We could merge these components into a super-component, and solve it by KLD between GMMs.

To estimate the KLD between MSD-HMMs, we need to substitute equation (6) for $D(b_i \| \tilde{b}_i)$ in equation (4).

3. KLD of Attributes

The characteristics of speech sounds are influenced by not only phonemes in place, but also contextual attributes associated with the sounds. These contextual attributes range from phonetic, prosodic, linguistic, to paralinguistic. In this paper, our research focuses on how to approximate a dissimilarity function of the speech-related attributes. The key concept is that the distances should reflect the differences of the attribute elements in respect of acoustic space. Here we propose to train context-dependent HMMs to characterize the attribute elements, and calculate KLDs between the corresponding models as the dissimilarity function of the attribute.

The first step is to train context-dependent HMMs [12]. By modifying monophones with their contextual attribute of interest, monophone transcriptions are converted to context-dependent phone transcriptions. A set of context-dependent phone models are created by copying monophones and re-estimating. Then, a decision tree based context clustering is applied to tie similar states of context-dependent phones for robust parameter estimation.

Given context-dependent HMMs, KLD is calculated between the models sharing the same central phones as dissimilarities between elements of the attribute. . Such a dissimilarity function is phone-dependent. Besides, we can calculate phone-independent dissimilarity function by averaging phone-dependent functions over all central phones.

For example, given an attribute of interest, monophones are rewritten into context-dependent phones in the form of *c:x*, where *x* is the attribute value of phone *c*. KLD between models $c:x_1$ and $c:x_2$ represents the dissimilarity between attribute element x_1 and x_2 with respect to phone *c*. The dissimilarity between x_1 and x_2 is an average of KLDs over all central phones:

$$D(x_1, x_2) = \frac{1}{N}\sum_{c \in P} D(M(c : x_1) \,\|\, M(c : x_2)) \quad (7)$$

Where N is the size of phoneme set P .

3.1. Graphical interpretation of attribute KLD with multidimensional scaling

While we have in hand a KLD function for attributes, we face the problem how to evaluate the approximation goodness of the KLD measure. [13] evaluated the accuracy of KLD in terms of correlating with the divergences estimated with Monte Carlo simulation. The other approaches [9] examined by means of the performance of applications which employ KLD as the distortion measure in comparison with one without KLD.

In the paper, we adopt multidimensional scaling (MDS) [14] to graphically detect meaningfulness of KLD as dissimilarity measure. MDS is a data analysis technique that represents distances among objects as distances between points of a low-dimensional space, i.e. each object in the domain is represented by a point in the space. The points are arranged in the space so that the distances between pairs of points best approximate the distances between pairs of objects.

MDS helps reveal the underlining relations among objects. This is why we employ MDS to analyze KLD matrix for attributes. Given a KLD matrix of an attribute, elements of the attribute are projected onto a space by MDS. Assuming that KLD is a meaningful measure, elements which are close together in the space should be similar in acoustic characteristics, and elements which are far apart should be dissimilar likewise. On the other hand, if we observe that the relative locations of elements in the space agree with our knowledge with the attribute, we have reasons to believe the effectiveness of the KLD measure.

3.2. KLD as target cost in unit selection

One application of the proposed measure is in unit selection systems. We exploit KLD between context-dependent HMMs as the target sub-cost between target and candidate units. Let t_i and u_i denote the target and candidate unit. The target cost $C^t(t_i, u_i)$ is presented in the form of the sum of the KLDs between context-dependent models:

$$C^t(t_i, u_i) = \sum_{j=1}^{J} w_j^t D(M_j(t_{ij}) \,\|\, M_j(u_{ij})) \tag{8}$$

Where $M_j(t_{ij})$ denotes the model specified by unit t_i in terms of its j^{th} attribute, and w_j^t is the weight of the j^{th} sub-cost.

Note that we assume the target cost is composed of categorical attributes, such as prosodic and prosodic contexts. It holds true in a number of systems [15][16]. When a target cost involves continuous attributes, the KLD measures still work on discrete parts.

Attributes may be in form of compound. That is we take into account the interaction of multiple attributes in the target cost. One advantage of compound attributes is that the efforts to tune weights of the sub-costs w_j^t are reduced. In an extreme situation, we could calculate KLD between HMMs in the context of all attributes as the target cost.

$$C^t(t_i, u_i) = D(M(t_i) \,\|\, M(u_i)) \tag{9}$$

Where $M(t_i)$ denotes the context-dependent model of target unit t_i.

4. Experiments and Results

4.1. Experimental setup

The Microsoft Mulan English speech corpus is used to evaluate the goodness of KLD to approximate acoustic distances for various attributes. The corpus consisted of about 6000 phonetically-balanced sentences recorded by a female voice talent. We manually annotated prosody labels on utterances, such as break levels, stress, and emphasis.

In stage of HMM training, we adopted a topology of 5-state left-to-right HMMs. Features include spectrum and F0 parameters. Spectrum features consist of 39 dimensional feature vectors (13 MFCCs, plus their delta and acceleration coefficients). F0 features consist of log F0, its delta and acceleration coefficients. Similarly, the state distribution consists of two parts: the first part models spectrum features by a single Gaussian distribution with diagonal covariance matrix; the second part models F0 features by an MSD[11]. The MSD is composed of a single Gaussian distribution with diagonal covariance matrix for voiced space and a discrete distribution outputting only one symbol, being unvoiced.

4.2. Contextual attributes

In this paper, the following contextual attributes are taken into account:

- **LPhC**: Left phonetic context. It consists of 40 phonemes. The phoneme set refers to one defined by Microsoft Speech SDK for American English [17].
- **RPhC**: Right phonetic context.
- **PinP**: Position of word in phrase. It takes 9 values. Values are decided by break indices surrounding the word, in the form of *n*-*m*, where *n* is the break index proceeding the word and *m* is the break index following. Values for the break index are chosen from the following set [18]:
    1. Word boundary.
    2. Short phrase boundary.
    3. Intonation phrase boundary.
- **PinW**: Position of syllable in word. It takes 4 values, head of word (H), middle of word (M), tail of word (T), and monosyllable (S).
- **PinS**: Position of phone in syllable.
- **Strs**: Word stress.
- **Emph**: emphasis in phrase.
- **Phns**: Number of phones in syllable. It ranges from 1 to 5. In case of more than 5 phones, set 5.
- **Syls**: Number of syllables in word. It ranges from 1 to 5. In case of more than 5 syllables, set 5.

4.3. Evaluation for phonetic contexts

The first experiment studies the capabilities of the KLD in capturing similarities between phonetic contexts, left phonetic context (LPhC) and right phonetic context (RPhC). Figure 1 and 2 display the planes which are transformed into by MDS from KLD matrices for LPhC and RPhC, respectively. In both graphs, we observe that phonemes, except /h/, are roughly grouped into 3 clusters:

1. Vowel.
2. Sonorant consonant. It consists of semivowels, liquids and nasals.
3. Obstruent. It consists of affricates, fricatives and stops.

Cluster Obstruent can be further subdivided into voiced and unvoiced sounds. Voiced obstruents come closer towards sonorants than does unvoiced.

In the graphs, /h/ stands apart from other phonemes. We credit it to that though phoneme /h/ in English is categorized as voiceless glottal fricative in International Phonetic Alphabet, sometimes it behaves more like a voiceless vowel due to the influence of surrounding vowels.

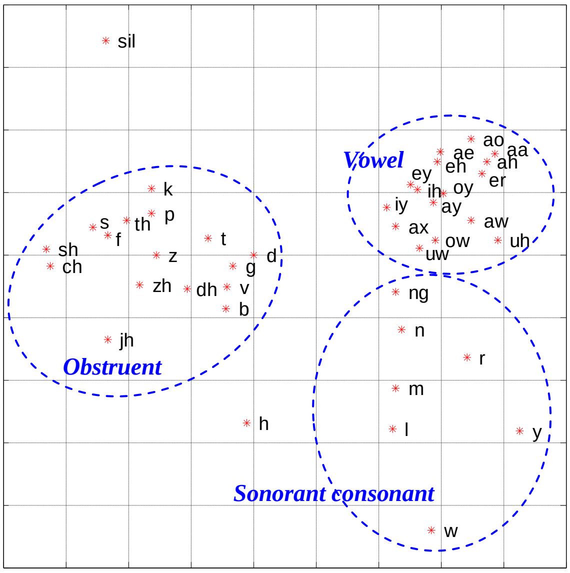


Figure 1. *MDS graph of the KLD matrix for attribute LPhC.*

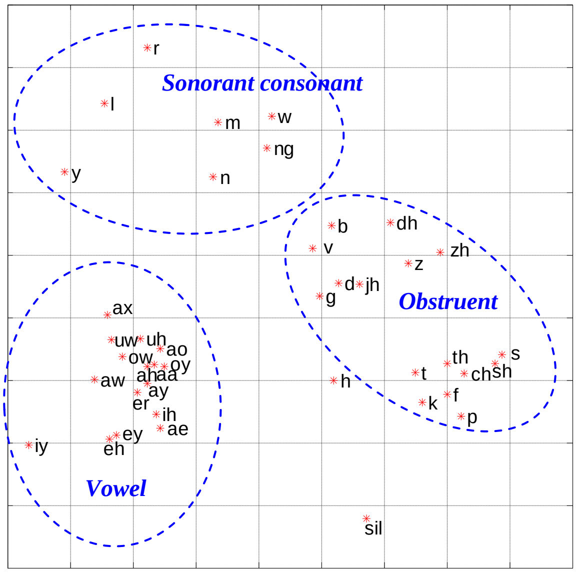


Figure 2. *MDS graph of the KLD matrix for attribute RPhC.*

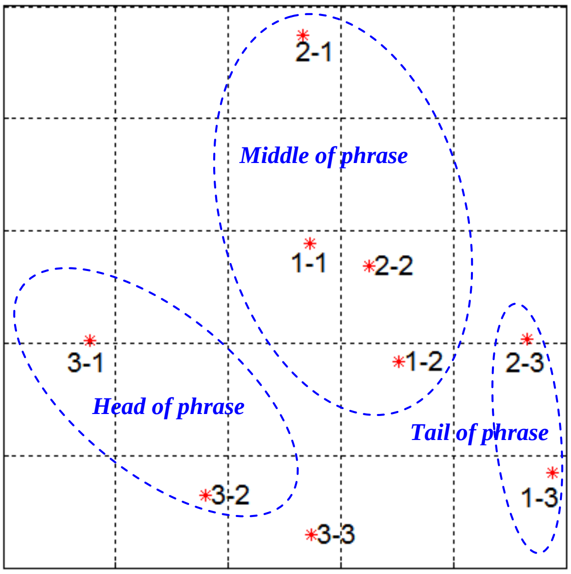


Figure 3. *MDS graph of the KLD matrix for attribute PinP.*

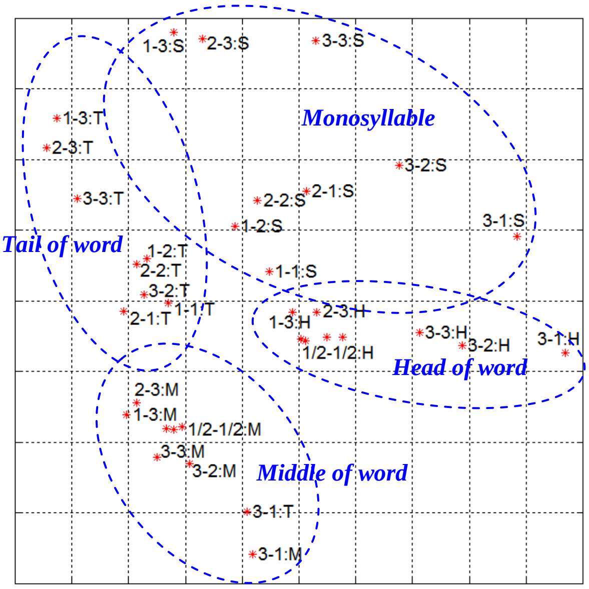


Figure 4. *MDS graph of the KLD matrix for attribute PinP*PinW.*

4.4. Evaluation for prosodic contexts

In this section, we examined the characteristics of KLD with respect to prosodic attributes, such as position of word in phrase (PinP), and position of syllable in word (PinW). Figure 3 displays the MDS plane of the KLD matrix for attribute PinP. It is observed that PinS elements are roughly grouped into 4 parts,

1. Head of phrase (PinS 3-1, 3-2).
2. Middle of phrase (PinS 1-1, 1-2, 2-1, 2-2).
3. Tail of phrase (PinS 1-3, 2-3).
4. PinS 3-3.

We intentionally separate PinS 3-3 from others, because it behaves in between head of phrase and tail of phrase, as is confirmed in Figure 4.

Further, by combining attributes PinS and PinW into a compound attribute, we could investigate the interaction between these two attributes. Figure 4 displays the MDS plane of the KLD matrix for attribute PinP*PinW. Symbols are expressed in the form of [PinS]:[PinW]. We observed that attribute PinW gains more priority in grouping elements than attribute PinP. Inside each PinW group, the group structure of PinS elements is generally maintained.

4.5. Evaluation for numeric contexts

In this section, we examined the characteristics of KLD with respect to numeric attributes, such as the number of phones in syllable (Phns), and the number of syllables in word (Syls). Frankly, if elements of a numeric attribute are of a limited set, there is nothing special in calculating KLD between these elements. What we emphasize here is that, though there exists an apparent metric for numeric attribute, KLD achieves a more reasonable one which agrees with their difference in acoustic characteristics. Here we projected elements on a one-dimensional space, Figure 5 for attribute Phns, and Figure 6 for attribute Syls. It shows that the elements keep the same order in the line as their values suggest, however they are not placed at as equal intervals. As values increase, their deltas, on the whole, gradually decrease. This conforms to our knowledge on these attributes. As for attribute Phns, multiple-phone syllables typically sound different from monophone syllable, and the more phones in a syllable, the less increments of the effect they take in acoustic characteristics. The thing works the same for attribute Syls.

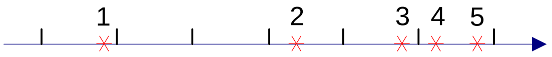


Figure 5. *MDS graph of the KLD matrix for attribute Phns.*

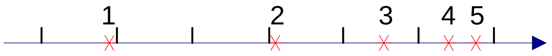


Figure 6. *MDS graph of the KLD matrix for attribute Syls.*

4.6. Subjective evaluation

In this section we evaluate the applicability of KLD as target cost in a task of speech synthesis. In our previous work on English TTS [15], the target cost consists of differences in phonetic and prosodic contexts, and the concatenation cost takes binary values: 0 when two segments to be concatenated are succeeding segments in the recorded speech, and 1 otherwise. Values in the cost function were perceptually tuned by language experts.

In the experiment, we substituted the manually-tuned target sub-costs with the KLD-based ones. Weights for each sub-cost were not studied in the paper and kept the same as original.

We did a preference test to compare the performance of KLD-based target cost with the original manually-tuned one. 30 sentences were synthesized as test stimuli based on a unit database of 5000 utterances. 8 subjects participated in the test and they were forced to choose one from each pair which sounds more natural.

The result for the preference test is given in Table 1. It shows that the synthetic speech obtained with the proposed KLD-based cost sounds slightly better than that with the manually-tuned costs.

Table 1: *Preference ratio for unit selection systems using KLD-based target cost and manually-tuned one.*

	Manually-tuned	KLD-based
Pref. ratio	45.3%	54.7%

5. CONCLUSION

In this paper, we employed KLD between context-dependent HMMs as the target cost in unit selection TTS systems. KLD between MSD-HMMs was presented. Also, we demonstrated that the KLD measure offers a statistically meaningful way to study the acoustic characteristics of speech-related attributes. With the help of MDS, we can visualize the underlining relations of elements from their KLD matrix. Perceptual experiments showed that the TTS system with the KLD-based target cost sounds slightly better than one with the manually-tuned.

Future works include examining the KLD measure on other speech-related attributes, such as part of speech. At this point, we lack the ability to optimize the weights for sub-costs. They may influence the voice quality more than sub-costs. We will investigate how to jointly optimize target sub-costs, weights of target cost, and even the concatenation cost.

6. References

[1] A. Hunt and A. Black, “Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database”, In *Proc. ICASSP 1996*, Atlanta, Georgia, 1996.

[2] A. Black and N. Campbell, “Optimising Selection of Units from Speech Databases for Concatenative Synthesis,” in *Proc. Eurospeech 1995*, Madrid, Spain, 1995.

[3] T. Toda, H. Kawai, and M. Tsuzaki, “Optimizing Sub-cost Functions for Segment Selection Based on Perceptual Evaluations in Concatenative Speech Synthesis”, in *Proc. ICASSP 2004*, Montreal, 2004.

[4] Y. Stylianou and A. K. Syrdal, “Perceptual and Objective Detection of Discontinuities in Concatenative Speech Synthesis”, In *Proc. ICASSP 2001*, Salt Lake City, 2001.

[5] H. Peng, Y. Zhao and M. Chu, “Perpetually Optimizing the Cost Function for Unit Selection in a TTS System with One Single Run of MOS Evaluation,” in *Proc. ICSLP 2002*, Denver, 2002.

[6] P. Taylor, “The Target Cost Formulation in Unit Selection Speech Synthesis”, in *Proc. of Interspeech 2006*, Pittsburgh, 2006.

[7] T. M. Cover and J. A. Thomas, Elements of Information Theory, Wiley Interscience, New York, NY, 1991.

[8] P. Liu, F. K. Soong and J.-L. Zhou, “Divergence-Based Similarity Measure for Spoken Document Retrieval”, In *Proc. ICASSP 2007*, Hawaii, 2007.

[9] J. Goldberger, “An Efficient Image Similarity Measure Based on Approximations of KL-Divergence between Two Gaussian Mixtures”, in *Proc. ICCV 2003*, Nice, France, 2003.

[10] K. Tokuda, T. Masuko, N. Miyazaki and T. Kobayashi, “Multi-Space Probability Distribution HMM”, IEICE Trans. Information and Systems, vol.E85-D, no.3, pp.455-464, Mar. 2002

[11] K. Tokuda, H. Zen, and A. Black, “An HMM-based Approach to Multilingual Speech Synthesis”, in *Text to Speech Synthesis: New Paradigms and Advances*, pp. 135-153. Prentice Hall, 2004

[12] J. Odell, D. Ollason, P. Woodland, S. Young and J. Jansen, The HTK Book for HTK V3.0, Cambridge University Press, Cambridge, 2001.

[13] M. N. Do, “Fast Approximation of Kullback-Leibler Distance for Dependence Trees and hidden Markov Models”, in *IEEE Signal Processing Letters*, Apr. 2003.

[14] F.W. Young, R.M. Hamer, Theory and Applications of Multidimensional Scaling, Eribaum Associates, Hillsdale, NJ, 1994.

[15] M. Chu, H. Peng, Y. Zhao, Z. Niu and E. Chang, “Microsoft Mulan - a Bilingual TTS system,” in *Proc. ICASSP 2003*, Hong Kong, 2003.

[16] J. Yang, Z. Zhao, Y. Jiang, G. Hu, and X. Wo, “Multi-Tier Non-Uniform Unit Selection for Corpus-Based Speech Synthesis”, in *Proc. Blizzard Challenge 2006*, Pittsburgh, 2006.

[17] American English Phoneme Representation, in *Microsoft Speech SDK Version 5.1*, http://msdn.microsoft.com

[18] M. Beckman and J. Hirschberg. The ToBI Annotation Conventions, Ohio State University, Columbus, 1994

Lagrangian Relaxation for Optimal Corpus Design

Jonathan Chevelu, Nelly Barbot, Olivier Boeffard, Arnaud Delhay

IRISA / University of Rennes 1 - ENSSAT
6 rue de Kerampont, B.P. 80518, F-22305 Lannion Cedex
France

jonathan.chevelu@eleves.bretagne.ens-cachan.fr,
{nelly.barbot,olivier.boeffard,arnaud.delhay}@irisa.fr

Abstract

This article is interested in the problem of the linguistic content of a speech corpus. Depending on the target task (speech recognition, speech synthesis, etc) we try to control the phonological and linguistic content of the corpus by collecting an optimal set of sentences which make it possible to cover a preset description of phonological attributes (prosodic tags, allophones, syllables, etc) under the constraint of a minimal overall duration. This goal is classically achieved by greedy algorithms which however do not guarantee the optimality of the desired cover. We propose to call upon the principle of lagrangian relaxation where a set covering problem is solved by iterating between a primal and a dual spaces. We propose to evaluate our proposed methodology against a standard greedy algorithm in order to estimate an optimal phone and diphone covering in French. Our results show that our algorithm based on a lagrangian relaxation principle gives a 10% better solution than a standard greedy algorithm and especially enables to locate the absolute quality of the proposed solution by giving a lower bound to the set covering problem. According to our experiments, our best solution is only 0.8% far from the lower bound of the phone and diphone covering problem.

1. Introduction

Within the field of automatic speech processing, many technologies (word recognition, speech synthesis, speaker recognition, etc.) rely on machine learning mechanisms. Such a methodological framework tries to estimate the parameters of given models using speech corpora recorded from different speakers. The quality of all these models strongly depends on the contents of these learning corpora. So as to cover the maximum of events and to guarantee a powerful modeling, two strategies are possible for the definition of a training corpus.

First of all, it is always possible to collect, at random, more and more acoustic materials. This is the simplest solution but which can quickly become the most expensive insofar as the collected sample undergoes the natural distribution of the required events. In linguistics, this distribution is of exponential nature, which means that very few events take place very frequently compared with a considerable mass of very rare events. Thus, a speech synthesis system which manages several acoustic alternatives of a rare phonological unit will have to harvest a very important amount of data to hope to collect these units.

An alternative to an undifferentiated collection consists in explicitly controlling the content of the learning corpus according to the concerned system. This idea is not recent since phone covering criteria were always applied for the definition of a speech corpus. However, if it is relatively easy to balance the phonetic diversity in a linguistic corpus, it becomes more difficult to control the presence of events longer than the phoneme given the heavy-tailed distribution of these events. For example, a speech synthesis system needs a corpus containing a large variety of 2, 3 or n-phones acoustic units. A speech recognition system using an allophonic modeling will may find it very beneficial to be based on a learning corpus which ensures a good representation of the allophonic alternatives in the language.

We think that it is conceivable to define an optimal covering corpus extracted automatically from huge text corpora and annotated by linguistic attributes (syntactic, grammatical and phonological). The process automatically defines an optimal subset of sentences, considering the overall speech duration, that enables the best *covering* of the different linguistic features necessary to the aimed task. The problem is connected to a *set-covering problem* (SCP) that is a NP-hard problem [1]. It is thus necessary to use sub-optimal or heuristic algorithms.

Many heuristics have already been published. In all cases, the methodology is based on a greedy algorithm. Thus, [2] applies a greedy algorithm to build a database for a speech recognition task thanks to hierarchically organized covering attributes. In [3], the aim is to build a corpus whose diphoneme/triphoneme distribution approximates a uniform distribution. The greedy strategy is driven by a sentence cost function based on the Kullback-Liebler divergence, but does not assure a complete unit coverage. From an algorithmic point of view, [4] proposes a pair-exchange mecanism. In [5], the first *reverse* greedy algorithm is introduced as a spitting algorithm, that deletes uninteresting sentences, and followed by a greedy pair exchange. In [6], several cost strategies and greedy algorithm variants are studied and applied to the construction of a speech synthesis corpus. This methodology has been recently implemented to build the Neologos corpus [7].

In this paper, as an alternative to a greedy algorithm, we propose a solution to the set covering problem based on the lagrangian relaxation. In spite of its simplicity, a greedy algorithm suffers from the sub-optimal results that it can produce, [8]. [9] shows that for one iteration of the computation, the next sentence that will be retained in the optimal solution is chosen in a large set of sentences of identical minimal cost, and can lead to an instable solution.

Solving a set covering problem by lagrangian relaxation may find an exact solution for problems of reasonable scale, ie. a few thousands lines by a few hundreds columns. The complexity order of covering problems that we are interested in speech processing is about millions of lines by thousands of columns (covering attributes). It is then necessary to consider solving algorithms that use heuristics in order to efficiently approach the optimal solution. This paper is essentially based on Caprara's work [10] which deals with crew scheduling in an italian railway company.

The paper is organized as follows. Section 2 first introduces notations and the principles of the set covering problem and next we present our lagrangian relaxation algorithm. Section 3 describes the experimental methodology that we applied to compare a greedy solution and methodology. Finally, in section 4, the results are presented and discussed.

2. Optimal covering using lagrangian relaxation

2.1. Notations and principles

Before presenting the algorithm designed for solving very large scale SCP instances, we introduce in this section some notations and properties relative to the SCP and lagrangian relaxation.

Let us consider a corpus $\mathcal{A}$ of n sentences composed of m distinct attributes $u_1, \ldots, u_m$ - phonological units, acoustic unit classes, prosodic attributes, etc. $\mathcal{A}$ can be represented by a matrix $A = (a_{ij})$, where a_{ij} is the instance number of u_i in the sentence s_j. We denote the unit set $\mathcal{U} = \{u_1, \ldots, u_m\}$ and define $M = \{1, \ldots, m\}$ and $N = \{1, \ldots, n\}$. With every sentence s_j, a cost c_j is combined.

A cover of $\mathcal{U}$ is a subset of $\mathcal{A}$ which contains, for every u_i, a minimal number b_i of instances. It is described by a column vector $X = (x_j)_{j \in N}$, where $x_j = 1$ if the sentence s_j belongs to the cover and 0 otherwise. In other words, a cover is a solution $X \in \{0,1\}^n$ of the following system :

$$\forall i \in M, \sum_{j \in N} a_{ij} x_j \geq b_i . \qquad (1)$$

Since the integer entries of B and A are not only 0 or 1 like in [10, 11, 12], but can be greater than 1, we can term our optimization problem as a *multi-represented* set covering problem. If it is quite easy to determine such a cover, we want a cover with the lowest possible cost. The cost of a cover corresponds to the sum of the costs of all its elements. This SCP can be written as:

$$X^* = \arg \min_{\substack{X \in \{0,1\}^n \\ AX \geq B}} CX \qquad (2)$$

where

$$\begin{aligned} C &= (c_1, \ldots, c_n) \\ B &= (b_1, \ldots, b_m)^T . \end{aligned}$$

We briefly recall the main properties of the lagrangian relaxation on which the algorithm we propose is based on - see, e.g. [13] for an introduction). Let a column vector $\Lambda \in \mathbb{R}^m_+$, we introduce the lagrangian subproblem associated with (2):

$$L(\Lambda) = \min_{X \in \{0,1\}^n} \Lambda^T B + C(\Lambda) X \qquad (3)$$

where the j-th coordinate $c_j(\Lambda)$ of

$$C(\Lambda) = C - \Lambda^T A$$

is called the lagrangian cost, or reduced cost, of s_j. The coordinates of $\Lambda = (\lambda_i)_{i \in M}$ are called lagrangian multipliers and can be interpreted as a weighting of the constraints (1).

The lagrangian function $L(\Lambda)$ satisfies the fundamental property: for every $\Lambda \in \mathbb{R}^m_+$ and every cover X, we have

$$L(\Lambda) \leq CX ,$$

which provides a lower bound of the minimal cover cost. Let us notice that this lower bound is not necessary reached. Its calculus is simple, a solution $X(\Lambda)$ of this optimisation problem in (3) is $x_j(\Lambda) = 1$ if $c_j(\Lambda) < 0$, $x_j(\Lambda) = 0$ if $c_j(\Lambda) > 0$ and $x_j(\Lambda) \in \{0,1\}$ if $c_j(\Lambda) = 0$. Moreover, the lagrangian function gives us information of the usefulness of each sentence within the optimal cover. Indeed, for a given Λ and an upper bound UB of the optimal cover cost, we can compute a gap $g = \text{UB} - L(\Lambda)$ which measures the quality of the relaxation. If $c_j(\Lambda)$ is strictly greater than g, we can check that any feasible solution of SCP containing s_j has a cost value strictly greater than UB. Hence, the variable x_j can be fixed at zero. The same reasoning shows that one can fix x_j to 1 whenever $c_j(\Lambda) < -g$. Therefore, an optimal cover is made up of sentences with a low lagrangian cost [10, 11].

The lagrangian dual problem of (2) consists in determining a lagrangian multiplier vector $\Lambda^* \in \mathbb{R}^m_+$ which maximizes the lower bound $L(\Lambda)$. This real variable function being concave and piecewise affine, a well-known approach for finding a near-optimal multiplier vector is the subgradient algorithm which uses the following subgradient vector:

$$S(\Lambda) = B - AX(\Lambda) . \qquad (4)$$

A simple iterative procedure generates a sequence (Λ^k) based on the updating formula

$$\Lambda^{k+1} = \max \left\{ \Lambda^k + \left(\mu \frac{\text{UB} - L(\Lambda^k)}{||S(\Lambda^k)||^2} \right) S(\Lambda^k), 0 \right\}$$

where Λ^0 is defined arbitrarily, and $\mu > 0$ is a adjustable step size parameter.

2.2. Algorithm

In this paragraph, we describe the algorithm used to produce the optimal linguistic corpus. We note hereafter the *LamSCP* (lagrangian based algorithm for multi-represented SCP) algorithm. This set covering algorithm is inspired by paper [12] and benefits from the main advantages of the lagrangian relaxation described previously in order to obtain the best possible solution as quickly as possible. Our main contribution is the generalization of this approach to take into account the multi-represented problem. This improvement lies on formulas (1) - (4) and the introduction of vector B. The main steps of the algorithm are presented in figure 1. For more details, please refer to [12] and the associated references.

The algorithm is structured into three main phases as mentioned on figure 1. In order to fulfill the objective of precision, we try to optimize a Λ^* vector through the *subgradient* phase. In the *heuristic* phase, the neighborhood of Λ^* is explored a great number of times. A sequence of lagrangian multipliers is generated according to the formula

$$\Lambda^{k+1} = \max \left\{ \Lambda^k + S(\Lambda^k), 0 \right\} ,$$

with $\Lambda^0 = \Lambda^*$, so as to allow for a change in a larger number of components of Λ^{k+1}. A procedure of greedy type is associated to each neighboring vector, in order to obtain a covering through the use of the lagrangian costs. From the best obtained solution, we identify "promising" sentences during the *column fixing* phase. The sub-problem of this residual covering is then processed similarly. The iteration of the 3-phase procedure is stopped when the residual sub-problem is empty, or when the associated lagrangian function is too costly. More precisely, since the lagrangian function indicates a minimal cost for covering the sub-problem, its addition to the costs of the sentences already retained gives a minoration of the total cost of the solution under construction, which should not rise beyond the UB cost of the best known solution in order to be potentially more advantageous.

To reduce the computing complexity, the most frequently used heuristic consists in downsizing the problem by considering mainly the sentences with the lowest lagrangian costs. The procedure known as *pricing*, called during the subgradient phase, consists in getting the 3 phases to work on a subset containing sentences with a low lagrangian cost. We complete this subset with some other sentences in order to make sure that the size of the sub-corpus is sufficient with respect to the number of units to cover. The reduction of the problem in the procedure known as *greedy* consists in selecting the sentence within a limited subset of sentences of lowest lagrangian cost. These costs are then updated. If the maximum in this subset is bigger than the minimal lagrangian cost of the sentences that were initially excluded, the algorithm also updates the working subset. Finally, the phase known as *column fixing* and the procedure known as *refining* consist in really reducing the size of the problem by fixing a set of columns and readapting the matrix as well as the constraints.

The sentences selected during the *column fixing* phase remain selected for the whole *3-phase* procedure. They are chosen among the sentences covering rare units, or with a very low lagrangian cost. More precisely, this phase considers the set of sentences with a lagrangian cost below a threshold τ, and it fixes those that cover lesser frequent units in that set. A greedy procedure is applied on the residual sub-problem, which results in a number of fixed sentences n_f, with a lowest lagrangian cost.

Finally, every time the *refining* procedure is called, the set of sentences is rebuilt. That step selects, up to a certain percentage of covering, the sentences that contribute the least to the gap g.

One should note that in order to adapt the algorithm to the multi-represented case, we threshold matrix A using the constraints B of the problem. Indeed, the covering potential of a sentence is only the minimum between what it really covers, and the minimum number of times that a unit should be covered in the solution.

3. Experimental methodology

3.1. Presentation of the corpora

The corpora on which we have worked have been built from the original text of the journal "Le Monde" during year 1997. The text database initially counts 172,168 annotated sentences structured in sequences of phonemes. To encode one instance of the covering problem, we have used a sparse matrix structure where a line represents a sentence chosen in the corpus, and a column represents a unit to cover. Each non-void cell of the matrix corresponds to a non-zero value which equals the number of occurrences of the unit u_i present in the sentence s_j.

Given the sentences of the original corpus, we have built two different initial corpora: in the first we have taken the whole set of the text sentences as is, in the second one we have cut the original sentences into words to get shorter but more numerous sequences of phonemes. These two corpora are respectively called *le-monde-sentence* and *le-monde-word*.

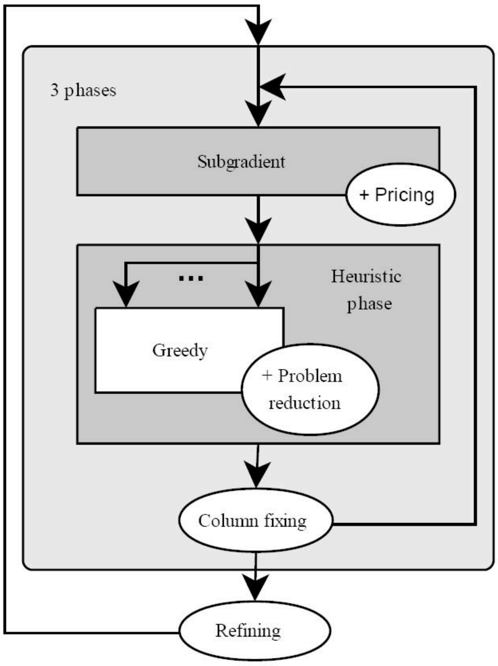


Figure 1: *The LamSCP structure. The rectangular boxes represent the stages which aim to improve the quality of the solution, and the ellipses correspond to the stages which are intended to reduce the problem size.*

With these two corpora we have studied a cover in terms of phonemes and diphonemes. The construction of the covering unit set has been done by examining each sequence and by collecting the phones and diphones encountered.

We present in Tab.1 the statistics concerning the corpora that we have built. One should note that the two databases cover the 35 phonemes of the French language. The corpus *le-monde-word* does not contain any unit composed of the silence symbol.

	le-monde-sentence	*le-monde-word*
Number of sentences	172,168	3,943,099
Number of units (phonemes and diphonemes)	1,207	1,019
Average length in phones (Standard deviation)	97 (60)	5 (2.5)
Matrix density	8.46%	0.87%

Table 1: *Statistics of the studied corpora.*

3.2. Experiment 1, mono-represented cover on corpus *le-monde-sentence*

The first experiment consists in a reduction of the corpus *le-monde-sentence* using two algorithms, a classical greedy algorithm against *LamSCP* in order to compare some results. As a constraint, we want to cover each unit (phonemes and diphonemes) at least once. The main purpose is to minimize the corpus length in term of the recording size. This is the reason why we define the sentence cost as its number of phones.

In this study, we have used a greedy agglomeration algorithm followed by a greedy spitting algorithm. Previous works have shown that this kind of combined algorithm provides most of the time good results with a score function defined as the covering potential of a sentence divided by its number of phones [6, 9].

We have used the *LamSCP* algorithm as described section 2.2. However, because of the exponential distribution of units [6, 9], we do not have to fix too many sentences in the *column fixing* procedure. This is the reason why we use a threshold $\tau = 1$ (see section 2.2) instead of -0.001 proposed initially in [12]. Thanks to this choice, we hope that the algorithm will find a better solution but an other side effect concerns an extra time consuming.

3.3. Experiment 2, multi-represented cover on corpus *le-monde-sentence*

As a second step, we propose to deal with the multi-represented covering case. Indeed, one of our main contribution is the generalization of a lagrangian relaxation algorithm in order to add constraint when more than one representative is needed. As an example, we could impose at least ten representatives per allophonic classes to satisfy a HMM learning requirement in speech recognition.

Once again, we compare the *LamSCP* and the greedy algorithm on the corpus *le-monde-sentence*. But now, we ask for five representatives per unit when it is possible (13 diphonemes are represented less than five times in the original corpus). Except this point, this experiment is realized in the same methodological framework as for the first experiment.

3.4. Experiment 3, mono-represented cover on corpus *le-monde-word*

Finally, it seems interesting to check if variations between the greedy and the *LamSCP* algorithms are related to the difficulty of the problem, i.e. the number of sentence available compared to the constraints and the number of units. That is why we try both methods, greedy and lagrangian relaxation, on the corpus *le-monde-word* which provides less constraints in spite of the great number of columns (around three millions)

We have made this experiment in the same conditions as the first one (see section 3.2). At least one phone and diphone in the solution is wanted.

4. Results and discussion

4.1. Experiment 1

On the one hand, the first experiment shows that the greedy algorithm, applied on *le-monde-sentence*, builds a drastically reduced corpus in size since it is composed of only 8,641 phones instead of the 16,496,441 ones in the original base. The use of *LamSCP* increases again the covering quality by more than 10%, with a solution costing 7,776 phones.

On the other hand, we can notice that *LamSCP* chooses longer sentences than the greedy algorithm, since the average sentence length are respectively 30.0 and 26.9 phones.

Finally, the best lower bound determined by *LamSCP* is equal to 7717 phones. It is close to the obtained solutions since the ideal improvement of the greedy result should be of 10.7%. Thus, relatively to this upper bound given by the greedy approach, our algorithm has done 93.6% of the maximum optimization potential.

Results are detailed in table 2.

4.2. Experiment 2

The second experiment concerns the multi-represented case. The greedy algorithm provides a covering composed of 1,278 sentences for 48,020 phones. So we have an average sentence length at 37.6 phones.

Our algorithm, for the same problem, find out a corpus with only 1,039 sentences for 45,401 phones, which means an average length of 43.7 phones per sentence.

We obtain an improvement of the greedy algorithm of 5.5%. Taking into account the lower bound provided by the *LamSCP* which shows that the cost cannot be lower than 45,109 phones, *i.e.* a maximum improvement of 6.1%, our algorithm has realized 90.0% of the ideal improvement.

Details of the results are provided in table 2.

4.3. Experiment 3

In the last experiment, we have tried to reduce a corpus where each sentence is in fact a word. It shows that the greedy algorithm can find a cover of phonemes/diphonemes with only 2,459 phones.

The lagrangian relaxation has found a solution with 2,163 phones, *i.e.* an improvement of 13.7%. Moreover the lower bound for this problem is 2,028 phones, so the theoretical maximum possible improvement is only 17.5% better than the greedy algorithm. Thus the *LamSCP* has made 68.7% of the maximum improvement.

We can notice that the covering cost in phones is divided by three and half with respect to the first experiment (see section 4.1). It could be interesting to study the relationship between the size of the optimal covering and the average lenght of the sentences. Indeed with a broader cut than at a word level, like syntagms for example, we could probably improve the solution compared to the one based on the corpus with full sentences and keep pronounceable sentences which carry some meaningful events like prosodic realizations.

Details of the results are provided table 2.

4.4. Discussion

The lower bound of the SCP given by the lagrangian relaxation algorithm permits to assess the real performance of the greedy algorithm, which provides a solution close to the best solution (a little more than 10%). For what concerns the *LamSCP*, its solutions are closer to the lower bound, which may be probably not reachable, and then better than the greedy results with a covering cost reduction around 10%.

We can see that it is important to have a good approximation the ability to qualify the results provided by heuristics. This is the main reason why the lagrangian relaxation is interesting for this problem.

Moreover, the results show that sentences chosen by *LamSCP* are around 15% longer than ones picked-up by the greedy

Experiment 1				
"le-monde-sentence" mono-represented	Original	Greedy	LamSCP	Lower bound
Corpus size (phones)	16,496,441	8,641	7,776	7,717
Sentence number	172,168	334	260	N/A
Sentences average length (phones)	96.8	25.9	30.0	NA
Improvement compared to greedy (phones)	NA	0%	10.0%	10.7%

Experiment 2				
"le-monde-sentence" multi-represented(5)	Original	Greedy	LamSCP	Lower bound
Corpus size (phones)	16,496,441	48,020	45,401	45,109
Sentence number	172,168	1,278	1,039	NA
Sentences average length (phones)	96.8	37.6	43.7	NA
Improvement compared to greedy (phones)	NA	0%	5.5 %	6.1%

Experiment 3				
"le-monde-word" mono-represented	Original	Greedy	LamSCP	Lower bound
Corpus size (phones)	16,496,441	2,459	2,163	2,028
Improvement compared to greedy (phones)	NA	0%	13.7%	17.5%

Table 2: *Each table is associated with an experiment. The first one is a mono-represented covering of the corpus le-monde-sentence. The second one is a multi-represented covering of the same corpus with a minimum constraint of five units. The third one is a mono-represented covering of the corpus le-monde-word. The column "Original" shows the main corpora features. The columns "Greedy" and "LamSCP" provide similar information about the covering. The column "Lower bound" indicates the best lower bound found by LamSCP.*

algorithm. That would tend to prove that lagrangian based methods made less local choices compare to greedy methods (taking at each iteration the shortest sentence).

Furthermore, it seems that the greedy algorithm gives good results on difficult problems. Indeed, the less the number of sentences in the problem is, the less the distance between solutions provides by greedy, *LamSCP* and lower bound are. We think that the reason is that for a large instance problem a lot of greedy scores are the same and the algorithm chooses one randomly. But our algorithm, thanks to the lagrangian coefficient, can easily make a difference between those sentences. This can be an explanation of this experimental behaviour.

Finally, for what concerns the computational aspect of *LamSCP*, the programs have been written in C language, expriments were carried out on a Compaq Apha server (ES-45) with a 12 Go user memory. The computation time of *LamSCP* for experiment 1 is approximatively 10 hours against only 1 hour for the greedy algorithm. We are now working on a deep profiling of our proposed solution in order to obtain a global execution time quite similar to the greedy approach.

5. Conclusion

In this paper we proposed an original algorithm as a solution to a set covering problem applied to the automatic creation of linguistic corpora in speech processing. Experiments carried out in French to cover phonemes and diphonemes showed that a solution based on lagrangian relaxation principles is more efficient than a standard greedy algorithm. Starting from a matrix made of 1,207 attributes (phonemes and diphonemes of French) by 172,168 sentences, our proposed *LamSCP* algorithm reaches an optimal solution in 260 sentences (7,776 phones) against 334 (8,641 phones) for a greedy algorithm. Beyond the effectiveness of this cover, the lagrangian relaxation approach gives us crucial information about the quality of the solution. Indeed we know that a lower bound to our covering problem is located at 7,717 phones. Thus, we can conclude that our solution is no more than 0,8% of the true optimal solution.

For the future, we will try to qualify the difficulty of a problem in connection with the optimal estimated solution. One can be interested in finding a relation between some characteristics of the covering matrix and, for example, the difference between the lower bound and the cost of the optimal solution.

6. References

[1] M. Garey and D. Johnson, *Computers and intractability: a guide to the theory of NP-completeness.* Freeman, 1979.

[2] J.-L. Gauvain, L. Lamel, and M. Eskenazi, “Design considerations and text selection for bref, a large french readspeech corpus,” in *Proceedings of the 1st International Conference of Spoken Language Processing (ICSLP)*, 1990, pp. 1097–1100.

[3] A. Krul, G. Damnati, F. Yvon, and T. Moudenc, “Corpus design based on the kullback-leibler divergence for text-to-speech synthesis application,” in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Pittsburg, USA, 2006, pp. 2030–2033.

[4] H. Kawai, S. Yamamoto, N. Higuchi, and T. Shimizu, “A design method of speech corpus for text-to-speech synthe-

sis taking account of prosody," in *Proceedings of the 6th InternationalConference on Spoken Language Processing (ICSLP)*, vol. 3, Beijing, China, 2000, pp. 420–425.

[5] M. Rojc and Z. Kaycic, "Design of an optimal slovenian speech corpus for use in the concatenative speech synthesis system," in *Proceedings of the 2nd InternationalConference on Language Resources and Evaluatio (LREC)*, vol. 1, 2000, pp. 321–325.

[6] H. François and O. Boëffard, "The greedy algorithm and its application to the construction of a continuous speech database," in *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, vol. 5, Las Palmas, Canary Islands, Spain, 2002, pp. 1420–1426.

[7] S. Krstulovic, F. Bimbot, O. Boëffard, D. Charlet, D. Fohr, and O. Mella, "Optimizing the coverage of a speech database through a selection of representative speaker recordings," *Speech Communication*, vol. 48, no. 10, pp. 1319–1348, 2006.

[8] M. Sviridenko, "Worst-case analysis of the greedy algorithm for a generalization of the maximum p-facility location problem," *Operations Research Letters*, vol. 26, 2000.

[9] H. François, "Synthèse de la parole par concaténation d'unités acoustiques : construction et exploitation d'une base de parole continue," Ph.D. dissertation, Université de Rennes 1, 2002.

[10] A. Caprara, P. Toth, and M. Fischetti, "Algorithms for the set covering problem," *Annals of Operations Research*, vol. 98, no. 1, pp. 1–18, 2000.

[11] S. Ceria, P. Nobili, and A. Sassano, "A lagrangian-based heuristic for large-scale set covering problems," *Mathematical Programming*, vol. 81, pp. 215–228, 1998.

[12] A. Caprara, M. Fischetti, and P. Toth, "A heuristic method for the set covering problem," University of Bologna - OR, Tech. Rep. 8, 1995.

[13] M. L. Fisher, "An applications oriented guide to lagrangian relaxation," *Interfaces*, vol. 15, pp. 10–21, 1985.

Adaptive Database Reduction for Domain Specific Speech Synthesis

Aleksandra Krul [1],[2] *Géraldine Damnati* [1], *François Yvon* [2], *Cédric Boidin*[1], *Thierry Moudenc* [1]

[1] France Télécom R&D Division, TECH/SSTP
2, avenue Pierre Marzin, 22307 Lannion Cedex, France
{aleksandra.krul,geraldine.damnati,cedric.boidin,thierry.moudenc}@orange-ftgroup.com

[2] GET/ENST and CNRS/LTCI
46, rue Barrault, 75624 Paris Cedex 13, France
yvon@enst.fr

Abstract

This paper raises the issue of speech database reduction adapted to a specific domain for Text-To-Speech (TTS) synthesis application. We evaluate several methods: a database pruning technique based on the statistical behaviour of the unit selection algorithm and a novel method based on the Kullback-Leibler divergence. The aim of the former method is to eliminate the least selected units during the synthesis of a domain specific training corpus. The aim of the latter approach is to build a reduced database whose unit distribution approximates a given target distribution. We compare the reduced databases. Finally we evaluate these methods on several objective measures given by the unit selection algorithm.

1. Introduction

Current Text-To-Speech systems are based on concatenative methods [1]. Such systems use a large database of pre-recorded speech from which acoustic units are selected for concatenation. The scalability of the database is an important issue in unit selection based speech synthesis. Indeed, the use of the full database is not always suitable or even possible for some applications. The database has to be reduced so that the speech synthesis system can be integrated into different devices.

Two approaches are commonly used for database reduction. In a ”bottom-up” approach the database is examined in order to remove spurious and redundant units. For instance, in [2, 3] units are clustered according to some similarity measures concerning prosodic and phonetic contexts. Only units that are representative of each cluster are kept in the reduced database. More recently an LSM (Latent Semantic Mapping) method was proposed in [4].

The ”top-down” approach is based on the investigation of the output of the synthesizer. One of the implementations consists in synthesizing a large amount of data and removing units which are not frequently used by the synthesizer. This approach is based on the statistical behaviour of the unit selection algorithm and was originally proposed in [5]. The advantage of such a method is that no knowledge about speech units is needed. It is closely dependent on the unit selection algorithm behaviour.

However, the reduced synthesis systems are often used for specific applications such as menu readers in the mobile phones. The reduced database has to be adapted to the domain specific application.

In this paper we are interested in this particular paradigm. Our goal is to prune the generic database and to adapt it in order to synthesize a domain specific application corpus in different devices that do not support a large amount of data. As the acoustic realization of a specific domain is not known the use of methods such as in [2, 3, 4] is not possible for the reduction adapted to a specific application. We investigate then two approaches: a variant of a reduction method based on the statistical behaviour of the unit selection and a novel reduction method guided by the Kullback-Leibler measure.

The first reduction method that we use is a ”top-down” approach. Instead of synthesizing a generic corpus we propose to use a domain specific corpus that reflects the application for which the reduction has to be performed. We will show that even if the specific corpus is not very large we obtain better objective results than if we collect statistics by synthesizing a much bigger generic corpus.

The second approach that we investigate is based on the Kullback-Leibler divergence and was introduced in [6]. This method was used for designing a textual corpus for the speech synthesis application. The main idea of this method is that the distribution of units in the constructed corpus aims to be close to an *a priori* distribution. In [6] the flexibility of this method is put forward: the algorithm is able to accommodate different distributions which may prove better for domain specific TTS synthesis applications. We use this method to construct a reduced database whose unit distribution is close to the domain specific distribution. The distribution of units in the reduced database can be adapted to any domain. The advantage of this method is that it is independent of the speech synthesis system.

In section 2, we present several approaches for adaptive database reduction. In section 3 we objectively evaluate all of the methods and present experimental results.

2. Presentation of methods

2.1. Database pruning based on the statistical behaviour of the unit selection algorithm

The main idea of this pruning method is to keep the units that are the most often used to synthesize a representative corpus while the least selected units are pruned. Our system uses diphone as elementary unit. Each diphone (about 1200 in French) is present several times (from 1 to thousands) in the acoustic database: each acoustic realization is called a diphone variant or a unit. When synthesizing a message, each variant can or cannot be selected. The number of times it is selected is called number of occurrences. The first step consists in synthesizing an important representative corpus and in counting the number of occurrences of each variant. Then the pruning step is performed independently for each diphone. All diphone variants are sorted according to their number of occurrences. The ones with the highest number of occurrences are kept while the ones with the lowest number of occurrences are pruned. The number of variants to be kept is calculated in order to reach a target coverage or a target reduction rate. This method is referred hereafter as Ps.

2.2. Method based on Kullback-Leibler divergence

The KL divergence [7] is a measure which assesses the similarity between two probability distributions. It is defined as:

$$D(P \parallel Q) = \sum_{i=1}^{t} p_i \log \frac{p_i}{q_i} \tag{1}$$

where P and Q are two discrete probability distributions.

The properties of this measure are the following. The divergence is positive or equal to zero. The two probability distributions are identical if and only if the KL divergence is null.

In the presented method all the speech database sentences are split into phrases also called breath groups. The KL based reduction method takes two steps. First we select phrases whose unit distribution approximates a target distribution from the corpus that was previously recorded for the database. Then we reduce the ordered phrases according to different reduction rates. The phrases are selected incrementally with a greedy algorithm. At a given iteration the unit distribution on the corpus that would be obtained by adding a candidate phrase is evaluated. The phrase for which this distribution results in the lowest KL divergence to the target is picked. The score of each candidate phrase is :

$$D(P \parallel Q) = \sum_{i, n_i \neq 0} \frac{n_i}{N} \big(\log \frac{n_i}{N} - \log q_i \big), \tag{2}$$

where Q denotes the target distribution and P is the constructed distribution. n_i is the number of occurrences of a diphone i in the constructed corpus, and N is the total number of units ($N = \sum_i n_i$).

In [6], we presented in details the behaviour of this algorithm. We also showed how to efficiently update, in an incremental manner, the Kullback-Leibler divergence at each step of the algorithm.

The target distribution is estimated on a training corpus which is representative of a specific domain. The adaptation of the selected corpus to various distributions is easy to implement: what is only required is to obtain Q from a given domain specific corpus and to set it as the target distribution in our algorithm.

We consider the diphone and triphone distributions. However, to ensure the full coverage of elementary units (diphones) we have to include the following constraint. Among the phrases that contain new distinct diphones the algorithm selects the phrase that minimizes the KL divergence to the target diphone (KL_{dip} method) or triphone distribution (KL_{trip} method). This constraint makes us sure that we will have at least one instance of each diphone in the reduced database. However, this method selects only units that are present in the target corpus. To tackle this problem we use an ϵ smoothing unigram technique. A fixed value ϵ is attributed to units that are not present in the estimation corpus. The smoothing formulas are as follows:

$$q_i = \begin{cases} f_i \cdot [1 - \epsilon \cdot C_0] & \text{if } c(d_i) \neq 0 \\ \epsilon & \text{otherwise} \end{cases} \tag{3}$$

where $c(d_i)$ is the count of the diphone i, f_i is the relative frequency of the unit i and C_0 is the number of unseen units in the estimation corpus.

After the selection process speech database phrases are ordered. The first phrases are kept to make the reduced database, the number of phrases to be kept depending on the reduction rates.

2.3. Random method

This approach consists in randomly ordering phrases of the textual corpus. In order to ensure the diphone coverage in the reduced databases the same process was used as for the KL based method reduction. In the first step the random selection is made only among the phrases that contains new distinct diphones. When the full diphone coverage is achieved the selection process becomes completely random. We will refer to this method as $random$.

3. Experimental setup

3.1. Data

For our experiments we used a large database of a French speaker. The database contains about $7,000$ sentences which correspond to $12,500$ phrases and $252K$ units. In order to collect statistics on the use of diphone variants by the system and to estimate the distribution of diphones and triphones in the domain specific corpus we used two domain specific corpora C_{re} and C_{cs}. C_{re} contains $8,866$ sentences. It is collected from the small ads from the real estate domain. C_{cs} contains $7,952$ sentences and it is collected from the small ads related to the computer science domain. C_{re} and C_{cs} are split into a training corpus and a test corpus. The training corpus is used to perform the synthesis and to estimate the diphone and triphone distributions. Table 1 presents some corpora description.

Table 1: Corpora description.

	number of sentences	number of phrases	number of diphone types
Real estate TRAIN SET	6685	25039	1014
Real estate TEST SET	1746	6067	901
Computer science TRAIN SET	6428	15521	1080
Computer science TEST SET	1524	4010	989

3.2. The KL divergence behaviour

In the figure 1 we present the KL divergence measure obtained during the first step of database reduction for the KL divergence based method i.e. the phrase selection. The KL divergence measure decreases quickly at the beginning of the process. The algorithm seeks for phrases whose unit distribution minimizes the KL divergence to the target unit distribution. After reaching the first minimum value the KL divergence increases significantly. This is due to the fact that a constraint on the selection process was added. Indeed, the algorithm picks sentences that

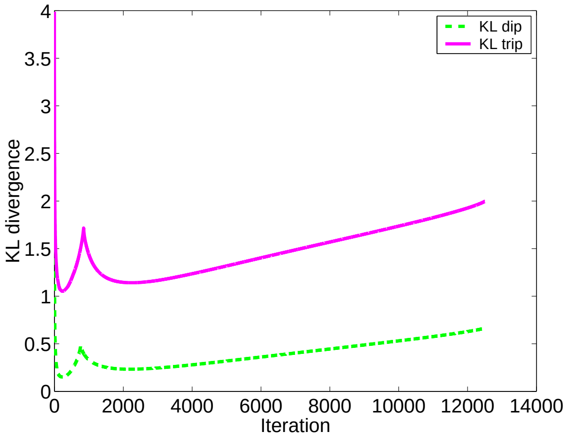


Figure 1: *KL divergence.*

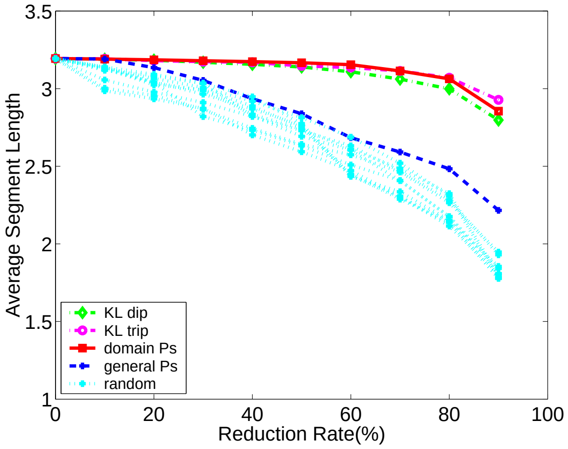


Figure 2: *Average segment length.*

contain new distinct diphones. When the full diphone coverage is reached the KL divergence decreases. After reaching the second minimum value the KL divergence increases. Any new sentence added starting from this minimum increases the KL divergence to the target distribution. This is because the entire phrases are selected. As the algorithm selects entire phrases, the resulting distribution inevitably reflects the characteristics of the original distribution.

3.3. Collecting statistics for the Ps database reduction

Two sets of statistics were collected for the Ps method.

Firstly, a corpus that contains about $359K$ newspaper text files was used. It corresponds to about $97M$ diphone occurrences. The selected variants represent 96% of the database. This method is referred hereafter as $general\ Ps$.

Secondly, we tried to run this reduction method on a smaller domain representative corpus. The C_{re} training corpus was synthesized. It corresponds to $403K$ diphone occurrences. Due to the limited size of the domain and the corpus, only 16% of the variants present in the generic database are selected at least once when synthesizing the corpus. This means that for reduction rates lower than 84% there is necessarily a random part in the algorithm in order to choose among the unused units. Among these unused units it was therefore decided to keep the first variants of the database, i.e. in the order they are stored in the database. It has to be noted that some unused units are useful for the synthesis of the C_{re} test set. Therefore it is important to be able to target reduction rates smaller than 84%, even though there is some randomness in the reduction process. This method is hereafter referred as $domain\ Ps$.

4. Objective evaluation

To evaluate the reduced databases we compare the aforementioned methods. The reduced databases are created by removing 10%, 20% ... 90% of units.

We consider four objective measures which are given by the unit selection algorithm: the average length of selected segments, the average concatenation cost, the average target cost and finally the average cost. The target and the concatenation costs are the classical notions of unit selection technique. The target cost estimates how close a database unit is to the desired unit. The concatenation cost estimates how well two adjacently selected units join together. The overall cost is a sum of the concatenation and the target costs. The selection algorithm minimizes the overall cost in order to find the optimal unit sequence. The average segment length measures the average number of units in the segment, i.e. a string of adjacently selected units. For instance, an average segment length equal to 1.0 means that none of the selected units are adjacent in the database. This measure is reverse proportional to the number of concatenations.

In [8] it has been shown that the average segment length and the average concatenation cost are highly correlated with MOS (Mean Opinion Score) tests. These measures are shown in figure 2 and in figure 3.

4.1. Average segment length

We investigate the average segment length in figure 2. The $domain\ Ps$, KL_{dip} and KL_{trip} obtain significantly longer segments than the $general\ Ps$ and $random$ methods. At first sight, one may think that the fact the KL methods select segments is due to the fact that they keep entire phrases in the reduction process. However, the $random$ methods that also select entire phrases have poor average segment length, even worse than $general\ Ps$ method whose reduced databases are discontiguous. Therefore the adaptation to the specific context of the reduced database seems to be important to select adjacent units; the three adaptive reduction methods are equivalent for this criterion.

4.2. Average concatenation cost

We can look then at the average concatenation cost. For each synthesized sentence the average concatenation cost is the sum of all concatenation costs normalized by the total number of units in the sentence. The average concatenation cost that is shown in figure 3 is the average of the average concatenation costs of each sentence. In these figures as well as in the following ones a cost of 2 means that the cost is twice as high as the initial cost obtained on the whole database (0% reduction).

The lowest average concatenation cost, i.e. the best, is obtained with the $domain\ Ps$ method. Then KL_{trip} is better than the KL_{dip} method which is better than $general\ Ps$ method. The random methods are significantly worse than all other methods. The order is the same as for the average segment

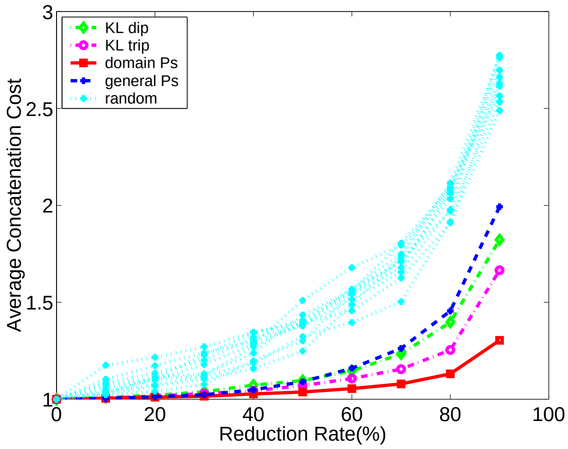


Figure 3: *Average concatenation cost, relative to the* 0% *reduction concatenation cost.*

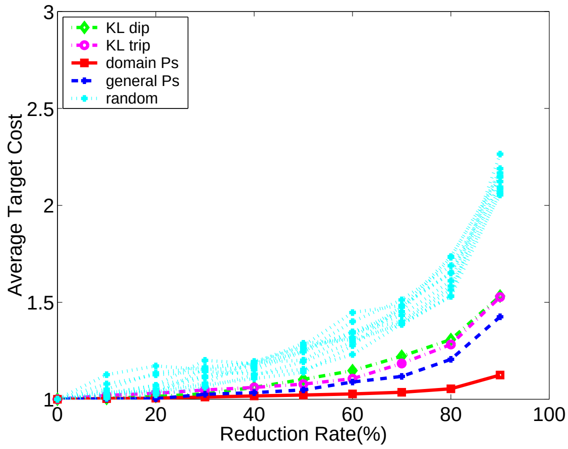


Figure 4: *Average target cost, relative to the* 0% *reduction target cost.*

length but the three methods $domain\ Ps$, KL_{dip} and KL_{trip} obtain distinct scores. The KL based methods obtain higher costs than the $domain\ Ps$ probably because they consider only basic units without taking into account concatenation cost criteria, i.e. acoustic features. We notice that KL_{trip} have better concatenation cost than KL_{dip}, it seems to be correlated to the small difference between their average segment length.

In table 2 we show the percentage of the minimum and the maximum concatenation cost values (respectively cct_{min} and cct_{max}) present in the synthesized C_{re} test set. As can be seen from these numbers, the percentage of cct_{max} is not high. There isn't a big difference between the number of cct_{max} present in the synthesized corpora using $domain\ Ps$, KL_{dip} and KL_{trip}.

Table 2: Concatenation cost comparison.

	$general\ Ps$	$domain\ Ps$	KL_{dip}	KL_{trip}
cct_{min} (%)	53.22	61.75	58.29	60.27
cct_{max} (%)	0.07	0.03	0.03	0.04

4.3. Average target cost

The average target cost is calculated in the same manner as the average concatenation cost. It is shown in figure 4. The best average target cost is obtained for $domain\ Ps$. The second best average target cost is obtained for $general\ Ps$, the KL based methods KL_{dip} and KL_{trip} seem to be equivalent with a higher cost than the two statistically based methods. As we consider only a simple distribution of basic units the KL based methods do not use enough information about the units that are selected. To improve the KL based method it may prove necessary to consider not only the phonetic nature of the units, but also features which characterize the units: length, stress, syntactic, lexical and phonetic context, etc.

Table 3 shows the percentage of the minimum and the maximum target cost values (respectively tgt_{min} and tgt_{max}) present in the synthesized C_{re} test set. As can be seen from this table, the tgt_{max} number is much higher for the KL_{dip} method than for the $general\ Ps$ and KL_{trip} methods. The percentage of tgt_{max} is very low for the for the $domain\ Ps$ method. The percentage of tgt_{min} is higher for the $domain\ Ps$ method than for the others methods. It has to be noted that the $domain\ Ps$ method was explicitly designed to minimize the target and the concatenation costs.

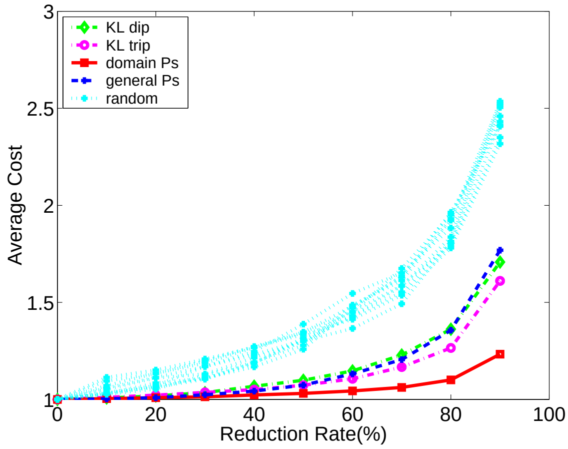


Figure 5: *Average cost, relative to the* 0% *reduction cost.*

Table 3: Target cost comparison.

	$general\ Ps$	$domain\ Ps$	KL_{dip}	KL_{trip}
tgt_{min} (%)	57.78	64.64	56.00	56.36
tgt_{max} (%)	0.72	0.31	1.37	0.73

4.4. Average cost

We finally examine the overall cost in the figure 5 which is the sum of the target and concatenation costs. The best method is logically $domain\ Ps$, the second best is KL_{trip}. KL_{dip} and $general\ Ps$ are close to each other. This shows that the average target cost loss of KL_{trip} compared to $general\ Ps$ is smaller than the average concatenation cost gain.

4.5. Reduced databases comparison

We compare the reduced databases with the $domain\ Ps$ and KL_{trip} methods. The higher reduction rate is the less similar

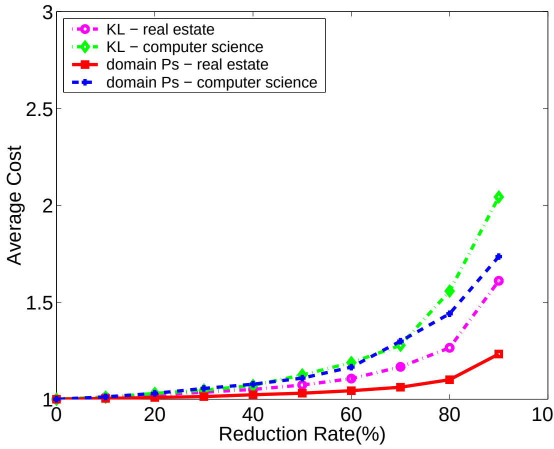


Figure 6: *Average cost tested on C_{re} test set.*

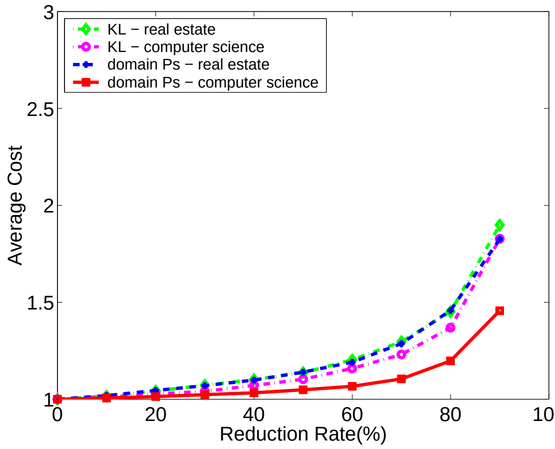


Figure 7: *Average cost tested on C_{cs} test set.*

are databases. At 90% reduction rate there are only 19% of units that are present in the reduced databases obtained for the real estate domain. While synthezising the C_{re} test set, 34% of units are used in common for the two different databases. While using the database obtained with the $domain\ Ps$ method $12,900$ units are selected by the synthesizer. $11,900$ units are selected by the unit selection algorithm using the database obtained with KL_{trip} method. As for the synthesis performed with the full database $20,000$ units are selected to synthezise the test corpus. We also count units that are selected from the reduced databases and from the full database. 78% of units are used in common between the full database and the reduced database obtained with the $domain\ Ps$ method. From these statistics we can conclude that the C_{re} training and test sets are relatively close. Only 37% are selected in common between the full database and the reduced database obtained with the KL_{trip} method.

4.6. Cross tests

We examine the behaviour of the proposed methods by performing cross tests. We use the training sets of C_{re} and C_{cs} corpora to construct the reduced databases with $domain\ Ps$ and KL_{trip} methods. With the reduced databases adapted for the computer science domain we perform tests on the C_{re} and C_{cs} test sets. This is shown on the figure 6. With the reduced databases adapted to the real estate domain we perform tests on the C_{re} and C_{cs} test sets. This is shown on the figure 7.

As we expected the database that is adapted to the specific domain distribution obtains better average costs on the test set from the same specific domain. This is perfectly illustrated by figure 6. This is also the case for the $domain\ Ps$ method on figure 7. However the results obtained with KL_{trip} are not significant. This can be explained by the fact that the size of C_{cs} test set is smaller than C_{re} test set one.

5. Discussion

In this study, we have investigated several reduction methods. The first observation is that the adaptive reduction methods overcome standard reduction $general\ Ps$ and $random$ reduction methods.

Even on the relatively small domain specific corpora the $domain\ Ps$ method seems to give the best results. This may be surprising as only 16% of the units have been used in the training corpus synthesis. Moreover, it has to be noted that some of the units that are used for test corpus synthesis were selected arbitrarily during the reduction process. That can be seen from figure 5 as the cost decreases between 80% and 0% reduction rate while only unused units are added arbitrarily to the reduced database.

The KL based method is almost equivalent to $domain\ Ps$ for the average segment length. These results are promising as we have targeted distribution estimated on the basic types of units, i.e. diphones and triphones. The information about prosodic and linguistic features can be introduced while estimating the target distribution. Indeed, the KL based method gives the possibility to globally control the unit distribution on a variety of features. We have therefore added some features to describe diphones. We take into consideration the position of the syllable which contains the diphone and the prosodic context of the syllable. In figure 8 we present the average cost for the KL based method for diphones described with the aforementioned features ($KL_{dip+features}$ method). We compare it with the $domain\ Ps$ and KL_{trip}. The $KL_{dip+features}$ method is almost equivalent to the KL_{trip} method. For 90% reduction rate the $KL_{dip+features}$ method is better than KL_{trip}. This raises the issue of which are the relevant features to describe the target distribution of a specific domain.

These methods might be combined in order to improve the reduction process, taking benefit from the close link of the $domainPs$ method to the cost function and from the possibility to globally control the unit distribution on a variety of features by the KL based method.

6. Conclusions

In this study, we have presented approaches for adaptive database reduction. We have adapted a classical reduction approach and we have proposed a method based on the Kullback-Leibler divergence. We have objectively evaluated the presented methods. Adaptive database pruning methods are promising reduction methods. It seems to be more suitable to use those techniques when the application for which the reduction has to be made is known.

The advantage of the presented reduction methods is that the reduced database can be adapted to any domain. For the statistically based approach it is simply a matter of collecting

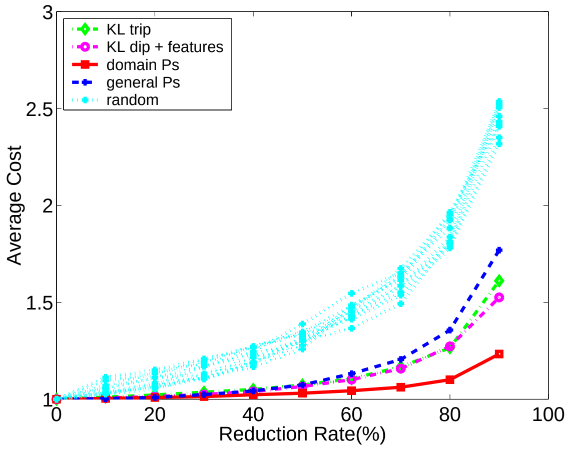


Figure 8: *Average cost tested on C_{re} test set.*

new statistics in the use of the database for a domain specific corpus. For the KL divergence based method it is only required to obtain Q from a given domain specific corpus and to set it as the target distribution in our algorithm.

Our future plans also include exploring this method on several domain specific corpora and examining other types of units, for instance contextual units.

Finally, the speech synthesis quality evaluation has to be performed.

7. References

[1] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *ICASSP-96*, Atlanta, Georgia, USA, May 1996, pp. 373–376.

[2] A. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in 5^{th} *European Conf. on Speech Communication and Technology (Eurospeech)*, Rhodes, Greece, Sept. 1997, pp. 601–604.

[3] S. Kim, Y. Lee, and K. Hirose, "Pruning of Redundant Synthesis Instances Based on Weighted Vector Quantization," in 7^{th} *European Conf. on Speech Communication and Technology (Eurospeech)*, Aalborg, Denmark, Sept. 2001, pp. 2231–2234.

[4] J. Bellegarda, "LSM-Based Unit Pruning for Concatenative Speech Synthesis," in *ICASSP-07*, Honolulu, Hawaii, USA, April 2007.

[5] P. Rutten, M. Aylett, J. Fackrell, and P. Taylor, "A Statistically Motivated Database Pruning Technique for Unit Selection Synthesis," in 7^{th} *Int. Conf. on Spoken Language Processing (ICSLP)*, Denver, Colorado, USA, Sept. 2002, pp. 125–128.

[6] A. Krul, G. Damnati, F. Yvon, and T. Moudenc, "Corpus Design Based on the Kullback-Leibler Divergence for Text-to-Speech Synthesis Application," in 9^{th} *Int. Conf. on Spoken Language Processing (ICSLP)*, Pittsburgh, Pennsylvania, USA, Sept. 2006, pp. 2030–2033.

[7] T.M Cover and J.A Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications, 1991.

[8] M. Chu, Ch. Li, H. Peng, and E. Chang, "Domain Adaptation for TTS Systems," in *ICASSP*, Orlando, USA, May 2002, pp. 453–456.

Statistical analysis of filled pauses' rhythm for disfluent speech synthesis

Jordi Adell[1], *Antonio Bonafonte*[1], *David Escudero*[2]

[1]Dpt. of Signal Theory and Comunications, Universitat Politècnica de Catalunya, Spain.
[2]Dpt. Computer Science, Universidad de Valladolid, Spain.

Abstract

Given that state of the art speech synthesis systems have already reached a high naturalness level, it is time to move to *talking* speech from the actual *read* speech framework. For this purpose it is thus necessary to investigate how disfluencies can be included in speech synthesis and even increase its naturalness. This paper builds on a previously presented work and focuses on finding a local model of filled pauses rhythm. A statistical study of rhythm effects around filled pauses is presented and based on the correlation between rhythm variables, a regression model is proposed to predict filled pauses duration and prepausal lengthening.

1. Introduction

Speech synthesis has already reached high naturalness, mainly due to the use of effective techniques such us unit selection-base systems [1] or other new arising technologies [2] based on the analysis of huge speech corpora. The main application of speech synthesis has been focused by now on reading style speech as it is plausible to assess that reading style is the most generalist style to be extrapolated to any other situation. But nowadays new applications of text-to-speech (TTS) systems like film dubbing, robotics, dialogue systems, speech translation or multilingual broadcasting demand different styles as the users expect the interface to do more than just reading information.

If synthetic voices want to be integrated in future technology, they must *speak* the way people talk instead the way people read. This objective has been already tackled in several manners such as emotional speech synthesis [3], voice quality modelling [4] or even pronunciation variants [5]. In our opinion, style is more important; it is desirable synthetic speech to be more conversational-like rather than reading-like speech. We call this *talking speech*, in contrast to *read speech*.

Talking speech differs significantly from reading speech due to the inclusions of a set of a variety of prosodic resources affecting the rhythm of the utterances. Disfluencies are one of these resources defined as *phenomena that interrupt the flow of speech and do not add propositional content to an utterance* [6]. Disfluencies are very frequent in normal speech [7] and they in fact contain information [8] and help human communication [9, 10]. Then, it is plausible to hypothesise the need to include this prosodic event in order to move towards to talking speech synthesis. In the present work we focus in one kind of disfluency: *filled pauses*.

There already exist published works on disfluent speech synthesis like the one done in [11], where they presented an algorithm for insertion of filled pauses and breathing into a text. Also in [12], where they present a study about prosodic cues of hesitations for speech synthesis.

We have also presented experiences on synthesising disfluencies (i.e. filled pauses and repetitions) in TTS systems in previous works [13], and here we present further work on the same direction focusing on filled pauses' rhythm. In our previous work, we claimed that filled pauses' pitch is lower than its segmental context. However, we were not able to find any simple model to predict the filled pause (FP) duration and a constant value was proposed. Although the synthesis of filled pauses reached higher degree of quality than repetition synthesis in informal tests, we have detected two main drawbacks: *coarticulation* and *rhythm*.

Since our work is based on a unit-selection approach, coarticulation problems come from the lack of FP units in the inventory and from the fact that filled pauses can be strongly coarticulated, some times it is hard even to differentiate, in human speech, filled pauses from strong vowel lengthening. The second drawback was that the sentence rhythm was not affected by the presence of the filled pause at all in the synthetic speech. It was inserted into a fully fluent utterance in terms of rhythm and it sounded unnatural.

In this paper, first of all the database used is described. Then in;3C Section 3, the use of silent pauses to avoid coarticulation is discussed. The study on the rhythm of sentence with filled pauses is presented in Section 4. Afterwards, due to the similar naturalness between filled pauses and silent pauses, the findings of the study will be analysed in the case of silent pauses in Section 5. Finally conclusions are summarised in Section 6.

2. Database and Synthesis

A database has been recorded specifically for unit-selection speech synthesis of conversational speech. Two large databases of about 10h of speech each one where recorded in order to build a couple of high quality voices for our TTS system, a male and a female voice. In addition, some extra sentences where recorded to study the synthesis of disfluent speech synthesis.

Prompts to record these sentences where extracted from real utterances from the European Parliament. They consisted on 65 sentences, which contained filled pauses, repetitions, restarts and breathing. These 65 sentences have been recorded by the male as well as by the female speaker. The prompt given to both speakers contained indications of where to do filled pauses, repetitions and others disfluent events.

In the case of filled pauses, prompts signalled when a filled pause had to be uttered but no acoustic specifications was given to the speaker. Therefore, the database contains a variety of realisations: *ehh*, *ahh*, *mmm*, *emm*. However, in the present paper all filled pauses have been considered equally.

Furthermore, the sentences have been manually segmented at phone level. These sentences have been added to the unit-selection inventory. The present work is focused on filled pauses and the database contains 138 of them.

These units, i.e. *ehh* or *mmm*, have been turned into phone-like units that can be used in the selection and concatenation process which need prosodic values to choose the most appropriate unit. Therefore, as well as for phones, prosodic models are requested and this is the main motivation of this work.

3. Silent Pauses Insertion

In order to avoid coarticulation problems, the insertion of silent pauses at both sides of filled pauses is proposed here. Experimental observations have motivated such proposal. In Figure 1 it can be observed how both silent pauses are present at both sides of the filled pause (*ehh*).

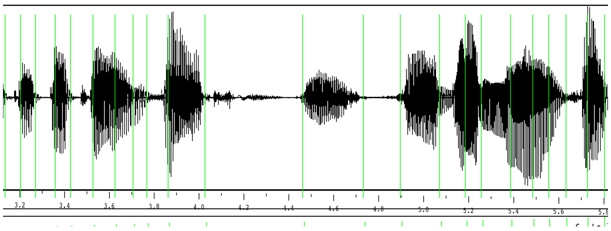


Figure 1: Audio example. It can be observed the pre-pausal syllable lengthening (/fa/) and also the short silences before and after the filled pause (/ehh/).

Table 1 shows how many times in the database silent pauses appear next to a filled pause. It can be observed that if we consider both speakers together 49% of filled pauses contain at least one of those silent pauses. This fact supports the insertion of these silent pauses. On one hand, the female speaker do not include both of them never in the database and only one forth of the filled pauses contain at least one SP. On the other hand, the male speaker uses this silent pauses more often since two thirds of the FP contains at least one SP.

	Both Spk.		Male Spk.		Female Spk.	
Combinations	n	%	n	%	n	%
FP	70	51%	25	32%	46	74%
SP·FP·SP	16	**12%**	16	20%	0	0%
SP·FP	12	**9%**	8	10%	5	8%
FP·SP	39	**28%**	28	36%	11	17%

Table 1: Number (n) and frequency of occurrences of silent pauses together with filled pauses. Silent pauses at both sides (SP·FP·SP), at one side (SP·FP or FP·SP), and no silent pauses (FP) have been taken into account.

Therefore, the use of both silent pauses is not the most frequent structure used by the two speakers we analyse here.

However, it is a possible structure and thus we are allowed to use it in order to avoid coarticulation problems in the insertion of FP. Therefore, the silent pauses' length is a variable that has to be predicted by the prosodic model.

4. Rhythm Study

In this section we will discuss rhythm implications in filled pauses. For this purpose, a set of rhythm-related variables will be defined. Afterwards, some summary statistics are presented in order to identify the general behaviour of such variables. Then, correlation between variables is explored and a regression model is proposed for prosody modelling of filled pauses.

4.1. Feature-set definition

We define *"rhythm"* as the mean syllable length of an utterance. This can be done in Spanish since syllable is the basic segmental unit for timing [14]. Since we are interested to discuss whether the filled pause produces a rhythm change or not, three rhythm variables are define: the total rhythm of the sentence (i.e. mean syllable length across the whole sentence), the rhythm previous to the filled pause, from the beginning of the sentence; and the rhythm after the FP. We will refer to these variables as: *totrh*, *prerh* and *posrh*. The filled pause duration has been excluded and is evaluated separately (i.e. *fpdur*). Through experimental observations we realised that prepausal syllables were larger than the mean syllable length. This phenomena can be observed in Figure 1. In this audio example, the prepausal syllable (*/fa/*) is significantly larger than the rest of syllables. Therefore, this value has also been excluded from rhythm calculus and variable syl_{-1} will represent syllable length of syllable previous to filled pause. Moreover, since silent pauses before and after the FP will be part of the model, two more variables are included in the study, they represent both silent pauses' length: *paupre* and *paupos*. In addition, in order to examine whether only the prepausal length is lenghtened or not, syl_{-2} was added; and given the importance of the syllable nucleus (i.e. the vowel) in the syllable length also its duration has been included: nuc_{-1} and nuc_{-2}. In Summary, the set of features extracted from the database for each filled pause are: *totrh*, *prerh*, *posrh*, syl_{-2}, nuc_{-2}, syl_{-1}, nuc_{-1}, *paupre*, *fpdur* and *paupos*.

4.2. Summary statistics

Table 2 shows mean, standard deviation, lower and upper quartiles for each feature corresponding to male speaker. Table 3 shows same statistics for the female speaker. It can be observed how rhythm distributions are very similar for the total, the previous and the posterior rhythm in the case of the male as well as for the female speaker. Hypothesis tests have shown that at 95% confidence level rhytm means are equal. This supports our claim that filled pauses do not imply a rhythm change in the sentence. Therefore, the prosody of the corresponding fluent sentence can be modelled and rules to predict *fpdur*, syl_{-1}, *paupre* and *paupos* could afterwards be applied. This is specially useful in our case, since the biggest part of the synthesis inventory are built by fluently uttered sentences, while only a small part of it contains disfluencies.

Unit:ms	*Mean*	*Std*	*Lower*	*Upper*
Name		*Deviation*	*Quartile*	*Quartile*
totrh.	167	35	143	182
prerh.	173	63	149	181
posrh.	180	42	154	199
syl$_{-2}$	234	125	151	301
nuc$_{-2}$	114	83	74	112
syl$_{-1}$	394	179	280	494
nuc$_{-1}$	228	118	140	288
paupre	348	282	100	465
paupos	242	270	71	404
fpdur	464	223	294	655

Table 2: Summary statistics for the male speaker and for filled pauses.

It can be observed how there is a significant lengthening of the prepausal syllable. Note that mean value of syl_{-1} is 2.3 times bigger than the mean syllable length of the sentences for the male speaker and 2.45 in the case of the female speaker. Figure 2 show the Box-and-Whisker graphic of the three rhythm-related and the syllable duration distributions. It can intuitively be observed how the rhythm has the same distribution before and after the filled pause, and how the prepausal syllable distribution is moved through the right in the graphics, what implies a lengthening of the syllable with respect to the sentence rhythm. Same effect appear for both speakers.

Unit:ms	*Mean*	*Std*	*Lower*	*Upper*
Name		*Deviation*	*Quartile*	*Quartile*
totrh.	154	24	146	169
prerh.	150	35	142	167
posrh.	165	28	150	176
syl$_{-2}$	232	131	145	317
nuc$_{-2}$	108	61	71	123
syl$_{-1}$	378	140	304	424
nuc$_{-1}$	222	70	168	277
paupre	434	326	180	595
paupos	268	298	82	360
fpdur	506	184	406	629

Table 3: Summary statistics for the female Speaker and for filled pauses.

These observations support the fact that there exists a prepausal syllable lengthening in filled pauses. A further issue will be to predict this lengthening. It can also be observed that the FP duration is much larger than the sentence rhythm.

The filled pause duration is significantly larger than the mean rhythm, also its standard deviation is bigger. This is related with the fact that filled pauses are used to re-plan what is going to be said. However, experimental synthesis have shown that not any length sounds natural. We believe that a certain relation between the syllable lengthening and the filled pause duration must exist, i.e. the prepausal length and the filled pause duration will be larger for slower speeches and vice versa.

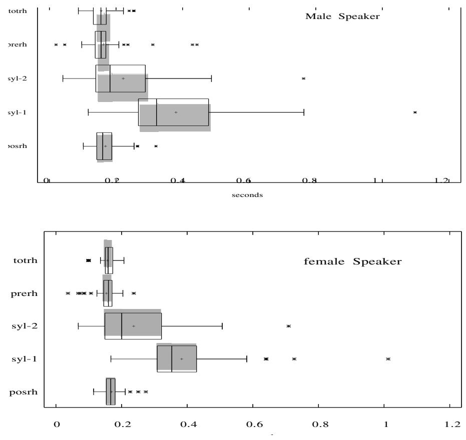


Figure 2: Box-and-Whisker graphics for rhythm variables and syllable lengths for filled pauses.

For these reasons in next sections we will look at the relation between the sentence rhythm and these two variable plus silent pauses' duration.

4.3. Correlation between variables

In order to analyse the relation between the sentence rhythm and the syllable, silent and filled pauses duration; we have calculated the correlation values between all variables. Since the database is small, we have also compute the statistical significance of the correlation values and only significant values ($P < 0.05$) are given. Table 4 summarises all correlation values.

Silent pauses duration do not have significant correlation with any other variable except for the female speaker, in this case they are only correlated with the FP duration. However, since the filled pause duration is an unknown variable, it can not be used to predict the silent pause duration. Therefore, with the approach presented here there is no way to predict the silent pause duration.

Since we have concluded that the rhythm does not change across the sentence, but that the rhythm across the whole sentence, the one previous to the FP and the posterior follow the same distribution instead, there are only two features left to model: $s$yl$_{-1}$ and $f$pdur.

	Male Speaker		**Female Speaker**	
Variable	*syl*$_{-1}$	*fpdur*	*syl*$_{-1}$	*fpdur*
totrh	-0.24	-	-0.45	-
prerh	-	-	-0.47	-
syl$_{-2}$	0.26	-	0.27	-
nuc$_{-2}$	0.32	-	0.56	-
syl$_{-1}$	1	0.29	1	-
nuc$_{-1}$	0.63	0.48	0.39	-
paupre	-	-	-	0.37
fpdur	0.30	1	-	1
paupos	-	-	-	0.39
posrh	-	-	-	0.30

Table 4: Statistically significant correlation between previous syllable length, FP duration a defined variables.

Significant correlation will guide us in order to find independent variables for modelling these features. It can be observed in Table 4 that both features are correlated with utterances that occur in advance in the sentences. For example, *syl*$_{-1}$ is significantly correlated with the previous syllable and also with the total rhythm of the sentence. In addition, *fpdur* is correlated with previous rhythm, the total rhythm, and also syl$_{-1}$ in the case of the male speaker. Unexpectedly, the *fpdur* in the case of the female speaker, is correlated with the silent pauses. However, in Table 1 in Section 3 we have seen that the female speaker do not insert any silent pause in 74% of the utterances.

Since the database was recorded in a studio, we have observed that the female speaker is less systematic in the realisation of such filled pauses, and also less natural. What would explain the lack of significance in the correlation between *fpdur* and *syl*$_{-1}$. However, the filled pause is significantly correlated, in this case, with the posterior rhythm. Moreover, since the rhythm do no change significantly across the sentences, the fact that *fpdur* is correlated with the posterior rhythm implies that is correlated with the other two rhythm variables in the study (i.e. *totrh* and *prerh*), but that lack of data makes this correlation not statistically significant.

In next section we will discuss the use of these correlation between features, in order to generate a regression model for synthesis of filled pauses.

4.4. regression models

When trying to synthesise filled pauses within the unit-selection framework, the first issue to take into account is what units to be used. Here we have choose to record a small database containing disfluencies. Filled pauses was one kind of disfluencies recorded. Therefore, filled pauses units are now available in the inventory to its use for disfluent speech synthesis (see Section 2).

After the unit inventory issue is solved, the desired prosody has to be generated. For this purpose our synthesiser already have a pitch, duration and energy model [15]. However, this model is trained on fluent speech. As we have stated in Section 4.2, it is possible to use state of the art prosody modelling to predict the rhythm of the whole sentence as if it was a fluent sentence, and afterwards some local model can be applied to modify this fluent prosody in order to achieve the desired disfluent one.

For this purpose, in the case case of filled pauses, only two variables need to be predicted: syl_{-1} and *fpdur* (i.e. prepausal syllable length, and filled pause duration). Also the silent pauses (i.e. *paupre*, and *paupos*) should be predicted, but we have not found any significant correlation here. Here we propose to use a multiple regression model due to its simplicity and to that these variables are correlated with the rhythm of the sentence.

Given results from Table 4 the prepausal syllable duration can be predicted by means of the total rhythm of the sentence and its previous syllable. In both cases the syllable duration and the syllable nucleus duration are very correlated thus only one of them is used, the one that gives a better fitting are mentioned here. In the case of the male speaker the regression function proposed is:

$$syl_{-1} = 568 + 0.58 nuc_{-2} - 1.45 totrh \qquad (1)$$

and it fits the data with a 106ms of mean absolute error(MAE). For the female speaker the regression function proposed is:

$$syl_{-1} = 692 + 0.96 nuc_{-2} - 1.68 totrh - 1.07 prerh \qquad (2)$$

and it fits the date with a MAE of 81ms.

The filled pause duration now can be predicted by means of the sentence rhythm but also depends on the pre-pausal lengthening. Since prepausal lengthening is part of the whole model then a cumulative error effect will be produced, since the error done on prepausal length prediction will be passed to regression function for the filled pause duration. The proposed regression function for the male speaker is as follows:

$$fpdur = 338 + 0.86 nuc_{-1} \qquad (3)$$

and for the female speaker this functions is proposed:

$$fpdur = 181 + 1.96 posrh \qquad (4)$$

both functions fit the data with a MAE of 126ms.

5. Comparison with Silent Pauses

As we have said in Section 1 we also want to evaluate whether conclusions concerning filled pauses can also be extended to silent pauses. For this purposes, we have used the whole databases recorded for speech synthesis, which contains about 10h of speech. Same features described in Section 4.1 have been extracted from this database and same statistics have been computed. However, now 15,300 silent pauses are available to compute statistics, what means a much larger amount of examples than for filled pauses.

Unit:ms	*Mean*	*Std*	*Lower*	*Upper*
Name		*Deviation*	*Quartile*	*Quartile*
totrh.	158	17	149	161
prerh.	162	24	149	164
posrh.	160	26	147	165
syl$_{-2}$	164	53	128	196
nuc$_{-2}$	78	27	63	88
syl$_{-1}$	237	61	200	272
nuc$_{-1}$	110	42	80	136
spdur	340	265	116	480

Table 5: Summary statistics for the male speaker and for silent pauses..

Tables 5 and 6 presents summary statistics for silent pauses. Same analysis is presented for filled pauses in Tables 2 and 3. We can observe how again the rhythm do no change in the silent pause, since rhythm previous to the pause and after it follow same distribution than total sentence rhythm.

Unit:ms	*Mean*	*Std*	*Lower*	*Upper*
Name		*Deviation*	*Quartile*	*Quartile*
totrh.	165	13	158	169
prerh.	167	18	158	172
posrh.	165	19	156	170
syl$_{-2}$	173	50	136	204
nuc$_{-2}$	82	24	68	96
syl$_{-1}$	245	64	195	288
nuc$_{-1}$	118	26	104	132
spdur	313	185	196	364

Table 6: Summary statistics for the female speaker and for silent pauses.

Furthermore, the well-known prepausal lengthening is observed. It can be observed more clearly in Figure 3, which is very similar to the corresponding to filled pauses (see Figure 2). Until now, same conclusion extracted from analysing filled pauses are extracted. This means, that it might be possible to predict sentence rhythm without taking silent pauses into account, and afterwards the silent pause prosody (i.e. pause duration plus prepausal syllable length) can be modelled locally.

Also correlations across features have been computed for silent pauses. Table 7 shows the corresponding values. Note that all correlations are significant since a lot more value are given. It can be observed how syl_{-2} is not correlated at all with silent pause duration (named as *spdur*) neither with pre-pausal length syllable. However, syllable length is strongly correlated with the rhythm and the pause length. Of special interest is the correlation value between pre-pausal syllable and pause duration since it is negative. This implies that the longer the prepausal syllable is the shorter the pause. These results are similar than the ones published in [12] claiming that what is perceptually important in hesitations is the sum of the pre-pausal syllable length plus the silent pause. We can conclude from correlations in Table 7 that the faster this speaker talk, the pre-pausal syllable is shorter but there is a longer pause. In contrast, if we talk slowly the pre-pausal syllable is, of course, longer but the silence is shorter.

Finally, note that results are similar than the ones presented

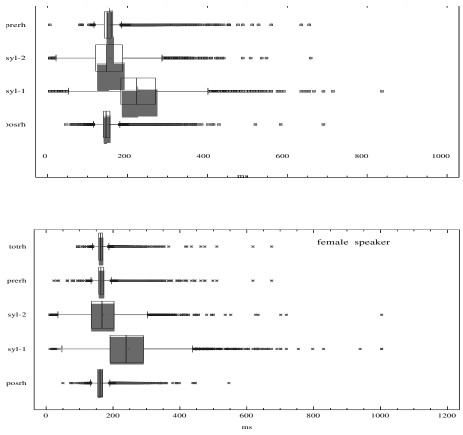


Figure 3: Box-and-Whisker graphics for rhythm variables and syllable lengths for silent pauses.

	Male Speaker		Female Speaker	
Variable	syl_{-1}	*spdur*	syl_{-1}	*spdur*
totrh	0.36	-0.22	0.28	-0.15
prerh	0.37	-0.18	0.33	-0.11
syl$_{-2}$	-0.05	-0.06	-0.13	-0.04
nuc$_{-2}$	-0.04	-0.04	-0.10	-0.02
syl$_{-1}$	1	-0.28	1	-0.03
nuc$_{-1}$	0.07	-0.24	-0.10	-0.05
spdur	-0.28	1	-0.03	1
posrh	0.20	-0.18	0.11	-0.13

Table 7: Statistically significant correlation between previous syllable length, SP duration a defined variables.

here, what means that filled pauses behave in a similar way than silent pauses, except for these negative correlations. Therefore, it could be possible to apply same rules and regression proposed here for filled pauses to silent ones.

6. Conclusion

In the present paper, we have studied the rhythm of filled pauses. Filled pauses may or not contain silent pauses before and after them. Despite in the database used here it appears in few cases, it is plausible to insert both silences in order to avoid coarticulation problems.

The main issue of the study presented here was to find significant correlations between a set of rhythm features in order to be able to predict filled pauses duration and rhythm related effects.

It has been found that when a filled pause is produced there is not any significant rhythm change in the sentence. However, a prepausal lengthening similar to the one produced before silent pauses is produced. The length of this syllable is correlated with the sentence rhythm. In addition, the filled pauses duration is correlated with the prepausal syllable length as well as with the sentence rhythm.

These both findings plus the evidence that global sentence rhythm is no affected by the filled pause presence, leaded us to propose a duration model for speech synthesis. It is linear regression model able to predict prepausal length based on the sentence rhythm, and filled pause duration is predicted using the previously predicted prepausal length and the sentence rhythm by means of another linear regression model.

Informal tests have shown a noticeable improvement with respect to the previously proposed method in [13].

7. References

[1] D. Mostefa, M.-N. Garcia, O. Hamon, and N. Moreau, “Deliverable 16: Evaluation report,” ELDA, Tech. Rep., Sept. 2006. [Online]. Available: http://www.tc-star.org

[2] C. L. Bennett and A. W. Black, “The blizzard challenge 2006,” in *Proceedings of Blizzard Challenge 2006 Workshop*, 2006, Pittsburgh, PA. [Online]. Available: http://www.festvox.org/blizzard/blizzard2006.html

[3] M. Shröder, “Emotional Speech Synthesis: A Review,” in *Proceedings of Eurospeech*, vol. 1, Sept. 2001, pp. 561–564, Aalborg, Denmark.

[4] C. Gobl, E. Bennet, and A. N. Chasaide, “Expressive synthesis: How crucial is voice quality,” in *Proceedings of IEEE Workshop on Speech Synthesis*, Sept. 2002, pp. 91–94, Santa Monica, California.

[5] S. Werner and R. Hoffman, “Pronunciation variant selection for spntaneous speech synthesis - a summary of experimental results,” 2006, dresden, Germany.

[6] J. E. F. Tree, “The effects on of false starts and repetitions on the processing of subsequent words in spontaneous speech,” *Journal of Memory and Language*, vol. 34, pp. 709–738, 1995.

[7] S.-C. Tseng, “Grammar, prosody and speech disfluencies in spoken dialogues.” Ph.D. dissertation, Department of Linguistics and Literature, University of Bielefeld, Apr. 1999.

[8] H. H. Clark, “Speaking in time,” *Speech Communication*, vol. 36, no. 1-2, pp. 5–13, Jan. 2002.

[9] J. E. F. Tree, “Listeners’ uses of *um* and *uh* in speech comprehension,” *Memory & Cognition*, vol. 29, no. 2, pp. 320–326, 2001.

[10] M. Watanabe, K. Hirose, Y. Den, and N. Minematsu, “Filled pauses as cues to the complexity of following phrases,” in *Proc. of Eurospeech*, September 2005, pp. 37–40, lisbon, Portugal.

[11] S. Sundaram and S. Narayanan, “An empirical text transformation method for spontaneous speech synthesizers,” in *Proc. of Eurospeech*, Sept. 2003, Geneva, Switzerland.

[12] R. Carlson, K. Gustafsson, and S. Strangert, “Modelling hesitation for synthesis of spontaneous speech,” in *Proceedings of Speech Prosody 2006*, Dresden, may 2006. [Online]. Available: http://www.speech.kth.se/prod/publications/files/1087.pdf

[13] J. Adell, A. Bonafonte, and D. Escudero, “Disfluent speech analysis and synthesis: a preliminary approach,” in *in Proc. of 3th International Conference on Speech Prosody*, May 2006, dresden, Germany. [Online]. Available: http://gps-tsc.upc.es/veu/personal/jadell/

[14] G. A. Toledo, *El ritmo en el español : estudio fonético con base computacional*, ser. Biblioteca románica hispánica ; II. Estudios y ensayos., Gredos, Ed., 1988, no. 361.

[15] A. Bonafonte, P. D. Agüero, J. Adell, J. Pérez, and A. Moreno, “Ogmios: The UPC text-to-speech synthesis system for spoken translation,” in *TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain, June 2006, pp. 199–204.

SSW6

universitätbonn

Quantitative Analysis of F_0 Contours of Emotional Speech of Mandarin

Wentao Gu and Tan Lee

Department of Electronic Engineering, the Chinese University of Hong Kong, Hong Kong
{wtgu, tanlee}@ee.cuhk.edu.hk

Abstract

The F_0 characteristics of Mandarin speech in four basic emotions (anger, fear, joy, and sadness) as well as in neutral reading are compared quantitatively. Two approaches are employed: analysis of surface features from time-normalized F_0 contours, and analysis-by-synthesis of time-intact F_0 contours based on the command-response model, which turns out to be also applicable to emotional speech. For surface F_0 features, the height and range of F_0, the local tonal variation, and the sentential F_0 declination are all investigated. In model-based analysis, the parameters of both phrase and tone commands are compared systematically. The study shows that those surface F_0 phenomena can be explained better by the model-based approach, which can later be used in F_0 generation for emotional speech synthesis.

1. Introduction

During the last two decades, there has been an inspiring growth in the works on emotional speech. Especially, the technologies on emotional speech synthesis and automatic recognition of emotional speech have progressed steadily by the aid of data-driven statistical methods, without having attained a really clear picture of the acoustic characteristics of various vocal emotions. However, such basic questions as to how a given emotion is expressed in speech still need to be answered, not only from scientific considerations but also for a further improvement of the related practical technologies.

Although it has been well known that both segmental and suprasegmental (prosodic) features play important roles in conveying vocal emotions [1], the latter is usually regarded to be primary. In the present study, we shall only investigate the characteristics of F_0 contours in vocal emotion expression.

In contrast to a great number of analysis works for non-tone languages like English (e.g. [1]), rather few studies on prosodic features of vocal emotions in tone languages like Mandarin Chinese have been reported in literature. The reason may partly lies in that the presence of lexical tones significantly constrains the manipulation of F_0 in emotional speech, as discussed in [2] where three F_0 features (F_0 slope, F_0 variation, and ΔF_0) are measured. Therefore, although the acoustic realizations of vocal emotions share many common properties across languages, there may be still some attributes specific to tone languages that need more investigation.

Among those very few studies on Mandarin, Yuan et al. [3] claimed that anger and fear are mainly realized by phonation; joy is mainly realized by F_0; and sadness is realized by both. Zhang et al. [4] investigated F_0, duration as well as short-time amplitude, not only at the sentential layer but also on the syllable-by-syllable base; their study also showed that stressed words carry more identifiable acoustic features for vocal emotions than unstressed words. Table 1 summarizes the qualitative results on F_0 features of Mandarin speech in the four basic emotions obtained in these studies, in company with those for English as obtained in [1]. It should be noted that the features in [2] and [3] are defined differently. As shown, the tendencies on both height and range of F_0 differ only slightly between the two languages.

Table 1: Summary of F_0 features for emotional speech in literature, [1] for English and [2, 3] for Mandarin

Lit.	F_0 feature	Anger	Fear	Joy	Sadness
Murray et al. [1]	Average	very much higher	very much higher	much higher	slightly lower
	Range	much wider	much wider	much wider	slightly narrower
	Inflection	abrupt on stressed	normal	smooth upward	downward
Yuan et al. [2]	Height	high	high	high	low
	Fluctuation of top-line	large	small	large	small
Zhang et al. [3]	Average	highest	higher	higher	slightly lower
	Range	widest	slightly wider	wider	slightly narrower
	Stressed in contrast to unstressed words	higher	higher	higher	wider range

However, many important phenomena in F_0 variation have not been studied yet. For instance, how lexical tone patterns and sentential F_0 declination vary with those emotions needs a systematic investigation. In addition, all the aforementioned works inspect surface F_0 features such as max/min/mean F_0 values, slope of F_0 curve, or range of F_0 values in a target syllable or in a larger domain. This kind of analysis, however, has the following drawbacks. First, it does not separate global intonation and local tone patterns explicitly, and hence only gives a confounded result. Second, the surface measurements are phenomenological and cannot capture the essential characteristics of F_0 movements efficiently. Third, the surface measurements are vulnerable to microprosody and noises in F_0 extraction.

For emotional speech synthesis, the construction of prosodic rules is necessary. Hence, a quantitative model giving a parametric representation of F_0 contours needs to be introduced. In this sense, the command-response model for the process of F_0 contour generation [5] is quite efficient, which was originally proposed for Japanese but later also applied to many other languages including Mandarin [6]. The method has been employed to analyze quantitatively F_0 contours of emotional speech of Japanese [7]. In the present study, we will investigate whether the model can be applied successfully to emotional speech of Mandarin, and if so, what parametric differences can be captured between those vocal emotions as a result of a fully quantitative analysis. The results will also be compared with the analysis of surface F_0 features observed directly from time-normalized F_0 contours.

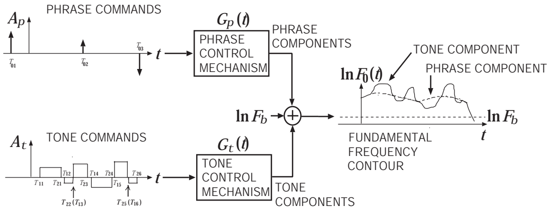


Figure 1: The command-response model for the process of F_0 contour generation.

2. The command-response model for F_0 contours of Mandarin

Figure 1 shows the diagram of the command-response model. It describes F_0 contours in the logarithmic scale as the sum of phrase components, accent/tone components, and a baseline level $\ln F_b$. The phrase commands (pulses) produce phrase components through the phrase control mechanism, giving the global shape of F_0 contours, while the accent/tone commands (pedestals) generate accent/tone components through the accent/tone control mechanism, characterizing the local F_0 changes. Both mechanisms are assumed to be critically-damped second-order linear systems. The model can give highly accurate approximations to F_0 contours from a small number of linguistically meaningful parameters, and has been applied to many languages [5]. The details of model formulation are described in [5]. In the present study, the constants α, β, and γ in the model are fixed at 3.0 (1/s), 20.0 (1/s), and 0.9, respectively, following the previous studies.

Unlike Japanese, tone languages usually require both positive and negative tone commands due to faster local F_0 changes. For a specific tone language, a set of tone command patterns needs to be specified in the model. As listed in Table 2, Mandarin has four lexical tones, as well as a neutral tone – any lexical tones can be neutralized in an unstressed syllable. The rightmost column of the table gives the tone command patterns for each tone [6]. The neutral tone does not have a stable tonal shape and hence it has no intrinsic tone command pattern; instead, it varies largely with the preceding tone.

Table 2: Mandarin tone system

Tone type	Pitch feature	Tone code	Command pattern
T1	high	55	positive
T2	rising	25	negative to positive
T3	low	21(4)	negative
T4	falling	51	positive to negative
T0	neutral	variable	context-dependent

3. Speech data

We designed ten short sentences (part of them are from [4]), each consisting of 4 to 9 characters. They are declarative sentences or wh-questions, hence inherently with a declining intonation in neutral reading. The sentence texts are neutral (i.e. not literally associated with any specific emotion) but can be placed in different contexts to induce various emotions. Each sentence was uttered in five styles: four basic emotions (anger, fear, joy, sadness) and a neutral reading at normal speech rate (i.e. neutral emotion). Here, anger and joy are active emotions, while fear and sadness are passive emotions. Each utterance was recorded with three repetitions at consistent degrees of emotion expression.

Two speakers, one male and one female, who were both graduate students, were asked to record the speech. Before the recording of each utterance, a designed context was prompted by the instructor to help the speaker induce the required emotion. For instance, for the following sentence, of which the text is not inherently associated with any of the four basic emotions: "Ju1 ran2 hui4 fa1 sheng1 zhe4 zhong3 shi4" (Unexpectedly this thing happened), a prompt story that apparently causes anger, fear, joy, and sadness was described to the speaker respectively before his/her recording.

Although there are always speaker differences in vocal emotion expression, our preliminary study shows that these two speakers largely share the common strategy of expression (though differ in quantity). Hence, in the present work, only the analysis of the female speaker's data will be presented.

4. Method of data analysis

F_0 values were extracted by a modified autocorrelation analysis, while syllables were segmented manually by visual inspection of waveform and spectrogram.

One difficulty in comparing F_0 contours lies in that they are not aligned in time. The F_0 contour is not a unit-based measurement like syllable duration; instead, it is a time-varying sequence which implicitly involves the timing information. For a direct comparison, F_0 contours need to be time-normalized. Hence, the measured F_0 values were first smoothed and interpolated for voiceless intervals to produce a continuous F_0 contour. Then, ignoring durational differences, a time-normalized F_0 contour was obtained by extracting a 10-point (equally spaced) sequence of F_0 values in each syllable from the continuous F_0 contour.

Unlike surface feature analysis, model-based analysis tries to give an optimal approximation to the entire F_0 contour through a set of parameters. This procedure, named analysis-by-synthesis, was first done manually with the aid of syllable timing and linguistic information such as tone identity and syntactic structure, and later the parameters were optimized by successive approximation, as discussed in more details in [6]. It should be noted that this is not merely a mathematical procedure of curve fitting; instead, the minimum error criterion is only effective under the linguistic constraints to ensure the linguistic meaningfulness of the analysis.

For the utterances of a fixed speaking style, the baseline frequency F_b can be considered to be constant for the sake of simplicity of modeling, and it is usually initialized by visual inspection of F_0 contours of many utterances in the same style.

In read speech of neutral emotion, tone commands in each syllable should basically comply with the inherent command patterns for the particular tone type, though closely neighboring tone commands with the same polarity are allowed to be merged. In emotional speech, however, the situation becomes complicated due to frequent reduction, neutralization, or change of lexical tones. Hence, tone identities should be based on acoustic realization instead of linguistic form. Also, the following two heuristic rules can be adopted. First, some tone commands may disappear, but it rarely occurs that the polarity of a tone command is reversed. Second, the stressed syllables tend to preserve the canonical form of tones better and hence a better coincidence with the inherent tone command patterns should be given there.

The occurrences of phrase commands are largely aligned with major syntactic boundaries and can be determined by comparison of syllabic F_0 pattern and the canonical form of tones, by comparison of F_0 patterns in adjacent tones, and sometimes also with the aid of prosodic perception. Phrase commands are only assigned when necessary and linguistically meaningful. At many places, whether to add a very small phrase command or not usually has little effect on the accuracy of approximation; in this case, we do not add it.

5. Results

5.1. Analysis of time-normalized F_0 contours

Figure 2 shows the average time-normalized F_0 contours of the utterances of five sentences (one sentence for each panel). For each sentence, the F_0 contours (averaged over the three repetitions) for five emotions (including neutral) are plotted. From the figure the following characteristics are observed:

(1) The five emotions are distinctly clustered into two groups: one is anger/fear/joy and the other is sadness/neutral, the former showing significantly higher F_0 than the latter.

(2) Anger shows the widest F_0 range, and usually gives the highest sentential maximum F_0. Within the higher group, anger also gives the lower sentential minimum F_0 than fear and joy. Especially, compared with other emotions, anger always raises F_0 in a certain syllable to produce an F_0 peak, as shown in /zhe4/, /jin4/, /zhe4/, /yue4/, and /zhe4/ in the five sentences, respectively (they happen to be all of T4 here). The F_0 values immediately before the syllable of peak F_0 are usually also raised, while those after the peak syllable are conspicuously lowered, hence resulting in a larger F_0 declination than other emotions.

(3) Within the higher group, fear shows the narrowest F_0 range. Especially, at the end of an utterance, fear gives higher F_0 than anger and joy (in fact, the highest among the five emotions). In other words, fear shows a weaker sentential F_0 declination than anger and joy.

(4) Sadness shows both the lowest F_0 value and the narrowest F_0 range. Especially, F_0 contours of sadness and neutral often coincide at the major F_0 valleys of the latter, but at other positions F_0 contours of sadness are significantly lower and flatter than those of neutral.

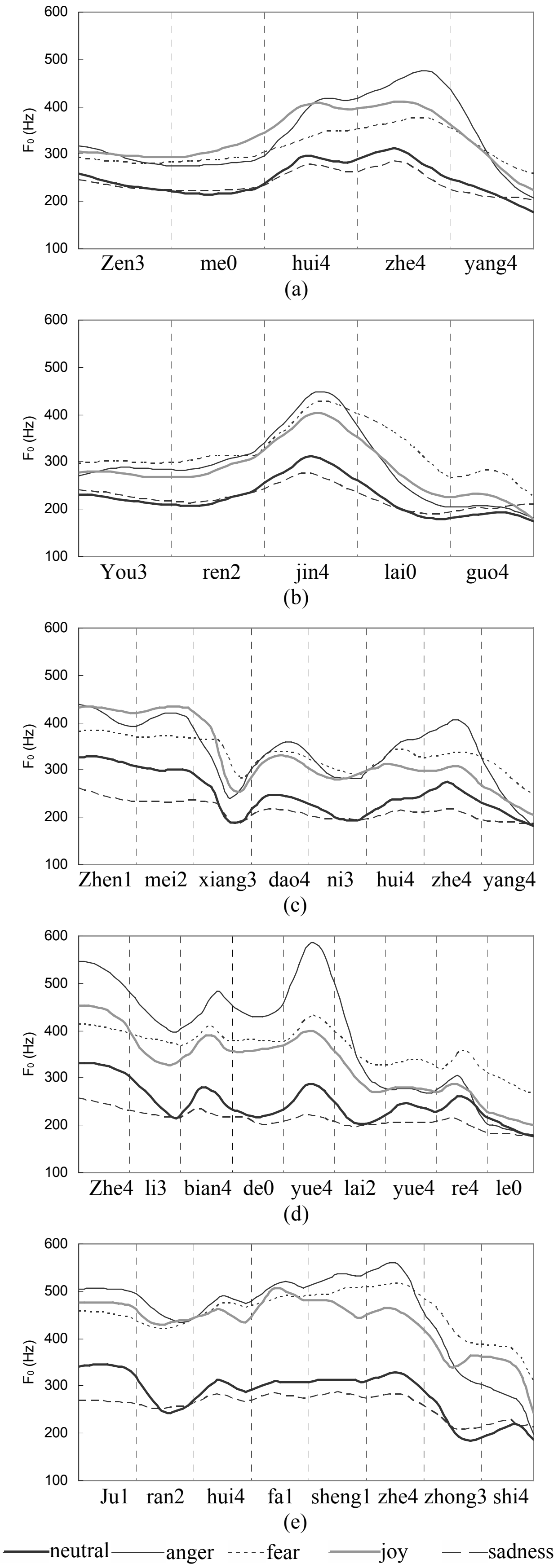
(5) Among the five emotions, fear and sadness (both are passive) show similar F_0 trajectories (nearly parallel) and differ mainly in the height – fear is higher (and as we will show later, they also differ in tempo – fear is faster). These two give flatter F_0 contours than others. This is especially distinct in panels (c) ~ (e).

(6) Among the five emotions, fear and joy compose a pair that seems the most difficult to be distinguished. The major distinction is that fear gives a flatter F_0 contour, i.e. with a narrower F_0 range. In the earlier part of an utterance fear has comparable or even lower F_0, but in the later part (especially at the end) it keeps significantly higher F_0 than joy.

Besides the above findings, we further look separately into sentential intonation and syllabic tones, though at this stage these two components cannot be separated effectively.

On the one hand, it is well known that F_0 declines gradually in an utterance of neutral reading (except some yes/no questions). This can roughly be observed from Figure 2, especially in those relatively longer sentences. For instance, among the five T4 syllables in the neutral reading of sentence (d), the utterance-initial /zhe4/ is higher than the utterance-medial /bian4/ and /yue4/, which are again higher than /yue4/ and /re4/ near the end of the utterance.

In emotional speech, the magnitudes of sentential F_0 declination rank as follows: anger > joy > neutral > fear > sadness, which also indicates the order of active emotions > neutral > passive emotions. This can be seen from Table 3, where the ratios of utterance-final F_0 to utterance-initial F_0



(a)

(b)

(c)

(d)

(e)

Figure 2: Average time-normalized continuous F_0 contours of the utterances in five emotions (including neutral).

Table 3: The ratios of utterance-final F_0 to utterance-initial F_0, approximately indicating sentential declination

	Neutral	Anger	Fear	Joy	Sadness
(a)	0.69	0.65	0.89	0.73	0.83
(b)	0.76	0.67	0.76	0.65	0.87
(c)	0.56	0.42	0.65	0.47	0.71
(d)	0.53	0.33	0.64	0.44	0.69
(e)	0.55	0.39	0.67	0.51	0.78
Average	0.61	0.47	0.72	0.55	0.78

Table 4: The percentages of correctly identified lexical tones when lifted out of continuous speech (%)

	Neutral	Anger	Fear	Joy	Sadness	Average
T1	100	100	100	100	75	95
T2	33	33	8	42	33	30
T3	100	67	44	94	50	81
T4	71	59	37	57	35	52
All	75	62	43	68	43	58

are given. Although the difference between the two ends is not entirely ascribed to sentential declination, a comparison of such differences can roughly show the effects of different emotions on sentential declination. A more accurate analysis will be given in the model-based approach in the next section.

On the other hand, it is also well known that tones in continuous speech deviate from their canonical form in isolated syllables, and tone identities of some syllables may be completely lost, especially in spontaneous speech. For example, Tseng's study [8] showed that in fluent spontaneous speech of Mandarin only 36% of syllables preserve their lexical tones. The reason lies in that speech production is a compromise between maximizing communicative function and minimizing articulatory effort, as suggested by the hypo- and hyper-articulation theory [9].

In order to investigate the acoustic realization of lexical tones in emotional speech, we conducted a perceptual test, in which each syllable (except those that can be predicted from the text to be in neutral tone) was lifted out of the utterance and played back to a native subject for a perceptual identification of tone ('unidentifiable' is also set as an option).

Table 4 lists the percentages of correctly identified lexical tones. Compared with neutral reading, the proportions of identifiable tones decrease significantly in emotional speech, especially in passive emotions (fear and sadness), indicating that the use of hypo-speech increases in expressing emotions (especially for passive ones). This is consistent with the observation that F_0 range is narrowed in fear and sadness. Besides, the contour tones (T2 and T4) are found to be less preserved than the level tones (T1 and T3). It coincides with our model-based analysis, in which a pair of tone commands inherently associated with a contour tone is frequently reduced to a single tone command.

5.2. Model-based analysis of F_0 contours

Table 5 gives the average model parameters as a result of model-based analysis-by-synthesis of F_0 contours of all the utterances. Since F_0 contours of the five emotions are clearly clustered into two groups according to the overall F_0 level, two different F_b values are set heuristically for the two groups, respectively. Because utterance-medial phrase commands may occur at different positions and some utterances even do not have medial phrase commands, only the magnitude (A_p) of utterance-initial phrase command is listed for comparison.

Table 5: Average model parameters for F_0 contours of emotional speech

Parameters	Neutral	Anger	Fear	Joy	Sadness
F_b [Hz]	160	220	220	220	160
Num of phrase cmd.	2.33	1.08	1.75	1.42	2.25
Utterance-initial A_p	0.49	0.52	0.49	0.49	0.43
Num of tone cmd.	6.58	7.42	6.83	7.17	5.50
Abs amp. tone cmd.	0.35	0.54	0.37	0.39	0.23
Dur of tone cmd. [s]	0.10	0.08	0.06	0.07	0.10

For tone commands, both absolute amplitude and duration (from onset to offset) are given.

From the statistics the following tendencies are observed:

(1) Baseline F_b: (anger, fear, joy) > (neutral, sadness).

(2) Number of phrase commands: (neutral, sadness) > fear > joy > anger.

(3) Magnitudes of phrase commands: sadness < others.

(4) Number of tone commands: anger > joy > fear > neutral > sadness. Especially, sadness gives an obviously smaller number than others.

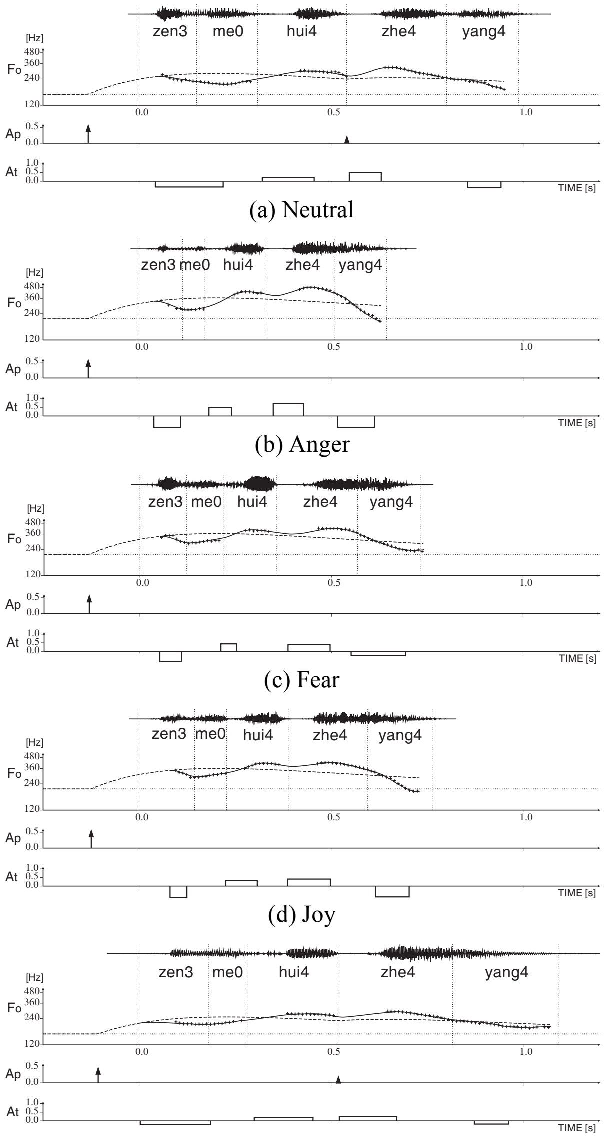
(5) Absolute amplitudes of tone commands: anger > joy > fear > neutral > sadness. This rank is the same as that for the number of tone commands, but the differences between joy, fear, and neutral are not as significant as between others. Both (4) and (5) suggest how big local F_0 variation is: fewer and smaller tone commands lead to smaller local F_0 variation.

(6) Duration of tone commands: (sadness, neutral) > (anger, joy, fear). Although duration of tone commands is not inherently correlated with syllable duration (as we will show in a later example), in many cases they are approximately in proportion. The result indicates that the speech in neutral or sadness is slower than in other emotions, which is basically consistent with the report on syllable/sentence duration in [4].

The different magnitudes of sentential F_0 declination as discussed in Section 5.1 can be explained from a combination of the above analyses. Among the emotions giving comparable speaking rates, a larger number of phrase commands indicates more F_0 resets and thus results in a weaker sentential F_0 declination – hence the declination ranks as anger > joy > fear. Sadness gives a weaker declination than neutral, mainly due to the smaller phrase command. However, a comparison between the emotions giving different speaking rates is a little complex. Although F_0 of slower speech may decline more due to the longer stretch of phrase components, the difference in the number of phrase commands should also be considered because slower speech usually needs more phrase commands – in this case the ultimate effect is a combination of these two factors.

In comparison, this model-based analysis captures sentential F_0 declination more accurately than the surface feature analysis in Section 5.1, because phrase and tone components are separated explicitly in the framework of the model and hence the differences in local tonal variation due to different amplitudes of tone commands can be excluded.

The above simple statistics, however, are still not sufficient to give a full view of the parametric differences between vocal emotions. The detailed distributions of model parameters need also to be investigated. For simplicity of illustration and comparison, we select the short sentence shown in Figure 2(a) as an example, which happens to give the same number of tone commands in all the five emotions. Figure 3 shows the results of analysis-by-synthesis of F_0 contours of five utterances in the respective emotions. The crossed symbols indicate the measured F_0 values, while the solid, dotted, and dashed lines indicate the approximated F_0



(a) Neutral

(b) Anger

(c) Fear

(d) Joy

(e) Sadness

Figure 3: Analysis-by-synthesis of F_0 contours of the utterances in five different emotions.

contours, the baseline frequencies, and the contributions of phrase components, respectively. The differences between the approximated F_0 contours and the phrase components correspond to the tone components.

As shown, F_0 contours of speech in all these emotions can be approximated at a very high accuracy with a small number of commands in the framework of the model. Among the five syllables, /zen3 me0/ are merged into a single negative tone command (because /me0/ is in neutral tone); /hui4/ almost loses the falling feature and is reduced to a high level tone – hence a positive tone command; and /zhe4 yang4/, as a prosodic word composed of two consecutive T4 syllables, shows the well-known tone coarticulation (viz., both T4's are reduced to half-falling), and hence can be modeled like a single T4 syllable – a positive tone command followed by a negative one. Besides, anger, fear, and joy only give an utterance-initial phrase command, while neutral and sadness also give an additional phrase command (with a lower magnitude than the utterance-initial one) before the utterance-final prosodic word /zhe4 yang4/, because neutral or sad speech is slower than others and hence needs more F_0 resets.

Table 6: Model parameters for F_0 contours of the example sentence in five emotions

		Neutral	Anger	Fear	Joy	Sadness
F_b [Hz]		160	210	210	210	160
1st A_p		0.48	0.50	0.44	0.49	0.44
2nd A_p		0.15	--	--	--	0.14
Amp. of tone cmd.	zen3-me0	-0.32	-0.70	-0.55	-0.53	-0.25
	hui4	0.26	0.50	0.44	0.42	0.22
	zhe4	0.45	0.67	0.45	0.41	0.27
	yang4	-0.35	-0.59	-0.34	-0.58	-0.14
Dur. of tone cmd. [s]	zen3-me0	0.16	0.06	0.05	0.06	0.15
	hui4	0.11	0.06	0.04	0.07	0.10
	zhe4	0.08	0.10	0.11	0.12	0.10
	yang4	0.09	0.11	0.10	0.10	0.14

Table 6 lists the relevant model parameters averaged over three repetitions of utterances for this example sentence in all the five emotions. Although the overall tendencies are consistent with the statistics given in Table 5, the differences in tone command parameters are not homogenous in the utterance; instead, they vary with word position, and some local features may be more important in charactering the particular emotions, for instance:

(1) In the earlier syllables /zen3 me0 hui4/, fear and joy give comparable amplitudes of tone commands, as is consistent with the average result shown in Table 5, but in the later part, especially in the utterance-final syllable /yang4/, fear gives significantly higher tone commands than joy. This explains the surface observation that fear ends with a higher F_0 and shows a weaker sentential F_0 declination than joy.

(2) In the earlier syllables /zen3 me0 hui4/, anger, fear, and joy give much shorter tone commands than neutral speech, as is consistent with the average result shown in Table 5, but in the final two syllables /zhe4 yang4/ the case turns opposite, viz., they give even longer tone commands than neutral, though syllable duration is still shorter – in fact the longer tone commands here are not due to syllable duration but due to the larger local F_0 variation than in neutral speech. For sadness, on the other hand, the durations of tone commands in the earlier syllables /zen3 me0 hui4/ are comparable with those in neutral speech, while the tone commands in the final two syllables /zhe4 yang4/ are longer, which is consistent with the longer syllable durations than in neutral speech. It should be noted that the final two syllables /zhe4 yang4/, as a prosodic word, is perceptually the most prominent part in the emotional utterances, as is consistent with Li's finding that in emotional speech sentence stress tends to be placed on the utterance-final prosodic word [10].

Hence, the model-based parametric analysis shows that the prosodic characteristics of emotional speech vary systematically with word position, or with the status of sentence stress. The above results of analysis for this short sentence are similarly observed in the utterances of other sentences. The major difference lies in that for those longer sentences the numbers of tone commands also vary with vocal emotions, as indicated by the statistics shown in Table 5.

6. Discussion and conclusion

The F_0 characteristics of emotional speech of Mandarin are investigated, both by surface feature analysis and by model-based analysis. Although the present study is still preliminary and does not involve a large amount of data, many valuable results have been obtained.

The quantitative differences in the time-normalized F_0 contours for different vocal emotions can be summarized qualitatively as follows. The five emotions are first clustered into two groups, i.e., anger/fear/joy vs. neutral/sadness, the former giving globally higher F_0 contours than the latter. Among the higher group, anger shows the largest sentential declination and the widest F_0 range, and usually an F_0 peak is raised to give a sentence stress, which is often placed on the utterance-final prosodic word; in the remaining, joy shows larger sentential declination and wider F_0 range than fear. Among the lower group, sadness gives smaller sentential declination and narrower F_0 range than neutral speech. Besides, in emotional speech the lexical tones are less preserved than in neutral speech, and the preservation of lexical tones is even less in passive emotions than in active emotions. Lastly, although syllable duration is not discussed in depth here, we found a rather similar result as that reported in [4], namely, neutral and sadness are slower than others.

The command-response model is applied successfully to F_0 contours of emotional speech of Mandarin, though tone command patterns show larger variation due to the increase in the use of hypo-speech in vocal emotion expression. The observations on surface F_0 features can be explained better by the model-based analysis. The global F_0 level is represented by the baseline frequency; the sentential F_0 declination is characterized by the number as well as the magnitudes of phrase commands; and the local F_0 variation is described by the amplitude and duration of tone commands, which also vary systematically with word position in the utterance.

In comparison, analysis of surface F_0 features is straight-forward, but durational information has to be discarded in comparing time-normalized F_0 contours. More importantly, such surface analysis cannot be used directly in emotional speech synthesis. Model-based analysis-by-synthesis of F_0 contours, on the contrary, is more efficient in capturing the essential characteristics of F_0 movements if a good model is introduced, though we believe that the two approaches can be employed together to give a better validation. With a set of quantitative parameters characterizing different vocal emotions, the model-based approach can be used for F_0 generation in emotional speech synthesis [11].

7. References

[1] Murray, I.R. and Arnott, J.L, “Toward the stimulation of emotion in synthetic speech: A review of the literature on human vocal emotion”, *JASA*, 93: 1097-1108, 1993.

[2] Ross, E.D., Edmondson, J.A., and Seibert, G.B., “The effect of affect on various acoustic measures of prosody in tone and non-tone language: A comparison based on computer analysis of voice,” *Journal of Phonetics*, 14: 283-302, 1986.

[3] Yuan J., Shen, L., and Chen, F., “The acoustic realization of anger, fear, joy and sadness in Chinese,” *Proc. ICSLP*, pp.2025-2028, Denver, USA, 2002.

[4] Zhang, S., Ching, P.C., and Kong, F., “Acoustic analysis of emotional speech in Mandarin Chinese,” *Proc. ISCSLP*, pp.57-66, Singapore, 2006.

[5] Fujisaki, H. “Information, prosody, and modeling – with emphasis on tonal features of speech,” *Proc. Speech Prosody*, pp.1-10, Nara, Japan, 2004.

[6] Fujisaki, H., Wang, C., Ohno, S., and Gu, W., “Analysis and synthesis of fundamental frequency contours of Standard Chinese using the command-response model,” *Speech Communication*, 47: 59-70, 2005.

[7] Hirose, K., Kawanami, H., and Ihara, N., “Analysis of intonation in emotional speech,” *Proc. ESCA Workshop on Intonation*, pp.185-188, 1997.

[8] Tseng, C., *An Acoustic Phonetic Study on Tones in Mandarin Chinese (2nd ed.).* Institute of Linguistics, Academia Sinica, Taiwan, 2006.

[9] Lindblom, B., “Explaining phonetic variation: a sketch of the H&H theory,” In *Speech Production and Speech Modeling*, pp. 403-439, Kluwer Academic Publishers, 1990.

[10] Li, A., 情感句重音模式, *Proc. 7th Phonetic Conference of China*, Beijing, 2006.

[11] Hirose, K., Sato, K., Asano, Y., and Minematsu, N., “Synthesis of F_0 contours using generation process model parameters predicted from unlabeled corpora: application to emotional speech synthesis,” *Speech Communication*, 46: 385-404, 2005.

Maximum-Likelihood Dynamic Intonation Model for Concatenative Text to Speech System

Slava Shechtman

IBM Research Laboratory, Haifa

slava@il.ibm.com

Abstract

In this work we present a Maximum Likelihood (ML) joint pitch curve modeling, inspired by HMM TTS synthesis concept. This model provides an optimal solution for the coarse target intonation curve (3 points per syllable) and incorporates both static and dynamic pitch values for better utterance intonation modeling. The coarse intonation curve may be optionally combined with the original pitch extracted from the concatenated units, by a technique, named *microprosody preservation*, which is also described. The latter is intended for reducing pitch modification ratio and improving sound naturalness for large-scale concatenative TTS systems. The proposed model was successfully applied on IBM's trainable concatenative TTS system improving the subjective intonation quality.

1. Introduction

Modern text-to-speech (TTS) systems are intended to support variety of languages, while maintaining natural speech quality. The growing demand for fast introduction of new languages favors fully trainable systems, which minimize the need for hand-crafted voice building process. High quality prosody generation is considered to be essential for natural sounding TTS systems. Prosody is a combination of a number of factors such as fundamental frequency (pitch), duration, energy and pauses. Here we only consider pitch, which is recognized as the most prominent factor for the perception of prosody [1].

Many state-of-the-art TTS system are based on unit-selection and concatenation [2] - [9]. Most of those systems have some rule-based or probabilistic model for prosody [2] - [8] . This model is either applied directly on the synthesized speech [8] or used as a target in the unit selection stage, while the actual prosody is derived from the selected units themselves [2][6][7].

Fully trainable intonation modeling and generation is crucial for multi-language naturally sounding concatenative TTS (CTTS) systems. Classification and Regression trees (CARTs), based on log-frequency pitch values are widely used for target intonation generation in those systems [2][3][6], and are known to perform at least as good as other intonation models [11].

The main disadvantage of conventional CART intonation modeling, based on log-frequency modeling, is that it does not always produce naturally sounding curve. In that case the CART model results either in abruptly changing intonation curve for large trees (which has to be empirically smoothed), or in over-smoothed solution for small trees, that sounds monotonous.

Hence, many CTTS systems do not actually synthesize speech with model-based intonation (which is usually very coarse, containing 3 points per syllable [2][3][6]), but rather use it only as a target in the unit-selection stage [2][6][7]. The final intonation may be a smoothed (and possibly processed) *segment intonation* (i.e. obtained from the selected units or *segments*) [6][7], or some combination of the target intonation with the segment intonation. Concatenative TTS systems, working without explicit pitch modeling, were also reported [9]. The actual segments are usually selected by a dynamic search, minimizing the distance from target to actual prosody summed with spectral and pitch transition errors. Direct usage of the *segment intonation* may result in undesired and uncontrolled final intonation of utterances. However, for long contiguous portions of speech, aligned with target prosodic content, this direct prosody copying is highly desirable, because it should increase the naturalness of the synthesized speech [7][10].

In the following work we will present a maximum likelihood solution for the CART intonation model, taking into consideration both absolute and differential pitch values to create target intonation. This model will result in an improved target intonation curve.

In addition, a microprosody preservation technique will be presented, allowing combining natural pitch fluctuations derived from the selected segments, with a target intonation curve, matching the textual and semantic context.

The work is organized as follows. First, the conventional CART intonation model, used by IBM's CTTS, is reviewed. Then, its dynamic parameter extension and maximum likelihood solution will be depicted. After that, the microprosody preservation technique will be described. Finally, the results of application of the proposed algorithms to the current IBM CTTS system will be presented and discussed.

2. Maximum-Likelihood (ML) Dynamic intonation model

2.1. Simple CART intonation modeling

The current IBM CTTS [2][3][10] intonation model uses phonetic and semantic features, gathered from the input text, to predict three pitch values per syllable.

The features include, among others:

- lexical stress of a syllable in a word
- word stress in a phrase
- offset of the current word from the beginning and the end of the phrase
- offset of the current syllable related to the word boundary and the stressed syllable in the word
- part of speech of the current word
- phonetic context

For each syllable, the feature vector associated with that syllable along with the feature vectors associated with the two syllables to the left and to the right are concatenated, and associated with an observation vector. The observation vector consists of three pitch values (in log-Hertz), obtained from the beginning of the first syllable's sonorant, the center of the syllable nucleus and the end of its last sonorant. From these feature vectors and observations, a decision tree (CART) is built. Only mean pitch values are used for intonation modeling.

During synthesis, the same features are extracted and used for tree traverse. The mean pitch values are optionally smoothed and used as an additive in construction of the target cost for the segment-selection dynamic search [2][3], This process inherently assumes normal i.i.d. distribution of consecutive syllable pitch triads. The final intonation curve is extracted from the segments selected from the voice database. (In the current system, usually each phone is composed of three such segments.)

It should be noted that the i.i.d. assumption, used in the construction of this model, is not quite realistic, because the pitch of a given syllable is heavily dependent on its surroundings. Moreover, even the Gaussian distribution assumption is not always true for cluster data modeling, because of high variability of spoken intonation. (In general, Gaussian Mixture Models could describe cluster distribution in more detail.)

Therefore, this target pitch cannot be used directly for the synthesis, but rather as an additive factor in the overall segment selection cost, combined with pitch discontinuity as well as spectral transition costs [2][3].

2.2. Dynamic features for CART intonation modeling

Generally, the incorporation of cross-syllable dynamic observations together with intra-syllable observations is highly desirable in order to give an expression of the inherent pitch curve smoothness. In [6] a decision tree for pitch transitions is used in addition to the syllable pitch tree for target pitch cost, although those are not combined together.

In the current work we propose to extend the static intonation features triad [4] to include cross-syllable dynamic features. The pitch measurements are non-uniformly spaced in time, so we choose to use time-normalized differences of static observations. The timing of observations used for difference calculation is chosen to guarantee non-zero time interval between the observation instances (See Figure 1).

Let

$$\mathbf{P}_1(n) = \left[P_{start}(n) \quad P_{mid}(n) \quad P_{end}(n)\right] = \left[\log p_{start}(n) \quad \log p_{mid}(n) \quad \log p_{end}(n)\right] \tag{1}$$

be the pitch observation vector of the n-th syllable, where the observations are taken at the beginning, the center and end of the syllable, denoted as $T_{start}(n)$, $T_{mid}(n)$ and $T_{end}(n)$, accordingly. We create a new pitch observation vector $\mathbf{P}_2(n)$, by extending $\mathbf{P}_1(n)$ with dynamic observations, as follows:

$$\mathbf{P}_2(n) = \Bigg[\mathbf{P}_1(n) \quad \alpha\frac{P_{start}(n)-P_{mid}(n-1)}{T_{start}(n)-T_{mid}(n-1)} \quad \alpha\frac{P_{mid}(n)-P_{end}(n-1)}{T_{mid}(n)-T_{end}(n-1)} \quad \alpha\frac{P_{start}(n+1)-P_{mid}(n)}{T_{start}(n+1)-T_{mid}(n)} \quad \alpha\frac{P_{mid}(n+1)-P_{end}(n)}{T_{mid}(n+1)-T_{end}(n)}\Bigg], \tag{2}$$

where α is a scaling factor, defining importance of the dynamic features compared to the static features. These extended feature vectors are used both for the CART tree build and for the maximum-likelihood intonation generation, depicted in the next sub-section.

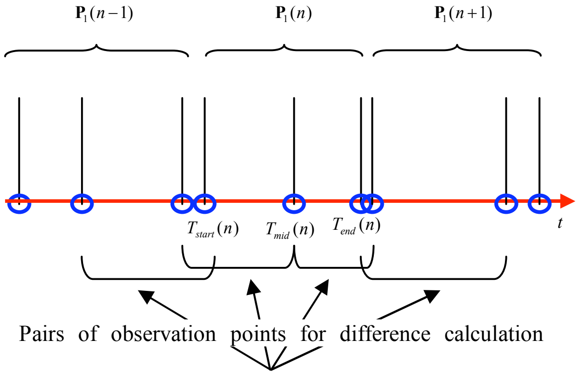


Figure 1. Dynamic feature calculation for n-th syllable

2.3. Maximum likelihood intonation model

A powerful data-driven HMM based synthesis, proposed by Tokuda *et al* [12]-[14] provides a convenient framework for combining both instantaneous and differential observations in order to obtain the most-likely smooth parameter contour, for a given clustering. HMM-clustered observations (e.g. pitch[1], energy, duration and spectral features) are modeled by a Gaussian Mixture Model (GMM) distribution. In such a system a smooth parameter contour, containing static features, is generated from the cluster statistic models by maximizing the likelihood criterion while considering both the static and the dynamic features of speech. When GMM consists of a single Gaussian, an analytic solution also exists [13].

In the current work we apply this ML synthesis, which was originally applied on densely and uniformly spaced feature

[1] Actually, the pitch data in [12] is modeled by a multi-space distribution, to support a binary voiced/unvoiced decision, but it may be also modeled by a simple GMM distribution, when using interpolated pitch for unvoiced portions of speech (which is in particular the case in our system) [3].

data, to the sparse pitch features given in (2). We present here an analytic solution, assuming single-Gaussian distribution for each cluster, although it is possible to apply EM-based iterative solution for multi-Gaussian mixture, formulated in [13].

Let ***O*** be a column vector of concatenated observations for a whole utterance (both static and dynamic observations):

$$\mathbf{O} = \left[\dots \quad \mathbf{P}_2(n-1)^T \quad \mathbf{P}_2(n)^T \quad \mathbf{P}_2(n+1)^T \quad \dots\right] \tag{3}$$

and

$$\mathbf{C} = \left[\dots \quad \mathbf{P}_1(n-1)^T \quad \mathbf{P}_1(n)^T \quad \mathbf{P}_1(n+1)^T \quad \dots\right] \tag{4}$$

is the matching static observation concatenation, where $\mathbf{P}_1(n)$ and $\mathbf{P}_2(n)$ are defined in (1) and (2) accordingly.

Assuming a cluster sequence **Q** is predetermined and each cluster is modeled by a single 7-dimensional Gaussian ($\mathbf{P}_2(n) \sim \mathrm{N}(\boldsymbol{\mu}_n, \mathbf{U}_n)$), the log-likelihood of **O** sequence is given by:

$$\log P(\mathbf{O} \mid \mathbf{Q}) = -\frac{1}{2}\mathbf{O}^T \mathbf{U}^{-1}\mathbf{O} + \mathbf{O}^T \mathbf{U}^{-1}\mathbf{M} + K, \tag{5}$$

where

$$\begin{aligned} \mathbf{U}^{-1} &= \mathrm{diag}[\mathbf{U}_1^{-1}, \dots, \mathbf{U}_n^{-1}, \dots, \mathbf{U}_N^{-1}] \\ \mathbf{M} &= [\boldsymbol{\mu}_1^T, \dots, \boldsymbol{\mu}_n^T, \dots, \boldsymbol{\mu}_N^T]^T. \end{aligned} \tag{6}$$

In addition,

$$\mathbf{O} = \mathbf{WC}, \tag{7}$$

where **W** is a sparse (and block-diagonal) transformation matrix, derived from (1) and (2).

Substituting (7) to (5) and maximizing the log-likelihood (5) with respect to **C**, we obtain the following equation:

$$\mathbf{W}^T \mathbf{U}^{-1} \mathbf{WC} = \mathbf{W}^T \mathbf{U}^{-1} \mathbf{M}^T \tag{8}$$

The solution of (8) may be found using efficient algorithms in a time-recursive manner [14]. This solution jointly determines the pitch curve for the full utterance. The actual length of the utterance *N* depends on allowable delay of the system.

The proposed algorithm creates intonation curve which depends both on individual CART cluster distributions and on their sequence in the synthesized sentence. One may notice (See Figure 2) that the ML dynamic solution acts as a smoother applied to a simple conventional mean solution. The smoothing is optimized and dependent on dynamic parameter distributions inside $\mathbf{P}_2(n)$. It can also be controlled by the scaling factor α.

It is possible (and desirable) to use larger intonation modeling CARTs with the proposed algorithm, with less concern of noisy, abrupt changes in the target pitch curve. Preliminary experiments have shown an improved prosody generated by this maximum likelihood dynamic solution, especially for composite and interrogatory sentences.

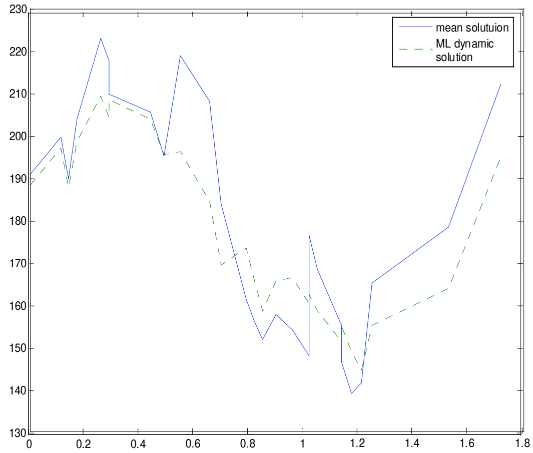


Figure 2: Pitch intonation curve: maximum-likelihood dynamic solution vs. static mean solution

2.4. Microprosody preservation technique

The intonation curve model, described above is sparse in time, so it obscures subtle pitch changes, which are essential for naturally sounding speech. Also, if longer-then-syllable contiguous speech portions are available, it is desirable to keep the original fine pitch curve structure inside that contiguous section for better naturalness, in a way that the original pitch curve is aligned with the CART model rough prosody. Here we refer to that intra-syllable fine pitch structure as *microprosody*. To improve the naturalness of the synthesized speech, a segment-pitch-target-pitch combination method, named *microprosody preservation* technique is proposed below.

As was mentioned in Section 1, many systems use prosody model during segment selection stage only. Once the segments are selected, the smoothed original segment pitch is used for synthesis. This direct usage of *segment intonation* [2][6][7][9] for pitch curve generation has a couple of disadvantages. The quality of output intonation is heavily dependent on the database size and the correspondence between the database and the synthesized text domains. Moreover, the spectral transition cost and other factors which comprise an overall segment selection cost may directly influence the output signal intonation. This dependence may result sometimes in an inconsistent treatment of intonation, where the final pitch curve does not fit the syntactic contents of the synthesized text. The lack of control over prosody in concatenative synthesis is especially harmful when dealing with specific intonation patterns used to express enumerations, emphasis, or questions. These cues are very common in human speech and often crucial to proper information delivery.

Hence, we cannot completely rely on the *segment* intonation solely and need to combine it with the target pitch curve. This combination should also compensate for the imperfectness of the CART model and the feature extraction during the synthesis.

The proposed technique combines pitch fluctuations, extracted from the selected segments, with the modeled pitch intonation curve. In this technique, which is applied only for long enough contiguous portion speech, we first remove the linear trend from the original pitch, and then add the target pitch piecewise-linear (3 points per syllable) function instead.

Let $L_1(t) = at + b$ be a linear regression for a contiguous portion of speech, calculated from start and end points of selected segments inside this speech portion, and $L_2(t)$ be a piecewise linear target pitch curve. The combined pitch for contiguous speech portion $P_2(t)$ is obtained from the original segment pitch, denoted $P_1(t)$, and target pitch curve $L_2(t)$, by:

$$P_2(t) = P_1(t) - L_1(t) + L_2(t) \qquad (9)$$

Appropriate factorization is carried out in order to confine the maximal pitch change (between the target pitch and the microprosody pitch) ratio and to gradually incorporate the microprosody of long contiguous sections with the unchanged target pitch at other areas.

3. Model implementation on IBM embedded CTTS system

3.1. System description

The IBM Trainable Speech Synthesis System is a decision tree based, unit selection, waveform concatenation speech synthesis system. It is composed of three major components: a front-end which does text normalization and pronunciation generation, a prosody module which generates pitch, duration, and energy targets, and a back-end module which finds the best segment from a large set of segments related to the relevant phonetic context, which minimize a cost function. The selected segments are then concatenated and signal processing is performed on the resulting synthetic speech. The segments are either stored as (uncompressed) waveforms for large scale systems [3] or as their parametric representations for embedded systems [5]. The basic speech units for concatenation are HMM states, usually three per phone. The HMMs are trained automatically on the speech corpus in advance. In order to determine the segment sequence to concatenate, a dynamic programming search is performed over all segments aligned to each leaf of the decision-trees in synthesis. A large discontinuity penalty was added to the overall cost to favor longer contiguous speech selections [3]. We tested the embedded modification of the system [5] on a large scale (15 hours of recorded sentences) American English female voice.

3.2. Application of modified prosody model

As a part of the voice building process an extended intonation CART model was built, based on 7-dimentional observation vectors, as defined in equation (2).

During the synthesis process, durations and energies were extracted as usual [3], while the maximum likelihood dynamic solution was used for intonation modeling from the new CART. The dynamic elements in (1) were constructed, based on the target pitch and durations. The scaling factor α in (1) was chosen to be equal to one for time given in seconds.

The target intonation curve, obtained by the ML dynamic model, described above, was used as a part of the target cost for the segment selection. After the unit selection, it was combined with *segment pitch* fluctuations by the microprosody preservation technique, described in Section 2.4. The microprosody was applied, based on segment prosody information, stored in the database, i.e. starting and ending pitch for each selection unit.

3.3. Subjective testing

A set of subjective evaluations has been conducted to assess the perceptual quality, obtained by application of the above algorithm on the IBM Trainable Concatenative TTS system (embedded version). All subjects were blind as to the identity of the synthesis system associated with each utterance. After listening to the two versions of an utterance, they were instructed to choose between 5 options: no preference, strong preference to either side or weak preference to either side. Participants were told that they may listen to the utterances in any order, and as many times as they liked. Long composite utterances were split for separate voting.

In the first evaluation, eight TTS experts evaluated the intonation of the speech, synthesized using the dynamic ML solution (model A), compared to a simple static CART pitch, slightly smoothed to prevent abrupt changes (model B). The results are presented in Table 1. The test results show statistically significant (p < 0.05) preference of the proposed system over the static mean solution. It should be clarified, that this test compared only target pitch curves without any combination with the original pitch, which is supposed to improve the baseline quality of TTS systems.

Table 1: A-B preference test for Dynamic ML solution

Pref.	No pref.	Static, smoothed (A)		Dynamic ML (B)	
		All pref.	Strong pref.	All pref.	Strong pref.
%	37.9	34.3	3.2	43.2	9.2

In the second evaluation, both systems were tested, where also the original database pitch was used for the final pitch curve generation. The baseline system (A), with static CART mean solution as a target pitch, used a smoothed original pitch curve extracted from the selected segments. The proposed system (B), having dynamic ML CART solution as a target pitch, used the microprosody preservation technique. Seven TTS experts and four Native American English speakers, unfamiliar with the TTS system, evaluated the two TTS systems. The results (see Table 2 and Table 3) showed subjective preference of the proposed constellation over the base system both by TTS experts and non-professional native speakers. The expert test results were found to be statistically significant (p < 0.05). The native speakers' results were less categorical, but the combined expert and native test results were also found statistically significant (p < 0.05).

Table 2: A-B preference test for full TTS system (experts)

Pref.	No pref.	Static, smoothed (A)		Dynamic ML + microprosody (B)	
		Strong or weak pref.	Strong pref.	Strong or weak pref.	Strong pref.
%	39	26.9	0.4	34.1	3.8

Table 3: A-B preference test for full TTS system (native speakers)

Pref.	No pref.	Static, smoothed (A)		Dynamic ML + microprosody (B)	
		Strong or weak pref.	Strong pref.	Strong or weak pref.	Strong pref.
%	36.6	29.1	8.2	34.3	11.9

3.4. Discussion and future research directions

It should be mentioned that the synthesized intonation of the base system depends only in an oblique manner on the target pitch curve. The latter serves as recommendation (a part of the overall additive cost) and is not directly used in the synthesis. Although, for rich speech corpus the quality of such a system is generally high, the results are hardly controllable and heavily dependent on the phonetic context. Hence, some utterances are generated with inadequate intonation. However, the resultant pitch fluctuations create an effect of expressiveness and enthusiasm which are for some utterances subjectively preferred even if exaggerated.

Those phenomena explain the relatively small preference of the proposed system over the base line system (the second test), despite the fact that the target pitch improvement was significant (the first test).

The above findings propose a further research direction for adjustable microprosody incorporation, where the extent to which the target pitch will be incorporated into the final intonation will depend on the correlation between the actual segment intonation and the target intonation.

The proposed *microprosody preservation* technique is applied only on relatively long contiguous speech portions, chosen during the segment-selection. For small footprint TTS systems, having relatively low number of available segments for synthesis, this technique can not be directly applied. Further work should be done to reduce the dependency of the resultant quality on the speech footprint size (for instance, by modeling the *microprosody* as well, rather than directly using the speech corpus for its extraction).

4. Summary

The Maximum-likelihood dynamic intonation model was proposed in the current work. This model provides optimal solution for target intonation curve (3 points per syllable) and incorporates both static and dynamic pitch values for better utterance intonation modeling. The target pitch curve is incorporated both in the segment selection stage and in the final intonation computation.

The final intonation is computed by the microprosody preservation algorithm in order to reduce the pitch modification ratio and improve the sound naturalness while remaining consistent with the target pitch curve.

The proposed model was applied on IBM's trainable concatenative TTS system to produce improved subjective intonation quality.

5. References

[1] Monaghan, A., "State-of-the-Art Summary of European Synthetic Prosody R&D", in "*Improvements in Speech Synthesis*", edited by E.Keller *et al*, UK, Wiley & Sons, 2002

[2] Eide, E., *et al.*, "Recent Improvements to the IBM Trainable Speech Synthesis System", *Proc. ICASSP 2003*, Hong Kong, Vol. 1, pp. 708-711.

[3] Donovan, R.E., *et al*. "Current Status of the IBM Trainable Speech Synthesis System", *Proc. 4th ISCA Tutorial and Research Workshop on Speech Synthesis*, Scotland, UK, 2001

[4] Strom, V., "From text to prosody without ToBI", *Proc. ICSLP 2002*, pp 2081-2084.

[5] Chazan, D., Hoory, R., Kons, Z., Sagi, A., Shechtman, S., Sorin, A. "Small footprint concatenative text-to-speech synthesis system using complex spectral envelope modeling", *Proc. INTERSPEECH-2005*, pp 2569-2572.

[6] X.J. Ma, W. Zhang, W. B. Zhu, Q. Shi and L. Jin, "Probability Based Prosody Model For Unit Selection", *Proc. ICASSP*, Montreal, 2004.

[7] A. Raux and A. Black, "A Unit Selection Approach to F0 Modeling and its Application to Emphasis", *ASRU 2003*, St Thomas, US Virgin Islands.

[8] Hunt, A. and A. Black, A. W., "Unit selection in a concatenative speech synthesis system using a large speech database", in *ICASSP '96*, Philadelphia, PA, 1996, pp. 373–376.

[9] Colotte, V., Beaufort, R., "Linguistic features weighting for a Text-To-Speech system without prosody model", *Proceedings of the 9th European Conference on Speech Communication and Technology*, Lisbon, Portugal, September 2005.

[10] Hamza, W., Eide, E., Bakis, R., "Reconciling Pronunciation Differences between the Front-End and the Back-End in the IBM Speech Synthesis System", *Proc. ICSLP 2004*, Korea.

[11] Moberg, M., Parssinen, K., "Comparing CART and Fujisaki intonation models for synthesis of US-English names", In *SP-2004*, 439-442.

[12] Tokuda, K., Zen, H., Black, A. W., "An HMM-based speech synthesis system applied to English", *2002 IEEE Speech Synthesis Workshop*, Santa Monica, California, Sep. 11-13, 2002.

[13] K. Tokuda, T. Masuko, T. Kobayashi, T. Kitamura, "Speech Parameter Generation Algorithms for HMM-

Based Speech Synthesis", Proc. 2000 *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2000*, pp.III-1315-1318, Istanbul, Turkey

[14] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Quantization of Vector Sequences Using Statistics of Neighboring Input Vectors", *J. Acoust. Soc. America*, 100, 4, Pt.2, pp.2762-2763 (1996), Proc. ASA and ASJ 3rd Joint Meeting, Honolulu, USA, pp.1067-1072

Data-driven Extraction of Intonation Contour Classes

Uwe D. Reichel

Institute of Phonetics and Speech Processing
University of Munich, Germany
reichelu@phonetik.uni-muenchen.de

Abstract

In this paper we introduce the first steps towards a new data-driven method for extraction of intonation events that does not require any prerequisite prosodic labelling. Provided with data segmented on the syllable constituent level it derives local and global contour classes by stylisation and subsequent clustering of the stylisation parameter vectors. Local contour classes correspond to pitch movements connected to one or several syllables and determine the local f0 shape. Global classes are connected to intonation phrases and determine the f0 register. Local classes initially are derived for syllabic segments, which are then concatenated incrementally by means of statistical language modelling of co-occurrence patterns.

Due to its generality the method is in principal language independent and potentially capable to deal also with other aspects of prosody than intonation.

1. Introduction

The prosody module of a speech synthesis system has to relate text or concepts to prosody in order to predict the latter from the former. To facilitate this mapping some representation of prosody is needed. Since this paper deals with the intonational aspect of prosody, some common description approaches for intonation are shortly listed here. They can roughly be divided into symbolic, parametric and perception-based approaches.

1.1. Symbolic Approaches

In the Tone Sequence Approach [19], which is grounded on auto-segmental phonology [22], intonation is seen as a succession of tones that are associated to accentuated or phrase final syllables. The tone inventory consists of two elementary tones (High and Low) that can be combined to complex tones. Possible tone sequences are controlled by an intonation grammar. There are rule-based [20] statistic approaches [21] for the generation of the concrete f0 values from this abstract representation of intonation.

The Kiel Intonation Model (KIM) [6] treats prosodic categories as bundles of distinctive features. It contains rules for mapping manual annotations to prosodic categories, and for mapping those categories to numeric f0 values. One emphasis lies on examining the synchronisation of syllable nuclei and f0 peaks (so called *early, middle and late peak*).

1.2. Parametric Approaches

The Fujisaki model ([8], [10], [11]) predicts intonation contours by a superposition of a baseline f0, a phrase component for global contours (intonation phrases), and an accent component for local contours (accented syllables). One possibility to estimate this model's parameter values is analysis by synthesis [11], i.e. analysing the given f0 contour by synthesis via the Fujisaki model.

Models like Tilt [12] and PaintE [7] try to approximate the f0 contour on accentuated syllables by stylisation functions. In PaintE furthermore the parameter vectors of the stylisation function are clustered in order to get categorised intonation building blocks.

1.3. Perception-based Approaches

The IPO model ([24], [23]) operates on a perceptually equivalent approximation of given f0 contours by a sequence of straight lines (the so called *copy contour*). Thus this stylisation is carried out interactively with subjects judging the approximation perceptually. The resulting lines of different slope form intonation units who's succession can be described by an intonation grammar.

1.4. Shortcomings of the Given Approaches

There are some shortcomings of the approaches described above:

- Leaving aside IPO, all models mentioned above rely on accent and phrase boundary labels of various complexity. Therefore at least initially hand-labelling of the data is necessary. This work is time consuming and needs trained experts. Especially in prosody inter-labeller agreement and intra-labeller consistency run the risk of getting relatively low [4] which leads to a loss of prosodic training data. Presumably this problem grows with the increasing size of the label inventory.

- The label inventories are not necessarily language independent. Inventories like ToBI for example need to get adjusted whenever they are applied to new languages [5]. Also the IPO model needs perceptual readjustment for each new language.

With our model we try to avoid these shortcomings. Since our approach is purely data driven, no manual prosodic labelling or manual adjustment to other languages is needed.

2. Data

Our training data consists of parts of the IMS Radio News Corpus [1] with a total length of about 14 minutes. The corpus part used in this study contains news texts read by one professional male speaker. It is segmented amongst others on the phone and syllable level. For f0 measurement we utilised autocorrelation implemented in *Praat* (version 4.1.5) software with a sampling rate of 100 Hz.

3. Extraction of Local Contour Classes

As in the Fujisaki model, our model distinguishes between local and global contours. Local contour classes correspond to pitch shapes connected to one or more syllables. They are derived by parameter clustering of stylisation polynomials. Starting with syllables, contour segments are iteratively merged to larger units. Figure 1 gives an overview over the processing steps which are described in greater detail in the following sections.

```
segments := syllables
iterate
    foreach $s \in$ segments
        - preprocessing: interpolation, smoothing, and time
          normalisation of f0 contour of $s$ in context of the pre-
          ceeding and following syllable.
        - adaptive stylisation of the contour by polynomials
    end
    - cluster polynomial coefficients to derive contour classes
    - segments := merge neighbouring segments if respective
      classes occur in dependence of each other
    - terminate if no merging possible
end
```

Figure 1: *Algorithm for incremental local intonation contour class extraction*

3.1. Contour Preprocessing

Preprocessing as shown in Figures 2 and 3 removes contour characteristics not related to intonation, among them microprosody, intrinsic pitch, speech rate, and syllable constituency. For each contour segment preprocessing took place in the context of the preceeding and the following syllable. Hertz values were transformed to the logarithmic semitone scale.

3.1.1. Smoothing

To eliminate f0 movements related to intrinsic pitch, coarticulation effects at voice on- and offsets and f0 measurement errors the contours were smoothed using a Savitzky-Golay filter [18] of order 3 and length 5. This filter is commonly applied for this purpose (see e.g. [17]) due to its capability to remove high frequency noise from pertinent information.

3.1.2. Time Normalisation

In order to exclude any influence of speech rate, phone number, and syllable constituent structure, all syllables were time normalised in the following way: the syllable head is mapped on the interval -0.4 to -0.2, the nucleus from -0.2 to 0.2, and the coda from 0.2 to 0.4. Missing heads or codas are padded by interpolation between the f0 values of the nucleus and the neighbouring syllables.

3.1.3. Interpolation

Since the subsequent stylisation step requires continuous contours, plosive closure phases and missing syllable constituents are bridged by cubic splines.

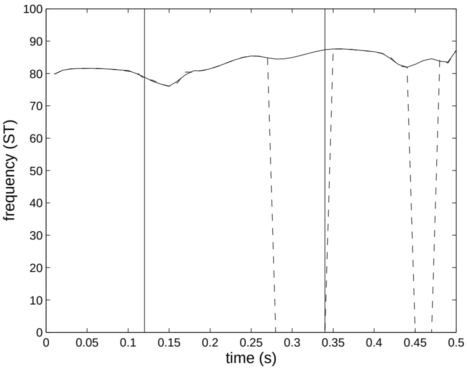


Figure 2: *Preprocessed f0 contour (solid line): spline interpolation, smoothing by Savitzky-Golay filter. Dashed line: original contour. The two vertical lines mark the boundaries of the contour segment and the preceeding and following syllable, respectively.*

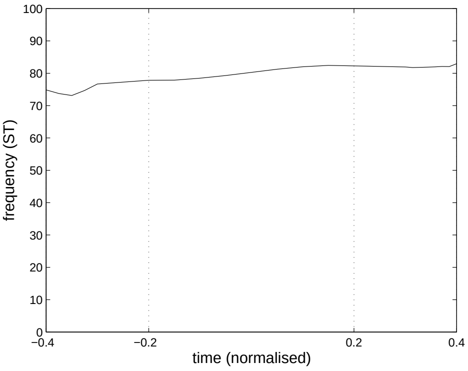


Figure 3: *Time normalisation of the f0 contour. The vertical lines separate syllable onset, nucleus and coda.*

3.2. Adaptive Stylisation

A polynomial stylisation was carried out, guided by the multidimensional unconstrained nonlinear Nelder-Mead minimisation [15] of the squared error between original and stylised contour.

The higher the polynomial order, the closer the fit to the original contour, but also the more unreliable the subsequent clustering of the coefficient vectors. Therefore for each contour stylisation the lowest possible polynomial order was chosen ranging from zeroth to third order (cf. Figure 4). The goodness of fit was determined by the maximum distance between corresponding values of the original and the stylised contour. If this distance did not exceed a certain value, the stylisation was judged to be sufficiently close. The threshold was set to 4 Hz with reference to Klatt [16] who reported a just noticeable difference of 2 to 5 Hz for non-stationary stimuli.

To make sure that all coefficient vectors had the same length for subsequent clustering, zeros were padded to the vectors of the polynomials of lower order than 3.

3.3. Clustering

As in the PaintE model mentioned in the introduction, intonation contour classes were derived by Kmeans clustering of the coefficient vectors of the stylisation polynomials. Since only the shape and not the frequency offset characterises a contour class, the first coefficient was ignored.

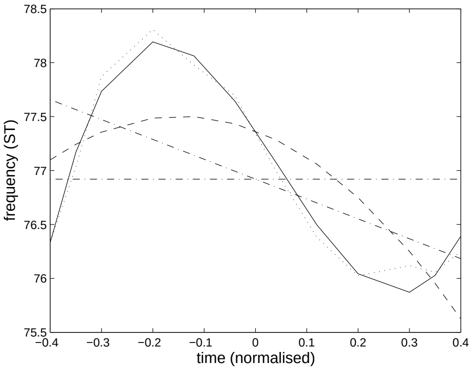


Figure 4: *Adaptive stylisation using polynomials of increasing order until maximum distance criterion is met. Dotted line: contour to be stylised, remaining lines: polynomial stylisation of increasing order from 0 to 3.*

Here the determination of the optimal number of clusters was guided by the Dunn index, a validity measure of hard clustering taking into account cluster compactness and separation between clusters. After having carried out Kmeans clustering 10 times for each given specification of number of clusters, the number connected to the highest mean Dunn score was chosen.

The centroid vectors served as cluster representatives.

4. Merging of Contour Segments

Contour segments are merged if the respective contour classes co-occur non-randomly. To determine whether the co-occurrence is random or not, the Log-Likelihood Ratio is utilised, a method used in the field statistical natural language processing for example to retrieve collocations [13].

This method compares the likelihoods L of the observed occurrences of the intonation classes c_i and c_j given two different hypotheses:

$$\begin{aligned} \text{H0:} &\quad P(c_i|c_j) = p = P(c_i|\neg c_j) \\ \text{H1:} &\quad P(c_i|c_j) = p_1 \neq p_2 = P(c_i|\neg c_j) \end{aligned}$$

According to $H0$, c_i and c_j occur independently (the probability p of c_i does not change in dependence of preceeding c_j), whereas $H1$ claims dependence. Under the simplifying assumption that the probabilities for the observed occurrence pattern for c_i and c_j can be described by a binomial distribution, the likelihoods for the observed data according to $H0$ and $H1$ are given as follows:

$$\begin{aligned} L(H0) &= b(n_{ij}; n_j, p) b(n_i - n_{ij}; N - n_j, p) \\ L(H1) &= b(n_{ij}; n_j, p_1) b(n_i - n_{ij}; N - n_j, p_2), \end{aligned}$$

where n_i and n_j are the observed frequencies of classes c_i and c_j, respectively, n_{ij} stands for the observed frequency of the sequence $c_i c_j$, and N is the total number of observations. The probability $b(n_{ij}; n_j, p)$ following a binomial distribution then represents the expectation of observing the sequence $c_i c_j$ n_{ij} times in n_j trials, if the probability of observing c_i given c_j is p.

A comparison of the log likelihoods leads to the Log-Likelihood Ratio $\ln \lambda$:

$$\begin{aligned} \ln \lambda &= \ln \frac{L(H0)}{L(H1)} \\ &= \ln L(n_{ij}, n_j, p) + \ln L(n_i - n_{ij}, N - n_j, p) \\ &\quad - \ln L(n_{ij}, n_j, p_1) - \ln L(n_i - n_{ij}, N - n_j, p_2) \end{aligned}$$

$-2 \ln \lambda$ follows approximately a χ^2 distribution, so a χ^2 test can be applied to decide whether the independence hypothesis $H0$ can be rejected in favour of $H1$. If the dependence hypothesis turned out to be significantly more appropriate (this study's significance level was set to 0.01), the corresponding segments were merged.

The next iteration step's preprocessing, stylisation and clustering then operated on the resegmented data. In case of impossibility of further merging the procedure terminated.

5. Extraction of Global Contour Classes

As explained above, local intonation contour classes were derived independently of registers. In order to model the f0 register of each syllable, global contour classes were extracted by stylisation and clustering of f0 baselines in intonation phrases which had been segmented automatically.

5.1. Segmentation of Intonation Phrases

The baseline f0 values served as a representation of registers. For each syllable such a baseline value was calculated by taking the mean of the n lowest f0 values measured within the syllable (n was set to 8 in this study). The mean was taken to reduce the effect of potential pitch measurement errors.

As shown in Figure 5 we then simply treated each speech pause and each baseline pitch discontinuity as a phrase boundary. The discontinuity threshold was set to 3 semitones. This is only a first approximation, since also prominent pitch accents and boundary tones show that large pitch differences compared to the neighbouring syllables.

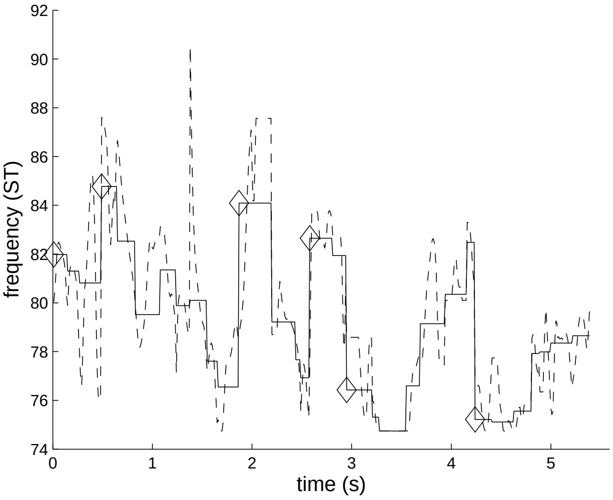


Figure 5: *Dividing the utterance into intonational phrases at baseline pitch discontinuities and speech pauses. Dashed line: original contour, solid line: pitch baseline. Intonation phrase starting points are marked by diamonds.*

5.2. Stylisation and Clustering

The time of each intonation phrase was normalised to the interval $[0\ 1]$ in order to remove any phrase length effects. The sequence of baseline f0 values of the contained syllables was stylised by straight lines, and the slope parameters were clustered by the same procedure as described in section 3.3. This led to a limited number of discrete global contour classes. As with local contour classes the centroids were taken as cluster representatives.

6. Resynthesis

In resynthesis the original f0 values were replaced by contours derived from the respective local and global contour classes. The local class determined the shape, the global class, together with the position of the segment in the intonation phrase, determined the register. An illustrative example is given in Figure 6. There the f0 contour of the one-syllable segment belongs to the local intonation class l_3 whos representative is the centroid parameter vector $p_{l3} = [16.0663, -16.1095, -88.2260]$ and to the global intonation class g_3 represented by the centroid shape parameter $p_{g3} = 6.7543$ (cf. Figures 7 and 8, respectively). The local contour $\mathrm{f0}_l$ of the time normalised segment (see section 3.1.2) is given by:

$$\mathrm{f0}_l(t) \quad = \quad 16.0663t - 16.1095t^2 - 88.2260t^3,$$

t stands for (normalised) time. The syllable dependent register $\mathrm{f0}_r(\sigma_n)$ is derived from the slope of the global contour line associated with class g_3 and the relative position $p(\sigma_n)$ of the n-th syllable σ_n within the intonation phrase. The starting point of the straight line $\mathrm{f0}_r(\sigma_1)$ is set to the original intonation phrase‘s initial baseline value. Future research is needed to predict this value, which reflects the amount of pitch reset at phrase boundaries. The registers for all syllables σ_n are then calculated the following way:

$$\mathrm{f0}_r(\sigma_n) \quad = \quad \mathrm{f0}_r(\sigma_1) + 6.7543p(\sigma_n)$$

In our example $\mathrm{f0}_r(\sigma_1)$ is 80 semitones (ST), and $p(\sigma_n)$ is 0.8 (e.g. $n = 8$ in a 10-syllable phrase). For each syllable σ_n involved in the contour segment the baseline $b_l(\sigma_n)$ of the corresponding part of the local contour $\mathrm{f0}_l$ is then replaced by the syllable’s register $\mathrm{f0}_r(\sigma_n)$ so that the actual contour f0 is calculated by:

$$\mathrm{f0}(t) \quad = \quad \mathrm{f0}_l(t) - b_l(\sigma_n) + \mathrm{f0}_r(\sigma_n)$$

Finally the resulting contour is aligned to the given time range and syllable structure of the segment.

To enhance naturalness of the resulting signals we added jitter in form of a quasi-random component Δf_0 as a sum of three sine waves according to a formula proposed by Klatt and Klatt [25]:

$$\Delta\mathrm{f0}(t) \quad = \quad \frac{fl}{50} \cdot \frac{\mathrm{f0}}{100}\Big[\sin(2\pi 12.7t) + sin(2\pi 7.1t) + sin(2\pi 4.7t)\Big]\mathrm{Hz}$$

The fluttering parameter fl was set to 25. Time t is given in seconds.

7. Perceptual Evaluation

In order to test the perceptual appropriateness of our model we conducted two perception experiments, one for naturalness judgements, and the second to test functional equivalence of original and modelled contours. The stimuli were created by MBROLA (version 3.01h) resynthesis [14] replacing the original f0 contour by a sequence of contour classes as described in section 6.

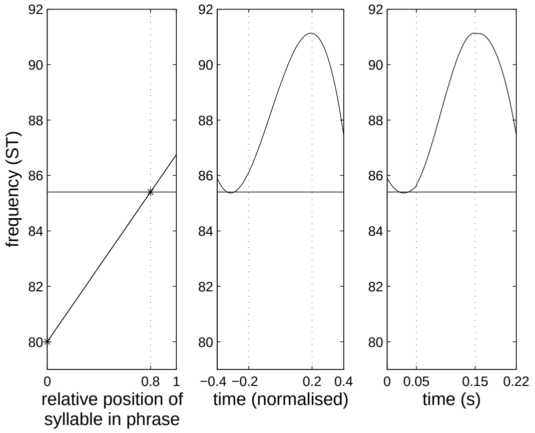


Figure 6: *Combination of global and local contour.* **Left:** *The segment’s register baseline is predicted by original frequency offset (here: 80 ST), global contour associated to corresponding global contour class (here: class g_3, cf. Figure 8), and relative position of the segment within the intonational phrase (here: 0.8).* **Middle:** *The baseline value (cf. section 5.1) of the local contour given here by local contour class l_3 (cf. Figure 7) is shifted to this value.* **Right:** *The contour is aligned to the original time ranges of syllable onset (0s–0.05s), nucleus (0.05s–0.15s) and coda (0.15s–0.22s).*

6 subjects, 2 male and 4 female, took part in the experiments, their age ranged from 24 to 50. All except one were trained phoneticians, and all except one were German native speakers (the non-native speaker has lived in Germany for more than 15 years, and her pronunciation showed no foreign language accent).

7.1. Naturalness

In the first experiment the subjects were instructed to judge the naturalness of 50 inter-pausal speech segments that comprised at least 3 syllables. Each segment was presented with original and modelled f0 resulting in 100 stimuli that were randomly ordered. The judgement scale contained 4 values: *completely natural, tolerably natural*, *rather unnatural*, and *completely unnatural*. Since all stimuli were created using MBROLA, none of the stimulus groups was penalised compared to the other concerning synthesis artefacts. The stimuli were faded in and out by superimposing a Tukey window (taper sections each set to 3% of the stimulus length).

The participants could listen to each stimulus as often as they wanted to and could revise their judgements at any time.

Table 1 shows the mean judgements for original and modelled f0.

7.2. Functional Equivalence

In the second experiment the subjects had to decide for stimulus pairs whether their intonation contours were functionally equivalent or not. The same 50 stimuli as in the first experiment were used and presented pairwise in random order. In half of the stimuli pairs both stimuli contained either the original or the modelled f0 (‘same contour’ case). In the other half one stimulus contained the original f0, and the other the modelled one (‘different contour’ case), original and model presented in random order.

Functional equivalence concerned weighting of information (position and prominence of accents), discourse embedding of the segment (progredient vs. final intonation contour) and, if applicable, sentence mode.

As in the first experiment, each stimulus could be listened to arbitrarily often, and judgements could be revised at any time.

8. Results

8.1. Resulting Contour Classes

The application of our method to the given data yielded 9 local intonation contour classes (see Figure 7) differing in shape and number of involved syllables (from 1 to 3), and 6 global contour classes differing in slope of the declination and inclination baselines (see Figure 8).

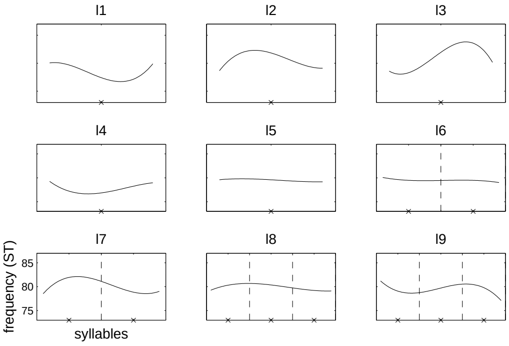


Figure 7: *Local contour classes. All contours are shifted to the mean of 80 ST. Time is normalised as described in section 3.1.2. Syllable boundaries are marked by vertical dashed lines, nucleus centers by crosses.*

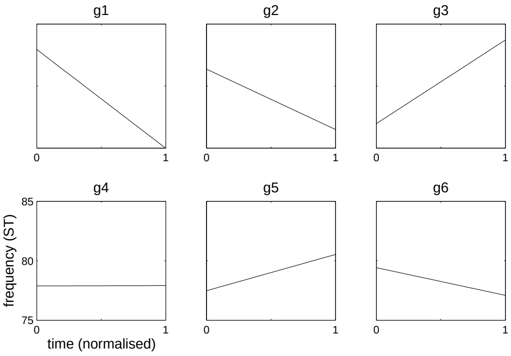


Figure 8: *Global contour classes (declination baselines). Time is normalised to the interval* [0 1].

8.2. Numerical and Perceptual Evaluation

The root mean square error between all original and generated f0 values amounted 10.26 Hz.

The results of naturalness and functional equivalence judgements are shown in Tables 1 and 2, respectively. The original f0 contours were judged highly significantly as more natural then the modelled contours (two-sided Wilcoxon matched pairs signed rank test, $alpha = 0.001$).

Table 1: *Mean subject judgements for the naturalness of original and modelled f0 contours.*

	mean	maximum	minimum
original	3.14	4	1
model	2.61	4	1

Concerning functional equivalence, Table 2 reveals that about 27% of the ‘different contour’ stimulus pairs were also judged as different. Figure 9 gives the numbers of ‘functionally not equivalent’ judgements for each of the ‘different contour’ stimulus pair types.[1]

Table 2: *Contingency table for ’functionally equivalent/not equivalent‘ judgements of stimulus pairs with same and different contours. Cramer‘s V = 0.38.*

	same contours	different contours
equivalent	149	109
not equivalent	1	41

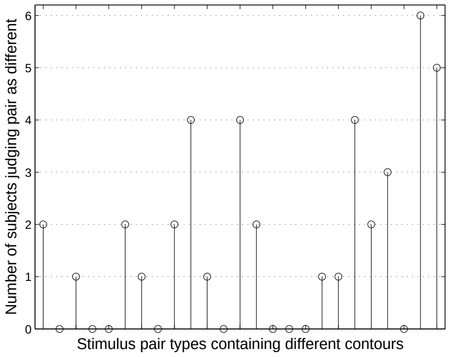


Figure 9: *Number of subjects with ‘functionally not equivalent’ judgements for each ‘different contour’ stimulus pair type.*

9. Discussion

9.1. Evaluation Results

The results of the naturalness experiment presented in the previous section clearly show that our model in its current state is not capable to produce contours that reach the quality of original intonation. This is not surprising since purely data driven models lack expert knowledge included into the models listed in the introduction, for example knowledge about perceptual equivalence (IPO) or position and types of accents and phrase boundaries. It is unclear whether the mean naturalness judgements for our model would rise, if the subjects would compare them not only to the original contours but also to a model worse than ours. Such a triple comparison had been carried out e.g. by Möhler [2] who additionally had presented flat intonation contours. Furthermore, as with all data driven models more training data is likely to enhance performance, so far we use just 14 minutes of speech.

Concerning functional equivalence, the results are already a bit more promising. As can be seen in Figure 9, 24% (6 out

[1] By stimulus pair *types* we mean the set of distinct stimulus pairs.

of 25) of the ‘different contour’ stimulus pair types were judged as functionally not equivalent by half of the subjects or more, indicating that the majority of the subjects was not able to functionally distinguish the other 76%.

9.2. Local Contours

Some of the extracted local contour classes can be related to other intonation description systems. Thus, classes $l2$ and $l3$ correspond to the events *early* and *late* peak [6] representing the alignment of f0 peaks and syllable nuclei.

9.3. Global Contours

The extracted global contours represent declination and inclination lines of different slopes. There are still open questions concerning the modelling of the global contours. First, a segmentation of a contour into intonational phrases guided by pitch discontinuities is not completely adequate since also boundary tones and prominent pitch accents correlate with such discontinuities. Here the syllable length between successive pitch discontinuities could help to distinguish between such prosodic events and real phrase boundaries, since the domain of pitch accents and boundary tones is in general limited to one syllable. The second open question concerns the amount of pitch reset. In this study we use the original phrase initial baseline values as starting points for the declination line. One potential approach is the prediction of pitch reset by a linear combination of factors like the final frequency of the preceeding intonation phrase, the durations of the preceeding and the current phrase, and their f0 slopes. A similar procedure was utilised to predict pause durations at prosodic boundaries in [3].

9.4. Generality of the Model

In this study we excluded other time-related prosodic aspects than intonation by time normalisation.

However, due to its data drivenness and generality our model is not just language independent but also principally capable to deal with other aspects of prosody than intonation. It would for example be of interest how it performs in modelling perceived local speech rate contours [9].

9.5. Relation to Linguistic Units

Another issue for future research is the question of linguistic significance of the extracted contour classes. They are only relevant for speech synthesis, if they can be related to linguistic dimensions like information and discourse structure. It is not yet known, whether the contour classes could be predicted from text.

10. References

[1] S. Rapp, “Automatisierte Erstellung von Korpora für die Prosodieforschung,” Ph.D. dissertation, University of Stuttgart, Institute of Natural Language Processing, Stuttgart, 1998.

[2] G. Möhler, “Theoriebasierte Modellierung der deutschen Intonation für die Sprachsynthese,” Ph.D. dissertation, Institut für Maschinelle Sprachverarbeitung, Stuttgart, 1998.

[3] H. Pfitzinger and U. Reichel, “Text-based and Signal-based Prediction of Break Indices and Pause Durations,” in *Proc. Speech Prosody*, Dresden, 2006, pp. 133–136.

[4] M. Grice, M. Reyelt, R. Benzmüller, J. Mayer, and A. Batliner, “Consistency in Transcription and Labelling of German Intonation with GToBI,” in *Proc. ICSLP*, New Castle, Delaware, 1996, pp. 1716–1719.

[5] M. Reyelt, M. Grice, R. Benzmüller, J. Mayer, and A. Batliner, “Prosodische Etikettierung des Deutschen mit ToBI,” in *Natural Language and Speech Technology, Results of the third KONVENS conference*, D. Gibbon, Ed. Berlin, New York: Mouton de Gruyter, 1996, pp. 144–155.

[6] K. Kohler, “A model of German intonation,” in *AIPUK*, Kiel, 1991, vol. 25.

[7] G. Möhler and A. Conkie, “Parametric modeling of intonation using vector quantization,” in *Proc. 3rd ESCA Workshop on Speech Synthesis*, 1998.

[8] H. Fujisaki, “A note on physiological and physical basis for the phrase and the accent components in the voice fundamental frequency contour,” in *Vocal physiology: voice production, mechanisms, and functions*, O. Fujimura, Ed. New York: Raven, 1987, pp. 165–175.

[9] H. Pfitzinger, “Phonetische Analyse der Sprechgeschwindigkeit,” Ph.D. dissertation, Institute of Phonetics and Speech Processing, 2001.

[10] B. Möbius, *Ein quantitatives Modell der deutschen Intonation: Analyse und Synthese von Grundfrequenzverläufen*. Tübingen: Niemeyer-Verlag, 1993.

[11] H. Mixdorff, “An Integrated Approach to Modeling German Prosody,” in *Studientexte zur Sprachkommunikation*. Dresden: Universitätsverlag, 2002, vol. 25.

[12] P. Taylor, “The rise/fall/connection model of intonation,” *Speech Communication*, vol. 15, pp. 169–186, 1995.

[13] T. Dunning, “Accurate methods for the statistics of surprise and coincidence,” *Computational Linguistics*, vol. 19, pp. 61–74, 1993.

[14] T. Dutoit, V. Pagel, N. Pierret, F. Bataille, and O. Van der Vrecken, “The MBROLA project: Towards a Set of High Quality Speech Synthesizers Free of Use for Non Commercial Purposes,” in *Proc. ICSLP*, vol. 3, Philadelphia, 1996, pp. 1393–1396.

[15] J. Nelder and R. Mead, “A Simplex Method for Function Minimization,” *Computer Journal*, vol. 7, pp. 308–313, 1965.

[16] D. Klatt, “Discrimination of fundamental frequency contours in synthetic speech: implications for models of speech perception,” *Journal of the Acoustical Society of America*, vol. 53, pp. 8–16, 1973.

[17] J. Van Santen, T. Mishra, and E. Klabbers, “Estimating Phrase Curves in the General Superpositional Intonation Model,” in *Proc. 5th ISCA Speech Synthesis Workshop*, Pittsburgh, 2004, pp. 61–66.

[18] A. Savitzky and M. Golay, “Smoothing and differentiation of data by simplified least squares procedures,” *Analytical Chemistry*, vol. 36, pp. 1627–1639, 1964.

[19] J. Pierrehumbert, “The phonology and phonetics of Englisch intonation,” Ph.D. dissertation, MIT, Cambridge, MA, 1980.

[20] ——, “Synthesizing intonation,” *Journal of the Acoustical Society of America*, vol. 70, no. 4, pp. 985–995, 1981.

[21] A. Black and A. Hunt, “Generating F0 contours from ToBI labels using linear regression,” in *Proc. ICSLP*, vol. 3, Philadelphia, 1996, pp. 1385–1388.

[22] J. Goldsmith, “Autosegmental Phonology,” Ph.D. dissertation, MIT, Cambridge, 1976.

[23] L. Adriaens, “Ein Modell deutscher Intonation: eine experimentell-phonetische Untersuchung nach den perzeptiv relevanten Grundfrequenzänderungen in vorgelesenem Text,” Ph.D. dissertation, University of Technology, Eindhoven, 1991.

[24] J. t’Hart, R. Collier, and A. Cohen, *A perceptual study of intonation: An experimental-phonetic approach to speech melody*. Cambridge: Cambridge University Press, 1990.

[25] D. Klatt and L. Klatt, “Analysis, synthesis, and perception of voice quality variations among female and male talkers,” *Journal of the Acoustical Society of America*, vol. 87, no. 2, pp. 820–857, 1990.

Word Accentuation Prediction Using a Neural Net Classifier *

Taniya Mishra, Emily Tucker Prud'hommeaux, Jan van Santen

Center for Spoken Language Understanding
OGI School of Science & Engineering at OHSU
20000 NW Walker Road, Beaverton, OR 97006, USA
{mishra, emtucker, vansanten}@cslu.ogi.edu

Abstract

Automatic prediction of pitch accent assignment is an important but challenging task in text-to-speech synthesis (TTS). Early work in accent prediction relied on simple word-class distinctions, but recently more sophisticated inductive learning models using multiple features have been applied to the problem. For our neural network accent classifier, we developed a corpus that was labeled according to judgments of accent assignment appropriateness in synthesized speech rather than the usual ToBI annotation guidelines. Because the resulting training set was imbalanced, the baseline neural network we developed for this task had a very high accuracy rate (84%) but performed only slightly better than chance according to our ROC analysis. Balancing our training data using downsizing, oversampling, and cost-based post-processing yielded significant improvement in this informative measure. We anticipate that balance adjustments and the inclusion of more complex features will lead to further improvement.

1. Introduction

Human speech is characterized by modulations in pitch, energy, segment duration, and spectral properties that cause certain words in an utterance to be perceived as more prominent, or accented. Part of being able to speak a language is knowing where to place accents in an utterance; native speakers assign accent with little thought or difficulty. Automatic word accentuation prediction, however, remains a difficult task akin, in the words of one researcher, to mind-reading [1].

Because speech with no pitch accents or incorrectly placed pitch accents can sound unnatural and even confusing, the ability to automatically predict which words in an utterance should be accented is an important task for text-to-speech synthesis (TTS). Word accentuation can be viewed as a classification task: a word in an utterance can be classified as either accented or unaccented. Early accent classification relied on only simple features such as function/content distinctions or part of speech [2]. Recent work, however, has made use of more complex semantic and syntactic features as well as techniques from machine learning, including hidden Markov models, rule-learning algorithms [3], decision trees [4], memory-based learning [5], and neural networks [6].

In this paper, we describe a neural network [7, 8, 9] approach to pitch accent prediction. We have chosen to use neural networks to model word accentuation for a number of reasons. In neural networks, mappings are learned from examples rather than from a priori rules, which require human supervision, or probabilities, which would require a large labeled corpus. Because we plan to use our accentuation model to predict pitch accent placement in novel utterances, we also need a system whose mappings extend effectively to new input data. Neural networks are also able to model nonlinear and complex relationships, which we suspect will be important for this particular task.

2. Description of the corpus

The corpus consists of 16263 words that were obtained from 1030 sentences. The sentences were extracted from the AP newswire using a greedy search algorithm in order to maximally cover the feature space generated by three features: part of speech tag of the previous word, part of speech tag of the current word, and part of speech tag of the next word. There were 72 possible types of part of speech tags. To label the words as accented or unaccented, an innovative iterative perceptual procedure was used in which a labeler used markup tags to indicate accented words in each sentence, synthesized the marked-up sentence using a synthesizer and listened to the resulting synthesized utterance. If the accentuation of any of the words was perceived as incorrect, the markup tags were changed and the rest of the process was repeated until the labeler was satisfied with the word accentuation. The labeler also adjusted the punctuation. For our project, the labeler was a female adult native speaker of American English.

This iterative perceptual labeling procedure that we have outlined has enormous advantages over the commonly used paper and pencil accent labeling procedure. In case of the paper and pencil labeling procedure, the labeler has to *imagine* (or mumble to him/herself) what it would sound like if different words were accented or unaccented. Whereas in case of the iterative perceptual labeling procedure, the labeler will be able to *listen and perceive clearly* the result of accenting or unaccenting different sets of words in a given sentence, which certainly makes the labeling task easier for the labeler, but more importantly, it makes the obtained accent labels more suitable for building a word accent predictor/classifier that will be used for TTS. It is an unavoidable fact that there exists an interaction between accentuation and the TTS-specific acoustic realization of pitch accents. For example, if a TTS system creates ugly pitch accents then it is better to accent fewer words - a reasonable heuristic that the labeler can use *and test* by employing the iterative perceptual labeling procedure, thus making the obtained accent labels more suitable for building a word accent predictor/classifier for text-to-speech synthesis.

* This material is based on work supported by the National Science Foundation under Grants No. 0205731 (ITR: Prosody Generation for Child Oriented Speech Synthesis, Jan van Santen, PI)

Of the 16263 words in the data, 2580 were labeled as accented while 13683 were marked as unaccented. The sentences were passed through the OGI version of the Festival Speech Synthesis System [10] to obtain the following set of eight features per word:

1. word position in the sentence (ranges from 1 to n);
2. type of left phrase boundary (0 or B, boundary type);
3. type of right phrase boundary (0 or B, boundary type);
4. distance from the left phrase boundary (0, 1, or 2);
5. distance from the right phrase boundary (0, 1, or 2);
6. part of speech of the previous word;
7. part of speech of the word; and
8. part of speech of the next word.

The part of speech tags were mapped to seven categories: "noun", "verb", "adjective", "adverb", "number", "pronoun", and "other". This mapping procedure was performed because using all the part-of-speech tags initially produced (18) would drastically increase the amount of training data required. In addition, the tags mapped to the category, "others", consist of function words of which approximately 93% are unaccented.

The accentuation labels are the targets in the classifier, while the low-level syntactic and prosodic features obtained from Festival are the training features of the classifier. A binary encoded training vector is obtained from the accentuation labels such that the class value for each element in the vector is 0 or 1, indicating unaccented and accented, respectively.

The training features, which have a variety of scales, are also normalized so that their values range from 0 to 1. The word position in the sentence is scaled to be between 0 and 1 using the simple scaling formula: $X_{scaled} = (X - X_{min})/(X_{max} - X_{min})$. The remaining features extracted from Festival are encoded as n binary inputs, where n is the number of values that each feature can take. For example, the feature relating to distance from the left phrase boundary can assume three values: 0, 1 or 2. This feature in its normalized form is represented as 3 binary input units, $\{$(1 0 0), (0 1 0), (0 0 1)$\}$. Normalizing the data in this manner yields a total of 32 scaled feature vectors that will be used as inputs to the classifier.

¿From this corpus, 20% of the data was randomly selected as a test set, and 20% was randomly selected as the validation set. The remaining 60% (= 9757 words) was used as the training set to train the neural net classifier.

3. Measuring classifier performance

The corpus is highly imbalanced. In this two-class corpus, 85% of the corpus is the class of unaccented words, while only 15% of the corpus is accented words, the class of interest in this prediction task. Since the training, test, and validation sets were selected randomly, it can be assumed that each of these sets contains a bias similar to the bias of the whole corpus.

Most classifiers trained on such a biased training set can predict instances of the majority class with a high degree of accuracy but have very low predictive accuracy on instances of the under-represented class. The measure, *classification accuracy* (defined as the ratio of the number of correctly predicted instances to the total number of instances in the test set) is not a good measure for assessing the performance of a classifier trained and tested on such biased sets. Classification accuracy assumes a test set in which both classes are equally represented. However, as in our case, if the two classes are distributed in the ratio 85:15, and the classifier predicts the unaccented class with 100% accuracy but misclassifies the instances of the accented class completely, the classifier accuracy would still be 85%. This metric would not reflect the fact that the class of accented words, which is the class of interest, is completely misclassified.

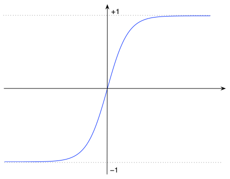


Figure 1: *The tangent-sigmoid (tansig) function*

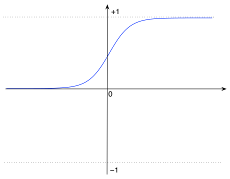


Figure 2: *The log-sigmoid (logsig) function*

The metric better suited to such imbalanced datasets is the *area under the ROC curve*. ROC refers to the Receiver Operator Characteristics of a classifier. It is obtained by plotting the *true positive rate* against the *false positive rate*, thus illustrating the tradeoff between these two quantities. The *false positive rate* is the rate at which negative instances were misclassified as positive, while the *true positive rate* is the rate at which positive instances were classified as positive. The ROC curve has a 0-to-1 scale on both axes.

The area under the ROC curve is a good metric for comparing the performance of different classifiers. The larger the area, the better the classifier. A perfect classifier has an area of 1, indicating a 100% true positive rate and 0% false positive rate. A classifier that randomly guesses has a ROC curve that lies on the diagonal line connecting (0, 0) and (1, 1) and has an area of 0.5. For our project, we will be considering the area under the ROC curve as the metric for measuring the performance of the classifiers that predict word accentuation.

4. Baseline classifier

The neural network classifier was created as a three-layer feedforward backpropagation neural network, using the Matlab command, `newff` [11]. This command takes as input the maximum and minimum values of the input nodes, the number of hidden neurons, the number of output neurons, the transfer functions of each layer, and the network training function that updates weight and bias values.

We selected `tansig` [12], the tangent-sigmoid function, depicted in Figure 1 as the transfer function at the hidden layer. The `tansig` function used in the Matlab Neural Networks Toolbox, is mathematically equivalent to the hyperbolic tangent function, whose output values range from -1 to +1. The input data was accordingly normalized to lie on the -1 to +1

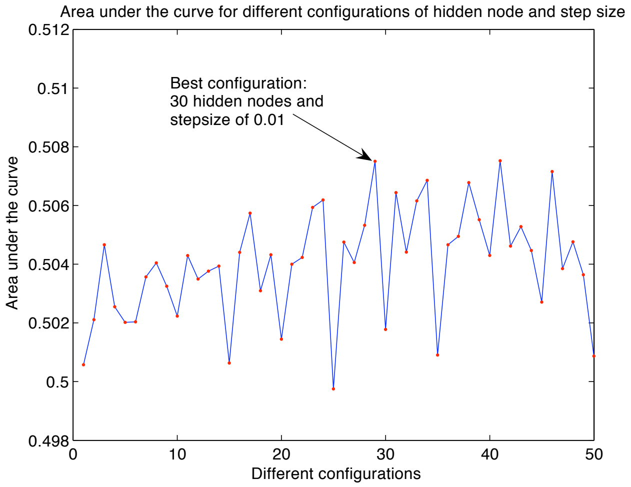


Figure 3: *The change in area under the ROC curve for different configurations of hidden nodes and step size and no momentum values.*

scale. The transfer function we selected for the output layer was `logsig` [13], the log-sigmoid transfer function shown in Figure 2, whose output values range from 0 to 1, the desired output range of our classifier. We selected `traingdx` [14] as the network training function. This function updates weight and bias values according to gradient descent momentum and an adaptive step size.

We specified a single output neuron for this classifier. Since the output transfer function is the log-sigmoid function, the output values range from 0 to 1. The output value is treated as the probability that the word is accented, and the word is classified in the following manner: if the output value is less than 0.5, the word is unaccented; if the output value greater than or equal to 0.5, the word is accented.

We also specified 30 hidden layer neurons, a learning rate (or step size) of 0.01, and a momentum value of 0.6. The last three values were obtained from a calibration process that is described in Section 5. The weights and biases were randomly specified by Matlab. Note that we used early stopping when building our neural nets in order to avoid overfitting.

5. Calibration of the neural net classifier

The calibration process was a two-step process. The first step involved using no momentum values and systematically varying the number of hidden layer neurons and the step size (implemented as a double for-loop in Matlab). The number of hidden layer neurons were varied in this way: (5, 10, 15, 20, 25, 30, 33, 40, 45, 50). The step size was varied as follows: (0.00001, 0.0001, 0.001, 0.01, and 0.1). For each choice of hidden layer neurons and step size, the neural net was initialized 10 times with random weight and bias values.

For each initialization, the obtained neural net classifier was used for accent prediction of the word examples in the validation set. Using the predicted values and the known true values, the area under the ROC curve (AUC) was computed. For each configuration of n hidden layer neurons, and step size m, the mean AUC of the 10 initializations was plotted on a graph in Figure 3. This was done for all 50 configurations that emerge from the possible values of step size and number of hidden layer neurons. From the graph, the configuration with 30 hidden layer neurons and a step size of 0.01 was selected for the baseline neural net, because it yielded the maximum mean AUC.

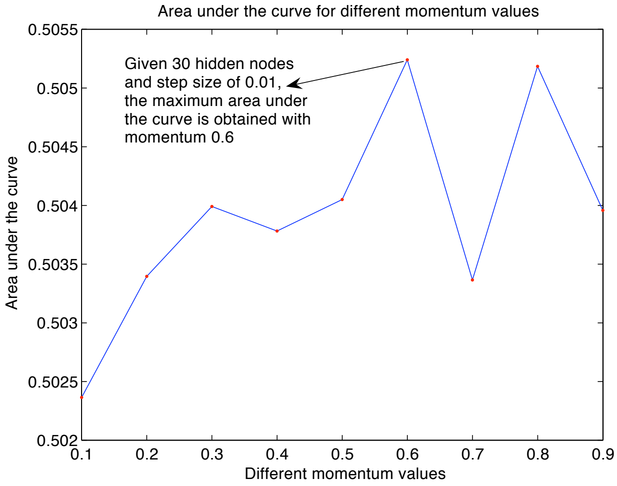


Figure 4: *The change in area under the ROC curve for different momentum values for 30 hidden nodes and step size of 0.01*

The next calibration step involved finding the best momentum value. Using 30 hidden layer neurons and the step-size of 0.01, the momentum values were systematically varied from 0.1 to 0.9. For each momentum value, the neural net was initialized 10 times with random weight and bias values. Again, the resulting neural net was tested on the validation set. and the mean area under the ROC curve for each momentum value was plotted on a graph, as shown in Figure 4. From this figure, we found that the maximum mean AUC was obtained for momentum value 0.6, which we selected as the momentum value for the baseline.

6. Imbalanced training set

As shown in Figure 5, the baseline classifier has a false positive rate (= 100-true negative rate) of 0.8364% on the validation set and 0.9158% on the test set. The low false positive rate indicates that the baseline classifier predicts unaccented words with high accuracy (99.17% for the validation set and 99.08% for the test set). However, it does not predict accented words well, as indicated by the low true positive rate (2.3857% on the validation set and 3.2505% on the test set). It misclassifies most of them as unaccented. This state of the classifier performance is reflected in the metric, the area under the ROC curve, which value is only a little over 0.5 for both the validation set and the test set, indicating that the baseline classifier performs barely over chance. (The classification accuracies of 84.1992% and 83.6766% on the validation set and test sets, respectively, are a reflection of the high proportion of unaccented examples in the test set, which the classifier learned to predict well, as discussed in Section 3.)

To improve the performance of the baseline classifier which was trained on a highly imbalanced training set, we employ three techniques that have been demonstrated to improve the classifier's ability to predict the minority class: 1) Downsizing, 2) Oversampling, and 3) Cost-based thresholding. Each of these

	Baseline NN		Downsizing NN		Oversampling NN		Cost-based NN	
	Valset	Testset	Valset	Testset	Valset	Testset	Valset	Testset
AUC	0.5077	0.5117	0.6759	0.6419	0.6808	0.6457	0.6597	0.6483
TPR	2.3857	3.2505	62.8231	56.2141	70.7753	62.9063	68.1909	64.2447
FPR	0.8364	0.9158	27.6364	27.8388	34.6182	33.7729	36.2545	34.5788
Acc	84.1992	83.6766	70.8884	69.5973	66.2158	65.6932	64.4328	65.2321

Figure 5: *Results of the performance of the four different types of classifiers on the validation set and the test set. AUC = area under curve, TPR = true positive rate, FPR = false positive rate, Acc = overall accuracy*

methods will be described in the next sections.

7. Solution 1: Downsizing

The first solution that we used to improve the performance of the neural net classifier is called downsizing. Downsizing involves removing random examples of the over-represented class (in our case, unaccented words) from the training set to match the number of examples in the under-represented class (in our case, accented words) [15]. This method is called downsizing because the size of the balanced training set is smaller than the overall training set. This method hinges on the concept that the under-represented class is the class of interest, and all examples of that class need to be retained. In [16], it was found that downsizing is effective in improving the performance of the neural net classifier.

For our project, we downsized the training set in this manner 100 times. For each of the 100 repetitions, the neural net was initialized with random weight and bias values; however, in all repetitions, the 30 hidden layer neurons, step size of 0.01, and momentum value of 0.6 were used. For each repetition, the resulting neural net was tested on the examples in the validation set, and the area under the ROC curve was calculated. After 100 repetitions, the neural net that maximized the area under the ROC curve was selected as the optimal neural net classifier obtained by downsizing our training set, and it was tested on the examples on the test set.

8. Solution 2: Oversampling

The second technique that we used to improve the performance of the neural net classifier was oversampling. Oversampling involves balancing the training set by duplicating random examples of the under-represented class [17] until the number of examples in each class is equal. When the target concept (in our case, ”is a word accented?”) is represented by fewer examples, oversampling can train the classifier to give more weight to features determining the target concept. This method was also studied in [16], and was found to improve the performance of neural net classifiers. As the size of the training set increased, however, downsizing outperformed oversampling.

For our project, we oversampled the training set in this manner 100 times. For each of the 100 repetitions, the neural net was initialized with random weight and bias values; however, in all repetitions, the 30 hidden layer neurons, step size of 0.01, and momentum value of 0.6 were used. For each repetition, the resultant neural net was tested on the examples in the validation set, and the area under the ROC curve was calculated. After 100 repetitions, the neural net that maximized the area under the ROC curve was selected as the optimal neural net classifier obtained by oversampling on our training set, and it was tested on the examples on the test set.

9. Solution 3: Cost-based thresholding

Cost-based classification is the third solution for improving the performance of the neural net classifier on the word accentuation task. This method looks at the different errors made by the classifier in terms of relative cost [18]. Cost-based classification is a post-process algorithm, i.e., the classification may be performed using any classifier that outputs the probability of an instance being positive or negative. However, the probability threshold that distinguishes the positive instance from the negative instance (so far, at 0.5) is not necessarily symmetric. It takes the cost asymmetry into account as follows. Let:

- p, n be the positive and negative classes,
- Y, N be the classifications produced by a classifier,
- $c(Y, n)$ be the cost of a false positive error,
- $c(N, p)$ be the cost of a false negative error, and
- $CR = c(Y, n)/c(N, p)$ be the cost ratio.

For an instance, E, the classifier outputs the probability of being positive, $p(p|E)$. The probability of being negative, $p(n|E)$, is computed as $p(n|E) = 1 - p(p|E)$, assuming a two-class dataset. Thus, according to cost-based classification theory, a word is classified as positive if $p(n|E) * CR < p(p|E)$.

For our project, we used the baseline neural net classifier obtained in Section 4. We systematically varied the cost ratio from 0.0001 and 2 with a step size of 0.001. For each value of the cost ratio, we tested the neural net classifier on the validation set and calculated the area under the ROC curve and plotted it on a graph shown in Figure 6. ¿From this graph, we found that the value of the cost ratio that maximizes the area under the ROC curve is 0.1731. Thus, the cost ratio of 0.1731 in combination with the baseline classifier formed the cost-based classifier.

10. Results and discussion

As the table in Figure 5 shows, the baseline neural net had the highest overall accuracy but extremely low rates of both true positives and false positives. We see that the area under the curve is slightly larger than 0.5, indicating accent assignment with this neural net is slightly better than chance. Earlier we speculated that the imbalanced nature of our data, in which only 15% of elements are classified as positive (accented), would lead to these kinds of results, since the classifier could learn to always return negative and still achieve 85% accuracy. We investigated three different ways to alleviate the imbalance of our data: downsizing, oversampling, and cost-based thresholding.

Figure 5 shows that the highest overall accuracy rate among the three balanced alternatives, though still lower than the baseline, comes from the neural network that was balanced using downsizing (Acc = 69.59% in the test set). The false positive rate of the downsized net is the lowest of the three alternative

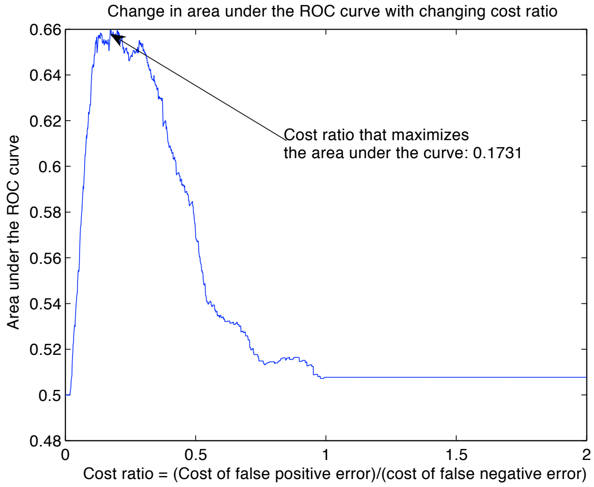


Figure 6: *The change in area under the ROC curve with the change in cost ratio (defined by cost of a false positive error to cost of a false negative error).*

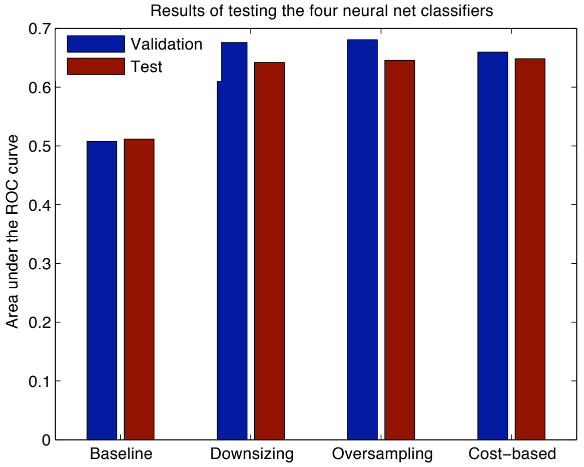


Figure 7: *The performance of the four neural net classifiers in terms of the area under the ROC curve.*

nets. The true positive rate, however, while much higher than that of the baseline neural network, is somewhat lower than the true positive rates of the other two balanced models.

The performance measures of the neural network using oversampling and the neural network using cost-based thresholding are remarkably similar, particularly on the test set. We observe only a slightly higher true positive rate in the cost-based model (64.2447 v. 62.9063), accompanied by a slightly higher false positive rate (34.5788 v. 33.7729). Notably, we see a larger drop in true positive rate from the validation set to the test set in the oversampled neural network than in the other two models. We also observe that the results of the cost-based classifier are more consistent across the validation and test sets that those of the other balanced neural networks.

Figure 7 illustrates perhaps the most interesting observation: our key metric, the area under the ROC curve, is roughly the same for the test set in all three balanced neural networks (0.6419, 0.6457, 0.6483) and is noticeably higher in all three balanced nets than in the baseline (0.5117)

Since all three balanced neural networks had similar AUC measures, choosing the ideal accentuation classifier from the three balanced neural networks will depend on other factors. Accenting a word that should be unaccented is perceived as much more “wrong” than leaving a word unaccented that should be accented. We therefore might prefer the downsized model, since its false positive rate is the lowest of the three. On the other hand, if we are concerned about replicating our findings with new data, we might select the neural network with cost-based thresholding since its performance was more consistent between the validation and test sets.

We had hoped to compare our results to those reported in other recent research in pitch accent prediction using machine learning techniques. Few of these results, however, include measures such as the area under the ROC curve or the false positive and true positive rates. Pan and McKeown [3] report 70-74% accuracy rate with their word-informativeness-based HMM and RIPPER models, a significant improvement over the 52% baseline they assume given the composition of their data set, in which 52% of words are accented. Unfortunately, they do not report other measures of accuracy. In addition, the composition of their data is so different from ours, in which only 15% of words were accented, that a comparison between the two models might not be valid. Similarly, Hirschberg [4] reports overall accuracy of 80-98.3% with both hand-written rule systems and classification and regress trees, but fails to report other measures. Ross and Ostendorf [19], who also used CART techniques, realized similarly high accuracy, but we cannot compare their results to ours since they predicted accent on the syllable-level rather than word level.

One valid point of comparison comes from Marsi et al. [5], who report both higher accuracy (86%) and a higher true positive rate (or recall, 82-88%) than we achieved with our neural network. Their data set, however, was more balanced (one-third of words were accented), and their feature set was far richer.

The most telling results are found in Müller and Hoffman [6], who also used neural networks to model accent prediction. Their word-level neural network classifier achieved 84.5% average overall accuracy, with a low false positive rate (9.3%) and a high true positive rate(85.5%), both of which are noticeably better than the results we found. The HMM they developed using the neural network output for the emission probabilities had very similar accuracy, false positive rate, and true positive rate. We suspect that their training data was more balanced than ours, given that they included secondary and emphatic accents, while we considered only primary accent. In addition, although their feature set was similar to ours, they looked at potentially very long sequences of parts of speech and phrase break locations rather than the simply the properties of the immediately adjacent words.

11. Conclusions

The three techniques we used to balance our unbalanced data set yielded noticeable improvements over our baseline neural network in the measure of area under the ROC curve and the true positive rate, two of our key metrics. We expect to see further improvement by expanding our feature set to include higher-level syntactic properties, such as parent node or depth in the syntactic tree, more precise information about the location of the previous accented word and phrase break [6], and semantic features, such as word informativeness [3, 5] or the features used with success in data-to-speech systems [20]. In addition, we might see gains from using training data that are nearly but not perfectly balanced, given Estabrooks' [21] findings that downsizing and oversampling achieve their lowest error rates when the training set is slightly imbalanced. Finally, we would like to investigate training our neural net with other corpora. Our corpus was labeled to maximize the naturalness of TTS output, which seems to have resulted in an imbalance not seen in the corpora used in other machine learning approaches to accent prediction. Our goal in choosing this labeling process was to create a word accentuation model that would translate elegantly to real-world speech synthesis applications, but perhaps by training with a less skewed set of training data, we can realize the recall and accuracy reported elsewhere in the literature.

12. References

[1] Bolinger, D., "Accent Is Predictable (If You're a Mind-Reader)." Language, 48:3, 633-644, 1972.

[2] Altenberg, B., "Prosodic Patterns in Spoken English: Studies in the Correlation between Prosody and Grammar for Text-to-Speech Conversion", Lund Studies in English, vol. 76, Lund: Lund University Press, 1987.

[3] Pan, S., and McKeown, K., "Word Informativeness and Automatic Pitch Accent Modelling", Proceedings of the Joint SIGDAT Conference on EMNLP and VLC, 148-157, 1999.

[4] Hirschberg, J., "Pitch Accent in Context: Predicting Intonational Prominence from Text", Artificial Intelligence, 63:1-2, 305-340, 1993.

[5] Marsi, E., Busser, G. J., Daelemans, W., Hoste, V., Reynaert, M., and van den Bosch, A., "Combining information sources for memory-based pitch accent placement", Proceedings of the International Conference on Spoken Language Processing, 1273-1276, 2002.

[6] Müller, A., and Hoffmann, R., "A neural network and a hybrid approach for accent label prediction", Proceedings of the 4th ISCA Tutorial and Research Workshop on Speech Synthesis, paper 102, 2001.

[7] McCulloch, W. and Pitts, W., "A logical calculus of the ideas immanent in nervous activity", Bulletin of Mathematical Biophysics 5, 115-33, 1943.

[8] Rosenblatt, F., "The Perceptron: A probabilistic model for information storage and organization in the brain", Psychological Review 65, 386-408, 1958

[9] Werbos, P., The Roots of Backpropagation: From Ordered Deriatives to Neural Networks and Political Forecasting, New York: Wiley, 1994

[10] Black, A., and Taylor, P., "Festival Speech Synthesis System: System documentation (1.1.1)", Human Communication Research Centre Technical Report HCRC/TR-83, Edinburgh, U.K, 1997.

[11] MathWorks, Inc. NEWFF. MATLAB function. Obtained from the MATLAB command 'type newff'. August 2005.

[12] MathWorks, Inc. TANSIG. MATLAB function. Obtained from the MATLAB command 'type tansig. August 2005.

[13] MathWorks, Inc. LOGSIG. MATLAB function. Obtained from the MATLAB command 'type logsig. August 2005.

[14] MathWorks, Inc. TRAINGDX. MATLAB function. Obtained from the MATLAB command 'type traingdx'. August 2005.

[15] Kubat, M., and Matwin, S., "Addressing the curse of imbalanced training sets: one-sided selection", Proceedings of the 14th International Conference on Machine Learning, 179-186, 1997.

[16] Japkowicz, N., "The Class Imbalance Problem: Significance and Strategies", Proceedings of the 2000 International Conference on Artificial Intelligence 1, 111-117, 2000.

[17] Ling, C., and Li, C., "Data mining for direct marketing: Problems and solutions", Proceedings of the 4th International Conference on Knowledge Discovery in Databases (KDD-98), New York, 1998.

[18] Pazzani, M., Merz, C., Ali, K., and Hume, T., "Reducing Misclassification Costs", Proceedings of the International Conference on Machine Learning 1994.

[19] Ross, K., and Ostendorf, M., "Prediction of abstract prosodic labels for speech synthesis", Computer Speech and Language, 10(3), 155-185, 1996.

[20] Theune, M., "Parallelism, Coherence, And Contrastive Accent", Proceedings of Eurospeech 1999, 555-558, 1999.

[21] Estabrooks, A., "A combination scheme for inductive learning from imbalanced data sets", Master's Thesis, Dalhousie University - Daltech, 2000.

Issues of Optionality in Pitch Accent Placement

Leonardo Badino, Robert A.J. Clark

Centre for Speech Technology Research
University of Edinburgh, Edinburgh, UK
l.badino@sms.ed.ac.uk, robert@cstr.ed.ac.uk

Abstract

When comparing the prosodic realization of different English speakers reading the same text, a significant disagreement is usually found amongst the pitch accent patterns of the speakers. Assuming that such disagreement is due to a partial optionality of pitch accent placement, it has been recently proposed to evaluate pitch accent predictors by comparing them with multi-speaker reference data. In this paper we face the issue of pitch accent optionality at different levels. At first we propose a simple mathematical definition of intra-speaker optionality which allows us to introduce a function for evaluating pitch accent predictors which we show being more accurate and robust than those used in previous works. Subsequently we compare a pitch accent predictor trained on single speaker data with a predictor trained on multi-speaker data in order to point out the large overlapping between intra-speaker and inter-speaker optionality. Finally, we show our successful results in predicting intra-speaker optionality and we suggest how this achievement could be exploited to improve the performances of a unit selection text-to-speech synthesis (TTS) system.

1. Introduction

In this paper we propose a new evaluation function for evaluating pitch accent predictors and a novel approach that exploits the variability of pitch accent patterns in order to improve the prosodic realization of a unit selection TTS system. In natural speech, alternative prosodic realizations of a given utterance can be equally acceptable. Even when a speaker is required to utter a sentence in a specific standard speech style (that of radio news speakers, for example) she/he will be free to choose amongst different prosodic patterns without altering the meaning of the sentence [1]. This freedom of choice affects different aspects of prosody, ranging from prosodic phrasing to the intonation contour. This prosodic variability offers a further degree of freedom to the developers of speech synthesis systems (or at least to those using the unit selection technique) who want to create systems able to go beyond a neutral prosodic realization making them able to convey additional meaning through prosody. In unit selection, a predefined prosodic target is usually expressed by a sequence of symbolic values describing F0 and segmental duration. These prosodic values are included into the specifications of the target utterance. The target is matched by selecting the appropriate acoustic units and, in some cases, by applying signal processing techniques. In such a context, imposing one single predefined prosodic target can involve a large amount of speech processing and a drastic reduction of the unit search space, thus resulting in a poor quality speech production, usually less acceptable than that of a system not supported by any prosodic model. As a consequence, and taking into account the prosodic variability of natural speech, new "softer" approaches have been recently proposed, for example, in [2] alternative prosodic patterns are implemented into a weighted-finite-state-transducer (WFST), which is then composed with the WFST describing the segmental information of the acoustic database. The unit sequence with the best combined cost is chosen. Prosodic constraints can be further relaxed by dropping the idea of explicitly defining the allowed prosodic patterns and selecting an implicit prosodic model by relying on the inherent prosodic structure of the speech database [3]. In our work we focused on the variability of prosodic patterns looking at a single type of prosodic event: pitch accent. We first analyzed the section of the Boston University Radio News corpus [4] where speech data have been collected by recording different speakers reading the same sentences. We show, for any combination of speakers, the intra-speaker disagreement in placing pitch accents. Then, starting from previous work, we faced the problem of evaluating pitch accent predictors on multi-speaker data, assuming that the intra-speaker disagreement is mainly due to a high degree of optionality in placing pitch accents. Our solution implies a simple mathematical definition of optionality which led us to the formulation of a new evaluation function. Subsequently, we tested our main work hypothesis, that is the assumption that the optionality observed when comparing the prosodic realization of different speakers (intra-speaker optionality) largely overlaps with inter-speaker optionality, that is the optionality that would be found if a speaker repeatedly read the same text without changing is speaking style. We compared a pitch accent predictor trained on single speaker data with a predictor trained on multi-speaker data. From the high similarity of performances of both predictors we inferred the validity of our hypothesis. Finally, we found out that our definition of optionality was determinant in our successful attempt of predicting optionality and, supported by the high similarity of intra and inter speaker optionality, we devised a simple method to exploit this achievement in order to improve the prosodic realization of a unit selection TTS system that uses pitch accent prediction to model prosody.

2. Disagreement Among Speakers

A section of the Boston University Radio News (BURN) corpus contains the speech of six different speakers (3females: f1a, f2b, f3a, and 3 males: m1b, m2b, m3b) reading the same text. All data have been prosodically labeled using the ToBI annotation conventions. We used this annotation only to see if a pitch accent occurred or not (see Figure1).This part of the BURN corpus was already analyzed in [5] to investigate the intra-speaker disagreement in pitch accent placement. However, here, we provide some further data, useful for our purposes. Figure 2 shows the percentages of intra-speaker agreement for each combination of speakers and the agreement mean, with respect to the

	f1a	f2b	f3a	m1b	m2b	m3b
may	N	A	A	A	A	A
be	N	N	N	N	N	N
the	N	N	N	N	N	N
most	N	A	N	A	N	A

Figure 1: *An example extracted from the BURN corpus. A and N stand for accent and no-accent respectively*

number of speakers involved, on a text of 1662 words. The vertical segments range from the lowest to the highest agreement percentage, given a certain number of speakers. For example, given a number of two speakers, there are 15 possible combinations of speakers. Among them the pair with the lowest agreement (79.19%) is f1a-m2b, whereas the highest agreement (85.86%) occurs in m1b-m3b. These two percentages may suggest a correlation between degree of agreement and speaker genre, but if we look at all the 20 possible triplets of the six speakers we see that the combination with the highest agreement (77.61%) is f2b-m1b-m3b, which consists of one female and two males. We did not carry out any study to investigate which are the factors that correlate to intra-speaker agreement and to what extent, but from an informal analysis it seems that speaker profession (is she/he a professional speaker?) is at least as significant as speaker genre.When comparing the agreement among speakers in pitch accent placement we can compute the proportion of agreement that is not due to chance by using the Kappa statistics:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where P(A) is the proportion of times speakers agree and P(E) the proportion we would expect them to agree by chance. In our case, assuming that accent and non-accent are equiprobable (the percentage of accented words for this speech style ranges from 45% to 55%) the κ value for the six speakers is 0.57.

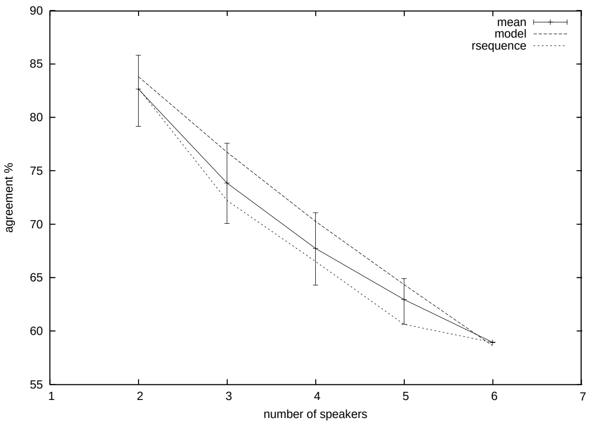


Figure 2: *Speakers agreement in pitch accent placement. The mean line represents the mean disagreement value. The rsequence line shows the disagreement resulting when adding a speaker in the order: f1a, f2b,f3a,m1b,m2b,m3b.*

3. Optionality and Pitch Accent Predictor Evaluation

3.1. Previous Works

If we make the assumption that when two or more speakers disagree in placing, or not, a pitch accent on a syllable, that pitch accent can be considered an optional accent, then we can reconsider the usual evaluation practice in which a pitch accent predictor is compared with only a single speaker. [5] and [6] used an evaluation function that considers a predicted event (accent or no-accent) wrong if it is not yielded by any of the speakers/annotators. Although the two works differ for the language (English vs Dutch) and the type of data test used (prosodically annotated speech vs prosodic labels directly derived from text) their conclusions are very similar: when optionality is taken into account in evaluating their automatic pitch accent predictors the performances of their predictors are very close to those of humans. This conclusion assumes that the optionality occurring when comparing speakers (intra-speaker optionality), is the same optionality that can occurs within a single speaker (inter-speaker optionality). As a consequence the accent pattern chosen by a speaker is made up of a compulsory part and an optional part, which can be exchanged with the optional part of (an)other speaker(s) without altering the coherence and naturalness of the whole accent pattern. There are however possible side-effects in this assumption. First, even if a pitch event is optional all the speakers can choose the same value. Second, the optional part of the pitch accent pattern of a single speaker can be related to the speaking style of the speaker herself/himself and, moreover, can be influenced by other factors that determine her/his speaking style, for example her/his speaking speed. As a consequence, mixing a speaker optional part with that of other speakers may result in an unnatural and "distorted" pattern. Finally, the evaluation function used in both works ignores a possible sintagmatic behavior of pitch accents: the placement of an accent can influence the placement of the following ones. In spite of that, in our work we kept the idea of evaluating accent predictors comparing them with multi-speaker data, supported by the fact that, as we will show later, fortunately, part of these side-effects is probably not so significant as it may seem at a first glance and can be reduced using a different evaluation function. Nevertheless, even assuming that these side-effects do not occur, the evaluation functions proposed in the previous works have still significant drawbacks. Figure 2 shows how the speaker agreement quickly decreases when the number of speakers increases. As a consequence it is easy to see how the evaluation function of [5] and [6] is strongly dependent on the number of speakers involved.

Figure 3 shows this fact by comparing three predictors (one of those is actually a speaker) varying the number of speakers involved in the test. The more the speakers in the test data are, the lower the intra-speaker agreement is and consequently the better the predictor results are. Consider the predictor A, which assigns a pitch accent to each words. If it is evaluated on six speakers, its accuracy rate is 73%, that means that we could build a predictor that accents the 73% of overall words, and performs a 100% of correct predictions. But, since the percentage of pitch accent in read speech ranges from 45% to 55% such a predictor is not appropriate to model pitch patterns of real speech. When looking at the speaker disagreement we should take into account that the steep decrease is partially due to the simple fact of adding new speakers even if the disagreement in each pair of speakers is low. In order to better illustrate that

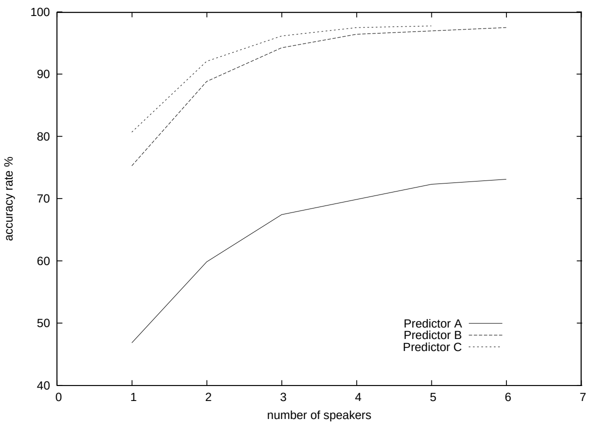


Figure 3: *Three predictors tested over different numbers of speakers. The sequence of speakers combination is f1a, f1a-f2b, f1a-f2b-f3a, f1a-f2b-f3a-m1b, f1a-f2b-f3a-m1b-m2b, f1a-f2b-f3a-m1b-m2b-m3b. Predictor A is an all-accented predictor. Predictor B is described in section 5. Predictor C is the speaker m3b.*

we could suppose that each word token in the test text has a non-zero probability of being optional, that is of being assigned both accent values (accented/non-accented) and that each pitch accent is independent from the others. If we assume p being the average probability of the most probable event for each word token, the agreement percentage can be modeled as:

$$(m1) \quad A(n) = 100[p^n + (1-p)^n]$$

where n is the number of speakers involved. In Figure 2 we plotted $A(n)$ (model) setting p to 0.9157. This value was obtained by imposing p^6 (the term $(1-p)^6$ was ignored) equal to the real agreement of six speakers (58.96%).

Even if our model is certainly approximate it clearly shows how even for high values of p the agreement percentage rapidly decreases by adding new speakers and gives a clue of what happens if more than six speakers are compared. Moreover this model allows us to see the intra-speaker optionality value not as a simple binary value but as a gradient one, which is a function of the probability of each word token of being assigned both pitch events. This concept is the base of our work.

The number of speakers is not the only parameter that can influence the predictors evaluation: the evaluation function of Figure 3 considers correct a pitch event if it is realized by at least one speaker, but we could be more strict and choose an evaluation function that marks as correct a predicted pitch event only if it is realized by more than one of the speakers involved. Considering n the number of speakers involved in the test and m (with $m < n$) the acceptable (for the evaluation function) number of speakers that realize the same pitch event of the predictor, we can write the evaluation function for each word token i:

$$OE(w_i) = \begin{cases} 1 & \textit{if at least m speakers realized} \\ & \textit{the predicted event} \\ 0 & \textit{otherwise} \end{cases} \qquad (1)$$

	$m = 1$	$m = 2$	$m = 3$
Predictor A	73.17	65.04	60.16
Predictor B	97.56	94.06	88.89

Table 1: *Accuracy rates of two predictors for different values of* m *(*$n = 6$*).*

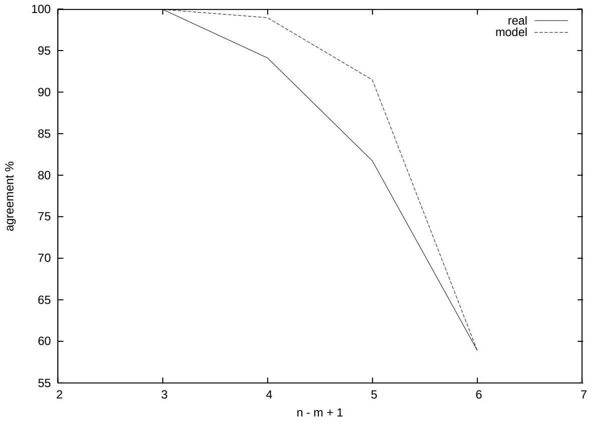


Figure 4: *Speakers agreement for different values of m (n=6)*

Table 1 shows the evaluation of two predictors already used in figure 3, this time always compared with all the six speakers ($n = 6$) but varying m. The high dependency of the evaluation function on m is again explained by the speaker disagreement: when m increases the number of cases in which the pretiction is considered correct independently on its value decreases. For example if $m = 1$ the prediction is always correct in all the cases where at least one speaker disagrees whereas it can be wrong or correct only when all the speakers agree. In figure 4 we plotted the percentage of pitch events that are consistent among all the six speakers (bottom right), at least five of the six speakers and so on. We also plotted an agreement function based on the same hypotheses made for (m1). Since the number of combinations of k speakers taken from a set of n speakers is given by $\left(\begin{array}{c} n \\ k \end{array}\right)$, in this case the agreement function is:

$$(m2) \quad A(n,m) = 100 \sum_{k=n-m-1}^{n} \left(\begin{array}{c} n \\ k \end{array}\right) [(1-p)^{n-k} p^k + (1-p)^k p^{n-k}]$$

where $0 \leq m \leq 4$, and the p value is set to the same value used for figure 2. Note that p was not set to find the best model of rsequence (in terms of Root Mean Square, for example).

3.2. An alternative evaluation function

Starting from the considerations made above we wanted to formulate an evaluation function that awarded those predictors able to match the average accent pattern of human speakers and that was less sensible to n and m.

To satisfy these specifications we associated an emission source to each word token. Each source can emit two symbols, one when the token is accented and one when it is not. The number of emissions is equal to the number of speakers and each emis-

	f1a	f1a-f2b-f3a-m1b-m2b-m3b	Δ(diff. between the first 2 colums)	ΔBaseline/ΔPredictor B
Baseline, OE(m=1)	46.88	73.17	26.29	-
Baseline, EE	48.88	69.54	20.66	-
Predictor B, OE(m=1)	75.34	97.56	22.22	1.18
Predictor B, EE	75.34	95.00	19.66	1.05

Table 2: *Comparison between OE and EE on predictor B and A (baseline).*

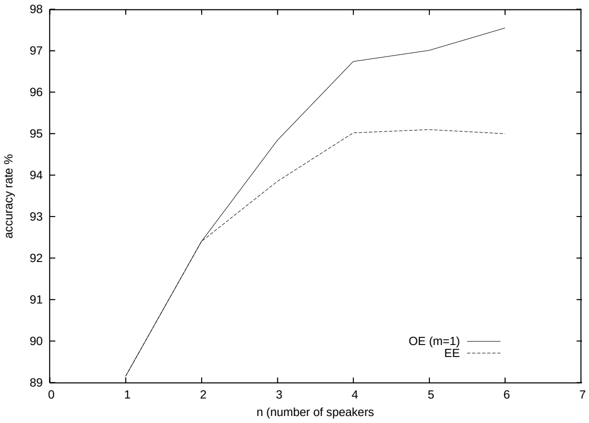


Figure 5: *Accuracy rates of Predictor B using OE and EE.*

sion is independent form the others.
From Information Theory ([7]) we know that the entropy of such a source is:

$$H = -\log(P(A))P(A) - \log(P(N))P(N) \qquad (2)$$

where $P(A)$ is the probability that the source emits an accent and $P(N)$ that it does not. The entropy says how much information we need (or more informally, how many questions we have to ask) to correctly predict the next symbol that will be emitted by the source. If the source has always emitted the same symbol than its entropy will be 0, whereas if the number of emissions of both symbols is equal then the entropy value will be 1. In all the other cases (and if the number of emissions is higher than 2) the entropy value will be less than 1 and more than 0. If we associate the optionality of a word token with its entropy, and search for an evaluation function that is dependent on optionality, we can write the evaluation function for a single token as follows:

$$EE(w_i) = 1 - [(1 - P_t(pe_i))(1 - H_t(w_i))] \qquad (3)$$

where $P_t(pe_i)$ is the probability that the predicted event pe_i is emitted by the test source and $H_t(w_i)$ is the entropy of the source. The overall EE is the sum of each $EE(w_i)$ divided by the total number of words.
The main novelty of EE is that intra-speaker optionality is no more simply considered as a binary quantity but as a gradient one.
Concerning the dependency on n and m, one of the practical advantages of EE is that we do not have to decide which the most appropriate value of m is, while regarding n we can see how EE is more stable than OE to n increase, if we suppose of having an infinite number of speakers. In that case, it is acceptable to assume a non-zero probability for each token of being assigned both pitch events, especially if we think that an error can be made by the speakers themselves or by the prosodic annotators. Both an all-accented and an all-non-accented predictors would score $OE(w_i) = 1$ per each token though neither of them would match the speakers average pitch pattern. Using EE both predictors would never reach the maximum score. This is an interesting characteristic of EE since usually a predictor performance is evaluated relatively to an all-accented or an all-non-accented baseline.
In order to provide some empirical evidence of the higher stability of EE we compared the two functions using different predictors. In figure 5 a predictor is evaluated on different values of n: for $n > 3$ the EE values are more stable than the OE values which keep on rising. Figure 5 shows the result for only one predictor evaluated over one out of 720 possible sequences of speakers. We carried out the same type of comparison using different predictors and different speaker sequences finding always the same kind of result. Table 2 reports the results of another type of comparison between EE and OE (with OE having $m = 1$). For both functions we computed the difference between the value obtained with $n = 1$ (first column) and that one with $n = 6$ (second column).
The table shows (third column) that for both measures the difference (Δ) between $n = 1$ and $n = 6$ obtained on the all-accented predictor is higher than our predictor, that means that the baseline increases more quickly than our predictor. Nevertheless, using EE, the increase of the baseline with respect to our predictor is slightly smaller: the fourth column of table 2 shows that when using EE the ratio between the Δ's of baseline and predictor (fourth row) is lower than that obtained using OE (third row). The choice of the speaker when $n = 1$ is not determinant since when a predictor is compared to a single speaker EE and OE assign the same score.

4. Intra-Speaker and Inter-Speaker Optionality

Until now we have seen how intra-speaker optionality can be taken into account when evaluating a pitch accent predictor assuming that the optionality occurring among speakers is the same optionality occurring within a single speaker (and consequently within a good predictor).

In order to explore to which extent this assumption is true, we compared two different predictors: a predictor trained on single speaker data (henceforth SSP) and a predictor trained on multi-speaker data (henceforth MSP) . Both training data consists of 8954 words. SSP was trained using a subset of the f2b section of the BURN corpus, whereas the MSP training set was built by grouping all the six speakers data of section p, r and t of the multi-speaker data, so the text read by the speakers (1293 words) and the values of the training features are repeated six

	f1a	f2b	f3a	m1b	m2b	m3b	All
SSP	76.15	83.2	82.93	87.26	82.93	84.01	93.87
MSP	75.34	82.93	83.74	89.16	82.66	84.82	95.00

Table 3: *Comparison between a predictor trained on single speaker data (SSP) and one trained on multi-speaker data (MSP).*

times (one for each speakers); as a consequence only the pitch accent values vary. The section j (369 words) was held out for testing both predictors. Both predictors were trained using the Classification and Regression Tree (CART [8]) available in the Edinburgh Speech Tools Library (Wagon CART [9]). We used training features that have been proven strictly correlated to prosodic prominence: part of speech (the MXPOST tagger [10] was used), logarithm of unigram and bigram of the word. Each example consisted of the feature values of a word and of the two words preceding and following it. Unigrams and bigrams were computed on a corpus of 17 million words (Herald news from 1998 to 2002) using the CMU toolkit for language modeling ([11]). Because of the smaller lexical variability of the multi-speaker data set we did not use lexical training features, like the accent ratio feature ([12]), that would have largely favored SSP. Both SSP and MSP were tested comparing their predictions with each one of the six speakers, and with all the six speakers at the same time using the EE evaluation function.
Looking at table 3, the most evident fact, when comparing the two predictors, is that their performances are very close. Surprisingly SSP performs slightly better than MSP when tested on three of the six speakers, whereas it is worse than MSP in the all-six-speaker evaluation. There results can be interpreted looking at a CART as a list of prediction rules: we can say, with a certain degree of approximation, that during the MSP training those rules that were sensitive to speakers, that is, appropriate for describing the pitch patterns of some speakers but not for those of the others speakers, were filtered out, so only the rules that assign the non-optional pitch events were successful. If the SSP performances are very close to the MSP ones we can conclude that, at least in our prediction model, the SSP has the same ability of the MSP to distinguish between intra-speaker optional and compulsory pitch events, but this is possible if the variability (with respect to training features strictly correlated to pitch accents) "seen" by the SSP during its training phase is very similar to the intra-speaker optionality seen by the MSP.
The Wagon CART provides, along with the predicted value, the probability of all the possible values (two, in our case) of the predicted variable. In the next section we compute the entropy of each prediction from the probabilities provided by Wagon and use this entropy as a training feature (henceforth called "uncertainty") to predict pitch accent optionality.

5. Predicting Intra-Speaker Optionality

Once we have formally defined intra-speaker optionality and shown the large overlap between intra and inter speaker optionality in our prediction model, we can try to predict optionality in order to improve the prosodic realization of unit selection TTS. In [13] it has been shown that including the pitch accent feature in the target cost function improves the quality of the unit selection speech synthesis. If we were able to associate to each predicted event its degree of optionality we would be able to tune the target cost associated with the pitch accent feature in accordance to the importance (optionality) of the pitch event. Informally, the less optional the pitch event is the more selective the unit selection module should be. This approach only considers the phonological aspect of a pitch event, that is its binary value accent/no-accent; optionality could be also correlated to the phonetic realization of pitch accents and this correlation could be used to improve prosodic modeling. However in this work we do not advance this possibility.
A predictor combining the prediction of the pitch event with the prediction of its correlated optionality could be evaluated using the following formula:

$$\begin{aligned} &EVA(w_i) = 1- \\ &\lambda[(1 - P_t(pe_i))(1 - H_t(w_i))(1 - H_p(w_i))] \\ &-(1 - \lambda)[H_t(w_i) - H_p(w_i)]^2 \end{aligned} \tag{4}$$

whith $0 \leq \lambda \leq 1$.
H_t and H_p are the actual and the predicted optionality respectively.
The first term of the sum in the squared parentheses evaluates the prediction of the pitch event taking into account how this event is considered optional by the predictor and how it actually is. The product of the predicted and the actual optionality guarantees a null error when at least one of the two optionalities is 1. The second term evaluates the optionality prediction. The two evaluation are weighted by the constant λ.
We tried to predict intra-speaker optionality training and testing the Wagon CART using again the multi-speaker section of the BURN corpus: 1293 words were used for training and 369 words hold out for testing.

	A	B	C	D
Otpionality	0	0.6500...	0.9182...	1

Table 4: *Entropy values given 6 speakers. Optionality values are associated to letters. A occurs when all the speakers agree, B when only one speaker disagrees, and so on.*

Unfortunately the data available were very small, so we have to consider the results we achieved still preliminary. The training features were the same used for training the pitch accent predictors (contextual features included) plus lexical form (only if the word occurred at least five times in the training set), distance (in number of words) from the closest punctuation mark form left and from right, and the "uncertainity" of the multi-speaker pitch-accent predictor. We thought that this last feature was not only an indicator of the approximation of the multi-speaker pitch accent predictor but also a quantity correlated to the intra-speaker optionality.

Given six speakers, there are only four possible values of optionality (table 4) for each word token. We found out that, in order to improve the learning phase, considering optionality as a categorical feature and associating to each optionality value a symbol, allowed us to achieve better results. The performances of our predictor were compared with an all-non-optional baseline, which assigns a zero-value to each token (this was also the most frequent optionality value). In table 5 we show the results

	ABCD	ABD	AD
Baseline	0.3066	0.3066	0.3066
Predictor	0.2718	0.2837	0.3066

Table 5: *Error rate in predicting optionality.*

when all the four optionality values were considered (ABCD) and when the number of values were reduced. For example, observing that the C and D values are very close we grouped them together (ABD). It is interesting to note that when we considered optionality as a binary feature by grouping all the non-zero values in a single symbol (D), we were not able to improve over the baseline.

In the training phase we used the Wagon "stepwise" option that only selects those training features that give a significant contribute in the learning phase. The "uncertainty" feature turned out to be the best one. Even using it as the only feature we achieved an improvement over the baseline. We also found out that if we substituted the uncertainty of the MSP with that of the SSP, the uncertainty feature was still the best one and we were still able to improve over the baseline.

6. Conclusion and Future Works

Our work has addressed some questions concerning intra-speaker disagreement and optionality in pitch accent placement: how "diffuse" is intra-speaker disagreement? How can we evaluate a pitch accent predictor on a multi-speaker testing data set? Is intra-speaker optionality predictable? Are intra-speaker and inter-speaker optionality the same thing with respect to our prediction model? How can we exploit optionality to improve unit selection text-to-speech synthesis?

We have shown the degree of intra-speaker optionality in read speech by analyzing six speakers and then we have proposed a new definition of intra-speaker optionality associating the concept of optionality to that of entropy. This mathematical definition allowed us to formulate a new evaluation function for evaluating pitch accent predictors which we proved to be more appropriate than the evaluation functions adopted in previous works. We then compared a predictor trained on a single speaker data with a predictor trained on multi-speaker data and from the high similarity of their predictions we inferred that a large overlap between inter and intra-speaker optionality exists. Supported by this result we suggested a simple strategy to improve the performances of a unit selection speech synthesis system that includes the pitch accent feature into its target cost features. Since this approach requires optionality be predictable, we tried to predict it and we achieved successful results. However we believe there is still room to improve our results and in the future we will try to improve them using larger data sets. Moreover in our experiment we only used training features that convey general properties of words. We believe that, since pitch accents have been proven to be prosodic correlates of the informativeness and significance of words (see [13], for example), the degree of optionality of a pitch accent is strongly correlated to the informative and significance status of the word the pitch accent is assigned to. Using POS, unigrams and bigrams we access only a part of that status, since we do not take into account the context in which words are and how their information status relates with it. In future work, we will consider linguistic features describing information structure (the contrast feature, for example) that have been proven being useful in detecting "meaningful" pitch accents [15] and evaluate our approach as part of a speech synthesis system.

7. Acknowledgments

This research was funded by the Institute for Communicating and Collaborative Systems (University of Edinburgh), the College Science Fund, the Chalmers Fund and the Colin and Ethel Gordon Fund.

8. References

[1] K. Ross and M. Ostendorf, "Prediction of abstract prosodic labels for speech synthesis", Computer Speech and Language, 10:155-185, 1996.

[2] J. Bulyko and M. Ostendorf, "Joint prosody prediction and unit selection for concatenative speech synthesis", in Proc. of ICASSP 2001, Salt Lake City, USA, 2001.

[3] R.A.J. Clark, S. King, "Joint Prosodic and Segmental Unit Selection Speech Synthesis", in Proc. Interspeech 2006, Pittsburgh, USA, 2006.

[4] M. Ostendorf, P. Price, and S. Shattuck-Hufnagel, "The Boston University Radio News Corpus". Technical Report ECS-95-001, Electrical, Computer and Systems Engineering Department, Boston University, Boston, USA, 1995.

[5] J. Yuan, J. M. Brenier, D. Jurafsky, "Pitch Accent Prediction: Effects of Genre and Speaker", in Proc. Interspeech 2005, Lisboa, Portugal, 2005.

[6] E. Marsi, "Optionality in Evaluating Prosody Prediction", in Proc. Of 5th ISCA Speech Synthesis Research Workshop, Pittsburgh, USA, 2004.

[7] C.E. Shannon, "A mathematical theory of communication". Bells System Technical Journal, 27:379-423 and 623-656, 1948.

[8] L. Breiman, J. Friedman, R. Ohlsen, and C.Stone, "Classification and regression trees", Wadsworth International Group, Belmont, USA , 1984.

[9] P.Taylor, R. Caley, A.W. Black, and S.King, "Edinburgh Speech Tools Library" System Documentation Edition 1.2, for 1.2.0 15th June 1999.

[10] "A Maximum Entropy Part-of-Speech tagger" in Proc. of the Empirical Methods in natural Language Processing Conferece, University of Pennsylvania, 1996.

[11] P.R. Clarkson and R. Rosenfeld. "Statistical Language Modeling Using the CMU-Cambridge Toolkit", in Proc. ESCA Eurospeech 1997, Rhodes, Greece, 1997.

[12] A.Nenkova, J. Brenier, A. Kothari, S. Calhoun, L. Whitton, D. Beaver, D. Jurafsky, "To Memorize or to Predict: Prominence Labeling in Conversational Speech" in Proceedings of NAACL 2007, Rochester, USA, 2007.

[13] V. Strom, A. Nenkova, R. Clark, Y. Vasquez-Alvarez, J. Brenier, S. King, D. Jurafsky, "Modelling Prominence and Emphasis Improves Unit-Selection Synthesis" submitted at Proc. Interspeech 2007, Antwerp, Belgium, 2007.

[14] S. Pan, and K. McKeown, "Word informativeness and aoutomatic pitch accent modeling". Proc. of joint SIGDAT conference on empirical methods in natural language processing and very large corpora, 1999.

[15] S. Calhoun, "Information Structure and the Prosodic Structure of English: a Probabilistic Relationship", PhD thesis, University of Edinburgh, 2006.

Single Speaker Segmentation and Inventory Selection Using Dynamic Time Warping Self Organization and Joint Multigram Mapping

Matthew P. Aylett, Simon King

Centre of Speech Technology Research, University of Edinburgh
Edinburgh, Great Britain

matthewa@inf.ed.ac.uk

Abstract

In speech synthesis the inventory of units is decided by inspection and on the basis of phonological and phonetic expertise. The ephone (or emergent phone) project at CSTR is investigating how self organisation techniques can be applied to build an inventory based on collected acoustic data together with the constraints of a synthesis lexicon. In this paper we will describe a prototype inventory creation method using dynamic time warping (DTW) for acoustic clustering and a joint multigram approach for relating a series of symbols that represent the speech to these emerged units. We initially examined two symbol sets: 1) A baseline of standard phones 2) Orthographic symbols. The success of the approach is evaluated by comparing word boundaries generated by the emergent phones against those created using state-of-the-art HMM segmentation. Initial results suggest the DTW segmentation can match word boundaries with a root mean square error (RMSE) of 35ms. Results from mapping units onto phones resulted in a higher RMSE of 103ms. This error was increased when multiple multigram types were added and when the default unit clustering was altered from 40 (our baseline) to 10. Results for orthographic matching had a higher RMSE of 125ms. To conclude we discuss future work that we believe can reduce this error rate to a level sufficient for the techniques to be applied to a unit selection synthesis system.

Index Terms: speech synthesis, unit selection.

1. Introduction

Recent research in unit selection synthesis has focused on the search problem (finding the optimal unit sequence from the inventory for a target utterance), the prediction problem (how to generate natural sounding pronunciation and prosody for a given utterance in a given context), and the performance/footprint problem (how to compress ever increasing databases and how to speed up ever more complicated join and target cost functions).

However, what we call the unit inventory problem has been neglected. Current systems invariably use conventional phone inventories (although the units may be diphones, half phones, fragments of phones, etc). There remain numerous problems in current systems which we argue are caused by the use of such pre-defined phone sets.

1.1. Problems with manually-specified inventories

The single root cause of the inter-related problems described below is this: describing continuous speech as a linear sequence of phones, drawn from a relatively small and manually-specified inventory, is fraught with problems, Ostendorf's paper ”Moving beyond the ’beads-on-a-string’ models of speech” is widely cited [1].

Describing continuous speech as a sequence of non-overlapping phones is too simplistic. In reality, phones (the acoustic realisations of phonemes) are not the atomic units of speech - they are subject to variation caused by their context, and this variation is continuous in nature; in other words, when a phone varies away from its canonical form, it does not necessarily change to become the canonical realisation of a different phoneme. More often, certain aspects of the phone change (formants move, voice onset time changes, etc). A description of speech in terms of discrete phoneme categories cannot represent these changes. This is even more of a problem for casual or affective speech where prosodic reduction and prosodic emphasis further increase segmental variation.

Currently, unit selection synthesis uses a set of ad hoc heuristics to deal with problems caused by a manually-specified phone inventory. For example:

Co-articulation Arguably the biggest contribution to phone variance is co-articulation. The typical solution to this problem in speech synthesis is to use diphones to model the speech. One affect of this is to massively increase data-sparsity as we move from a typical inventory of 40 phones to around 1600 diphones. However diphones alone are not sufficient to deal with variance caused by co-articulation. The extent of co-articulation varies and can cross several phone boundaries in extreme cases. Generally a set of ad-hoc rules are added to minimise this problem, for example taking special care not to join a vowel with right ’r’ context to ones without such a context.

Vowel and consonant reduction and deletion Reduction occurs naturally and frequently throughout continuous speech. The solution often applied in unit selection is to allow a limited set of discrete pronunciation variants to model reduction and deletion. However the type of reduction and its extent is affected by speaker, speaking style and prosodic structure. Often pronunciation alternatives model this variation quite badly and can lead to errors.

Accent variation For many languages and accents there is no agreed phonetic description. Individual speakers can vary extensively. The variation can be arbitrary, context dependent, and often fundamental for conveying the character and naturalness of the speaker.

Circularity A crucial problem with the unit selection approach is that the phone inventory is used to determine sparsity and thus the text required for an audio database. Thus developers are required to create phone set inventories before having the audio data from a speaker and before encountering synthesis problems directly dependent on this data. It then becomes resource intensive to re-tune the inventory to optimise the system.

Finally, changes to the inventory have a dramatic impact on the lexicons used for synthesis and the effect of sparsity on the data. Lexicons typically contain many thousands of words and tailoring a lexicon to a specific accent is non-trivial. In turn this makes it hard to alter the phone set. Sparsity is a big problem in unit selection. The amount and type of phones present in the inventory have a dramatic effect on the sparsity. Thus to a large extent the 'ideal' phone set would be dependent on the amount of audio data available in the database. In current system the phone set is fixed no matter how much or how little data is available for a speaker.

1.2. A Machine Learning Paradigm

A separate problem arises from the requirements of so many ad hoc heuristics and so much manual intervention. It becomes impossible to caste the unit selection process into a well defined machine learning problem and thus use constraints and priors in a formalised manner.

In contrast, if the phone inventory can be determined based on a machine learning paradigm it may be easier to extend a machine learning approach throughout the system and make unit selection synthesis much more formalised and more adaptable.

2. Method

In reality, the problem we need to solve is to model the variation for a *single database only* and relate this to a lexicon which can generalise the database to speech that we wish to synthesise. In other words, over fitting a single speaker, a curse in speech recognition, is not a problem for unit selection synthesis. Thus a solution to the inter related voice building problems caused by a manually-specified phone sets can be solved by automatically learning a set of sub-word units. We term this set of sub-word units emergent phones or *ephones* as, unlike a prescriptive phone set, the ephones emerge from the occurrence of regular patterns within the data. By imposing suitable constraints on the properties of these ephones, we can ensure that the resulting set of ephones, and the corresponding ephone inventory, are optimised for use in concatenative speech synthesis.

Figure 1 gives a schematic of how this process could work. First a self organisation method is used for determining a set of ephones, *acoustic ephone selection*. The ephones are then mapped onto a lexicon to produce a *database ephone lexicon*. Phonological rules are then extracted from this database lexicon, and the relationship is generalised to generate ephone transcriptions for all words in the lexicon. The result of this process is then analysed against a set of lexical and acoustic constraints, such as the similarity between generated lexical entries and those aligned in the database, the extent minimal pairs are maintained, the extent sparsity is controlled, and, given a unit selection engine, the extent the system generates acoustic stability for joining units. The results of this analysis are then used as constraints and priors to further improve the initial acoustic ephone selection.

The work we report here is concerned only with the initial acoustic ephone selection and the creation of the initial database lexicon.

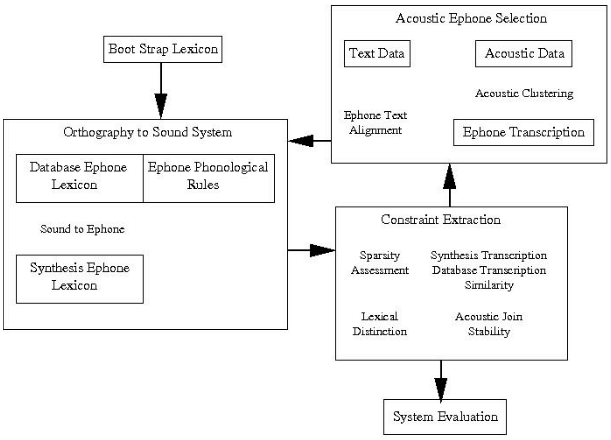


Figure 1: *A three stage machine learning process for unit selection voice building using ephones.*

2.1. Acoustic ephone selection

2.1.1. Segmentation

Automatically determining the phone set used to describe speech already has been examined, with some success, in speech recognition research (e.g [2, 3, 4, 5]), In this paper we focus on an approach using dynamic time warping (DTW) to find repeated patterns in speech and use these as ephones.

This approach is inspired by work by Park and Glass in speech pattern discovery [6]. We may regard a good unit of speech as a pattern that occurs regularly across the speech stream. A method for determining these patterns is to compare each utterances with all other utterances and find patterns that often co-occur.

Figure 2 shows how this comparison is accomplished. A full two dimensional comparison matrix is constructed with each cell containing the result of a distance calculation between every frame of speech in the first utterance and every frame of speech in the second utterance.

In the experiments reported here the speech was parametrised into 10ms frames containing 12 MFCCs and an energy component. All parameters were normalised and then the energy component was increased in size by a factor of ten. In initial studies this was found to improve the classification of silent sections of the speech. A Euclidean distance metric was used.

The algorithm then iterates down one side of the matrix and analyses the diagonal starting at this position. Three parameters are used to determine 'matching sections' within the diagonal:

1. A maximum threshold for the average comparison distance allowed over a matching segment.
2. A minimum time for a matching segment.
3. A maximum distortion allowed over the matching segment, expressed as the width W of the diagonal that the DTW algorithm is permitted to use (see figure 2).

A DTW path is computed along the permitted diagonal. Sections greater than the minimum length and with an average comparison below a threshold are then retained. We chose a a minimum length of 10ms, a maximum distortion of 210ms and

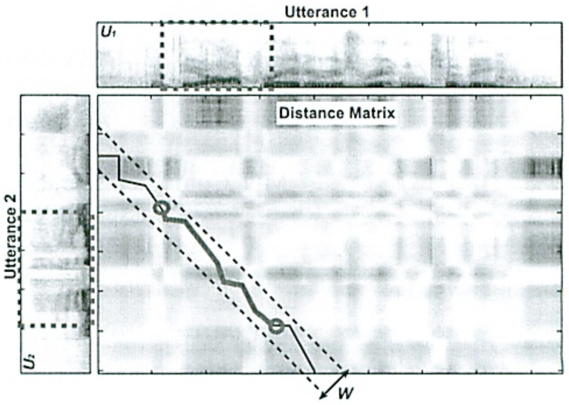


Figure 2: *Using dynamic time warping (DTW) to find co-occurring patterns in two utterances. w is the distortion allowed during the match. The bold line shows a section of the matching path where the average match is below the required threshold. (Taken from [6] p54).*

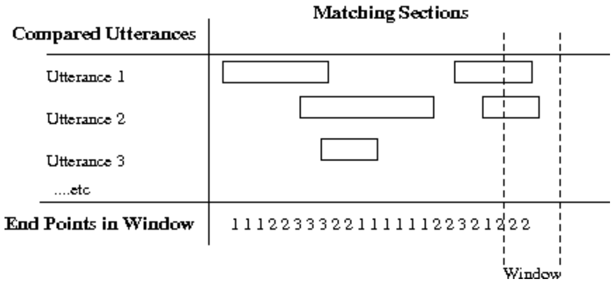


Figure 3: *The number of matching section end points are computed in a window. Maxima of number of end points present in a window are then used to place ephone boundaries.*

a maximum average comparison distance of 3.5. All sections found in this way are then written to a results file.

The results of this file were then analysed for section boundaries. A window of 50ms was passed over all sections. The number of sections starting and ending in the window were summed. This produced a parametric value that was high for frames in the speech, where matching sections terminated and others began. Figure 3 shows a schematic of this scoring process. A peak picking algorithm was then applied to this data by passing a window of 90ms across the result and placing a ephone boundary where the centre of this window was a maxima with regards to its full left and right context.

Figure 4 shows the result of this segmentation on the words 'for real change in' taken from the phrase 'we're also looking for real change in public behaviour' and comparing it to a traditional hidden markov model (HMM) segmentation carried out using HTK[7].

It is worth noting that this process is not the same as using a discontinuity metric. The matching segments may (and do) contain sharp spectral changes. However these are changes which are repeated throughout the data in similar contexts. The boundaries that this process finds are where no consistent matching sections were found. Arguably such a boundary marks a transition between matching regions and thus a location of a ephone boundary.

This segmentation process is processor intensive as it is quadratic with regards to database size. We applied this technique to a single database recorded at CSTR as part of the Festival unit selection system. The speaker was a young RP accented woman and the database we examined consisted of 728 utterances, 13k words, 60k phones, 1.8 hours of total speech and 1.38 hours of total phonetic material (total speech time with silence subtracted). This was approximately a third of the total database but is similar in size to many small unit selection databases.

To reduce computation time a reference set of utterances were selected to compare with all others. These were selected on the basis of entropy. The higher the entropy of the parameter distributions in the utterance, potentially, the more the variation within it. For example an utterance file of complete silence would have a low entropy where as an utterance of babble would have a high entropy. Utterances with the highest entropy scores and a total combined duration of not more than 200 seconds were selected as reference speech.

2.1.2. Ephone identity

In order to group segmented ephones we carried out a k means clustering using ephones as medoids. This was carried out on two numbers of clusters, 40 and 10. Once the reference data was clustered all ephones were grouped according to this initial clustering. The same dynamic time warping metric was used to compare clusters as was used initially in the segmentation.

We envisage this k-means clustering approach to be used as a baseline for further work. In further systems we expect the number of clusters to reflect the variation in the data rather than be set in advance.

Every ephone was then named according to its relationship with the baseline HMM) segmentation. For each ephone the phone that overlapped with the greatest number of frames was chosen as a name for the ephone together with the percentage of this overlap and the overall duration of the ephone in frames.

Clusters were named based on the largest set of member ephones with the same associated HMM based phone name, together with a three digit index.. The largest clusters were named first with an index of '000'. Smaller clusters with the same majority phone content were named with the phone and an incremented index.

Figure 4c shows an example of the words 'for real change in' with the ephones are labelled by cluster name. Care is required when interpreting the names of clusters. For example the first ephone '@0:002' is named as such because the majority of the ephones in the cluster mostly overlapped a '@0' (unstressed schwa) in the HMM segmentation. However this is the third largest cluster of this kind and given it contains unvoiced frication suggests it represents mostly elided schwas with heavy contextual frication.

2.2. Initial database lexicon

The creation of the ephone inventory is completely driven by bottom up processing. In order to carry out synthesis with the ephones we need to relate sequences of ephones to the words we wish to synthesise. These words can be regarded as a string of symbols. Given the vagaries of English spelling it was decided to use two alternative sequences: 1) The lower case letters themselves without hyphens, apostrophes or capitalisation.

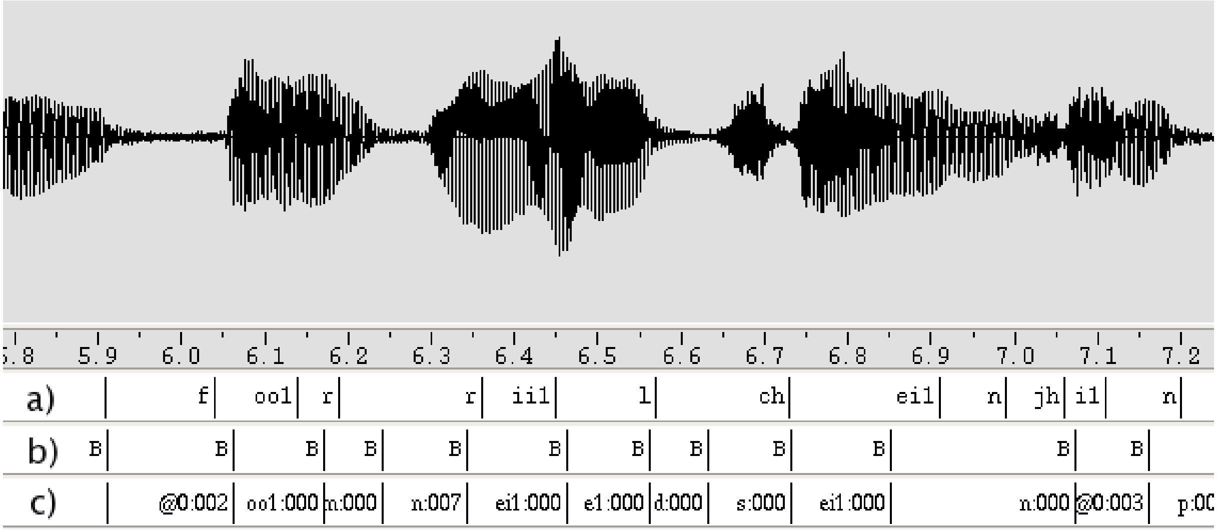


Figure 4: *Example of a) segmentation carried out by HTK b) boundaries proposed by DTW segmentation c) ephone names proposed by clustering. The sequence is "for real change in" taken from the phrase "but we're also looking for real change in public behaviour".*

2) The pronunciation of the word in terms of phone sequences from the traditional HMM segmentation. The phone set acts as a 'best case' baseline in that the acoustics of the words should relate more closely to the phone series than the letter series.

In this study we used silence detection and location information from the traditional HMM segmentation to reduce the degrees of freedom within the system. In the long term, silence insertion will need to be modelled in any sequence matching system.

We applied a joint multigram approach to matching sequences together based on work by [8]. We chose a joint multigram formalism because it allows multiple to multiple matching between letter/phone sequences and the emergent phones.

The multigram model was originally developed by Bimbot et al in order to model variable length regularities within streams of symbols hence the term *multigram* as opposed to *n-grams*. The joint multigram [9] relates two multigrams from separate streams and can be used to segment two streams into concurrent multigrams.

See [8] for a full description. Briefly, the probabilities of each multigram are recalculated based on a set of co-occurring streams using expectation maximisation. Observed probabilities are calculated using the forward backward algorithm. For example Table 1 shows the result of this process when applied to the problem of segmenting letters and phones. The result is to split the phones and the orthography into morphologically appropriate sequences.

The process for matching letter or phone sequences and ephones is made more difficult because the sequences are much longer and thus the number of possible multigram segmentations can be very large. Table 2 shows the result of applying the joint multigram algorithm to the speech in figure 4. The word boundaries are, in most part, the closest ephone boundaries to the phone word boundaries, except where the ephone n:000 at the end of the word 'change' has been co-segmented with the 'i' in the word 'in'. These types of co-segmentation error could have a serious impact on synthesis quality using this segmentation.

Table 1: *Using joint multigrams to co-segment letters and phones in a pronunciation dictionary. (MRPA phone set)*

Word	Pronunciation.	Letter Sequence	Phone Sequence
accompany	@ k uh m p @ n ii	ac com p any	@ k uh m p @ n ii
accomplice	@ k o m p l @ s	ac com pl ice	@ k o m p l @ s
accomplish	@ k o m p l i sh	ac com pl ish	@ k o m' p l i sh
accounts	@ k au n t s	acc oun ts	@ k au n t s

3. Results

Although the traditional HMM segmentation suffers from many of the problems we are expressly trying to address with the techniques described here, it can still act as an effective means of evaluation. Although we would not expect a perfect ephone segmentation to match boundaries in a traditional segmentation we would not expect boundaries to be grossly different in many locations. This is especially true at word boundaries.

If we compare the closest ephone boundary to each word boundary in the HMM segmentation the root mean square error (RMSE) of this comparison is 35ms. Thus 95% of all boundaries in this best case comparision are within 70ms of the traditional HMM word segmentation. Currently, without a perceptual test, we do not know whether the HMM boundary or the ephone boundary is correct and given this uncertainty such a

Table 2: *Using joint multigrams to co-segment a phone sequence and an ephone sequence. See figure 4 to compare word end times to the HTK segmentation.*

Word	Phone	EPhone
for	f	@0:002
	oo	oo1:000
	r	m:000
real	r	n:007
	ii	ei1:000
	l	e1:000
change	ch	d:000
	ei	s:000
	n_jh	ei1:000
in	i	n:000
	n	@ 0:003

word boundary error may be acceptable. However in our final system we will not have an HMM segmentation, instead, as we have described in the previous section, we will need to map our units onto a series of symbols, such as orthography, that represents the speech contents.

A means of evaluating the symbol mapping process is as follows:

- Use the HMM phone symbols as a representation of the speech.
- Map these phone symbols onto the ephones.
- Compare the location of the mapped phones with the HMM segmentation (especially at word boundaries).
- If sequence matching is effective we would hope that the error between the mapped phones and the HMM boundaries would approach an RMSE of 35ms which is the best match we could hope for given the ephone segmentation we have produced.

This process can then be compared with the same mapping algorithm but instead applied to orthographic information. By comparing the errors we can assess mapping algorithms, the differences between orthography and a traditional phone set, and the effects of cluster identity. We report results on the following conditions:

1. Matching orthography against traditional phone sequences.
2. Using ephones constructed with 40 and 10 clusters.
3. Varying the multigrams allowed. For example we can describes a joint multigram as *1-1*, where one symbol only matches one ephone, or *2-1* where two symbols match one ephone and so on. The ratio of phones to ephones and letters to ephones is respectively 1.4 and 1.9. Therefore a mixture of 1-1 and 2-1 multigrams are the minimum types required to allow a match between sequences. We then added further multigram types to see if this increased or decreased word boundary error.

Table 3 shows results for all phone boundaries for the phone matching conditions and for word boundaries for all conditions.

4. Discussion

The sequences we are trying to co-segment are quite long compared to word/pronunciation sequences shown in table 1. The average length of each speech chunk separated by silence is 22 ephones (standard deviation = 13). Segmenting the words to within 100 to 200ms would be regarded as quite good for say a search application, especially given no phone model is required. However the results from the joint multigram co-segmentation are significantly worse than the best case of matching closest ephone boundary to closest word boundary. In addition this granularity is too poor for unit selection synthesis where an error of much more than a phone size will cause the addition of unwanted acoustics or the loss of required acoustics.

Table 3: *Root mean square error (RMSE) between word boundaries proposed by an ephone segmentation and a baseline HTK segmentation. Multigram types are expressed as [no. symbols]:[no. of ephones]*

clusters: 40, Phones, Multigrams 1-1, 2-1 All Boundaries: *RMSE 104ms* Word Boundaries: *RMSE 0.102ms*
clusters: 10, Phones, Multigrams 1-1, 2-1 All Boundaries: *RMSE 126ms* Word Boundaries: *RMSE 124ms*
clusters: 40, Phones, Multigrams 1-1, 2-1, 1-2, 2-2 All Boundaries: *RMSE 116ms* Word Boundaries: *RMSE 109ms*
clusters: 40, Letters, Multigrams 1-1, 2-1 Word Boundaries: *RMSE 126ms*
clusters: 40, Letters, Multigrams 1-1, 2-1, 1-2, 2-2, 3-1 Word Boundaries: *RMSE 131ms*

As expected using a lower cluster size for the ephones resulted in worse performance. However the additional multigram types, for example 1-2, 2-2, 3-1 for orthographic mapping, reduces the performance. We believe there may be two reasons for this:

1. The extra multigram types are over fitting and the data.
2. The lack of a duration penalty. A 2-2 letter to ephone match is not regarded as having an intrinsic cost for crossing 2 boundaries. Thus in most cases longer multigrams are selected over shorter multigrams. This in turn contributes to data sparsity and poor co-segmentation.

However we believe the use of word boundary as an evaluation metric will allow us to improve the co-segmentation, perhaps with the addition of priors relating duration to the multigram identity. If the co-segmentation is improved it then becomes possible to improve the self-organisation and clustering approach to the acoustic segmentation.

This work is still in its early stages. Currently a set of engineering decisions have been made purely to generate a working baseline and a working evaluation of this baseline. Although the segmentation may not be ideal, it is the ephone identity derived from the clustering process and the sequence matching between these derived ephones which requires most improvement. We, believe, with the use of an automatic evaluation criteria that these processes can be improved. In future work we expect to consider ergodic HMMs as a clustering process, using Bayesian information criteria (BIC) to select cluster sizes and number, and looking more deeply into the effect of the parameters used in the current model on the segmentation and inventory selection.

5. Acknowledgements

Support for this research was provided by EPSRC (award number EP/D058139/1).

6. References

[1] M. Ostendorf, "Moving beyond the 'beads-on-a-string' model of speech," in *IEEE ASRU*, 1999.

[2] T. Holter and T. Svendsen, "Incorporationlinguisticknowledge and automatic baseform generation in acoustic subword unit based speech recognition," in *Proceedings of Eurospeech 97*, 1997, pp. 1159–62.

[3] M. Bacchiani and M. Ostendorf, "Joint lexicon, acoustic unit inventory and model design," *Speech Communication*, vol. 29, pp. 99–114, 1999.

[4] R. Singh, B. Raj, and R. Stern, "Automatic generation of sub-word units for speech recognition systems," *IEEE Transactions on Speech and Audio Processing*, vol. 10(2), pp. 89–99, 2002.

[5] S. Chen and P.S.Gopalakrishnan, "Clustering via the bayesian information criterion with applications in speech recognition," in *ICASSP 98*, 1998, pp. 645–8.

[6] A. Park and J. Glass, "Towards unsupervised pattern discovery in speech," in *IEEE ASRU*, 2005, pp. 53–58.

[7] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland, *The HTK Book*. Entropic, 1996, version 2.00.

[8] S. Deligne and F. Bimbot, "Inference of variable-lengthlinguistic and acoustic units by multigrams," *Speech Communication*, vol. 23, pp. 223–241, 1997.

[9] S. Deligne, F. Yvon, and F. Bimbot, "Variable-length sequence matching for phonetic transcription using joint multigrams," in *Eurospeech*, vol. 3, 1995, pp. 169–172.

How (Not) to Select Your Voice Corpus: Random Selection vs. Phonologically Balanced

Tanya Lambert §, *Norbert Braunschweiler* ‡, *Sabine Buchholz* ‡

‡ Speech Technology Group, Cambridge Research Laboratory,
Toshiba Research Europe Ltd., Cambridge, United Kingdom

tlambert@freeola.net, {norbert.braunschweiler, sabine.buchholz}@crl.toshiba.co.uk

Abstract

This paper compares the effect of two different voice corpus selection methods on the overall quality of unit selection-based text-to-speech (TTS) voices resulting from training on these corpora. The first selection method aims to maximize the coverage of stressed as well as unstressed diphones (phonologically balanced: *Phonbal*) while the second method simply selects sentences at random (*Random*). We show that, as expected, the *Phonbal* method results in better phonetic and phonological coverage for the training as well as unseen test sentences. However, we also provide evidence from an objective evaluation and a subjective listening test that the *Random* method results in an overall better voice quality when only automatic corpus annotation tools (such as forced alignment) are used, and potentially even with manual annotation. This result has general implications for the fast creation of TTS voices.

1. Introduction

For corpus-based text-to-speech systems, the quality of the corpus is one of the important factors of the resulting TTS voice quality. Corpus quality in turn has several independent factors: the suitability of the voice talent, the quality of the recording, the quality of the annotation, and the choice of sentences to be recorded. This paper reports about experiments and analyses concerning this last factor. Traditionally, sentences have been chosen to maximize the diphone coverage [1, 2]. Recently, this approach has been extended to the coverage of diphones in stressed as well as unstressed positions, henceforth called "lexical diphones" [3, 4]. However, it is not clear whether this approach is optimal for all types of corpus-based TTS systems. This paper presents a case study aimed at answering the following question: what is the effect of different sentence selection methods on a halfphone-based unit-selection system with fully corpus-based prosodic components when only automatic corpus annotation is used? In particular, we compare two methods: one in which sentences are sampled at random from a much larger corpus [5] and another in which sentences are chosen in order to maximize the coverage of lexical diphones. Section 2 describes the background of this work and the two methods in detail. Section 3 compares the phonetic and phonological coverage of the two sub-corpora, while Section 4 compares the sub-corpora in terms of other aspects that are important for training a TTS voice, in particular phonetic alignment and prosody. Section 5 describes the listening tests that were conducted to compare the overall quality of the voices based on the two sub-corpora. Finally, Section 6 presents conclusions and plans for future research.

2. Selection of sub-corpora

The experiments described in this paper took place in the context of the Blizzard Challenge 2007 [6]. Participants in this Challenge received the "ATR American English Speech Corpus for Speech Synthesis" [5], henceforth referred to as the *Full corpus*. It consists of utterance-length audio files totalling about 8 hours, corresponding text files and automatically created annotation. As the annotation supplied uses different conventions from what our system expects, we decided not to use that annotation but automatically created our own. See Section 4 for a summary of that method.

The *Full* corpus consists of sentences from three text genres: conversational (*BTEC*), news, and novels (*Arctic*). Upon receiving the *Full corpus*, participants had 4 weeks to create 3 TTS voices: one from the *Full corpus*, one from the *Arctic* sub-corpus, and one from a sub-corpus consisting of sentences that could be chosen freely from the *Full corpus* on condition that their total duration did not exceed the duration of the *Arctic* sub-corpus [7], which is 2,914 seconds (0.8 hours), and that the selection process does not rely on the audio files in any way. Our general system and results for the Blizzard Challenge 2007 are described in our Blizzard workshop paper [8], whereas the present paper focuses on the third voice condition only. The motivation for this third condition is to simulate the situation that one faces if one wants to record a new voice: given limited resources (e.g. budget, time) for recording, what is the best set of sentences one could record? The following two sections describe the two different corpus selection methods that we investigated: phonologically balanced versus random selection.

2.1. A phonologically rich sub-corpus

The phonologically balanced sub-corpus (*Phonbal*) was selected from the *Full* corpus using a greedy style set cover algorithm [3, 1]. This method focused on selecting lexical diphone types [3] from the *Full* corpus. A clear distinction is made between diphone types in stressed and unstressed lexical environments. For clarification, every phoneme in a phonetic string is assigned a lexical stress which it inherits from its parent syllable, e.g. /bs/ is a diphone type with no stress marking but /b0s1/ and /b1s0/ are lexical diphone types, where 0 and 1 indicate unstressed and primary stressed environments respectively.[1] The number of lexical diphone types in any text sample is much greater than the number of diphone types that are considered without stress markings. The text of the *Full* corpus was processed by Toshiba's TTS linguistic engine. Grapheme-

§ Affiliated to Toshiba when the work reported in this paper started.

[1] Secondary, tertiary and/or emphatic stress could be considered in this way as well. However, as it was not used in the experiments described in this paper, it is ignored here.

to-phoneme conversion was performed and unstressed and primary stress assigned.

The creation of the phonologically rich sub-corpus initially focused on selecting all lexical diphone types present in the *Full* corpus. Based on the phonological transcription used here it was found that the *Full* corpus contained 368,039 lexical diphone tokens and 4,332 lexical diphone types. Lexical diphones also included silences (predicted from text-based features only). There were 631 lexical diphone types that appeared once in the text, and the most frequent lexical diphone type appeared over 8,500 times. The objective of the greedy style set cover algorithm was to capture the highest number of lexical diphone types within the smallest number of sentences. The phonologically rich sub-corpus generated in this way consisted of 1,133 sentences with speech duration of just over 6,000 seconds. As this is much more than the allowed 2,914 seconds, it had to be reduced in size.

The nature of the greedy-style algorithm is to rank sentences according to their phonological richness, where the lower ranking sentences cover only one unit of interest. In this selection, the lowest ranking 594 sentences (out of the 1,133) covered only one lexical diphone of interest. These 594 sentences were then reprocessed by excluding the primary stress information from lexical diphone combinations consisting only of consonants. The reason why the stress information was sacrificed in some consonant-consonant combinations is because past research has shown that any spectral discontinuities at concatenation points in the synthesis of CC (consonant-consonant) combinations are less likely to be detected aurally than in the synthesis of VC (vowel-consonant) combinations [9, 10].

As it was believed that different intonation types were necessary for the training of data used by the TTS system, some of the lexical diphone combinations were sacrificed at the cost of (i) intonationally rich phrases and (ii) consonant clusters preceeded and followed by a silence. With respect to (i) it was ensured that there was a sufficient coverage of interrogative sentences and multisyllabic words. With regards to (ii) consonant-vowel clusters preceeded by a phonetically marked silence (e.g. /#splɪ/, /#striː/) and vowel-consonant clusters followed by a silence (e.g. /ɪmd#/, /ɪkst#/ were added to the set. It was hoped that this inclusion would enable unit selection to choose phonetically and phonologically better suited consonants when synthesizing cluster combinations (i.e. to avoid the synthesis of e.g. /spl/ by combining /s/ and aspirated /pl/] or by combining /sp/ and clear /l/). In addition, it was hoped that this inclusion would offer better coverage with respect to falling or rising prosody depending on whether such clusters are preceeded or followed by a silence.

The phonologically rich corpus contained in the end a set of 728 sentences amounting to 2,906.25 seconds.

2.2. A randomly selected sub-corpus

The second sub-corpus was generated from the *Full* corpus by randomly selecting sentences until the maximum allowed duration was nearly reached. Then, a last sentence was selected that exactly filled the remaining duration. Therefore, the total speech duration for this *Random* sub-corpus equalled the *Arctic* speech database, i.e. 2914 seconds. The corpus consisted of 687 sentences.

Table 1 shows a comparison of the footprints of the *Full* corpus and its sub-corpora (*Arctic*, *Phonbal*, and *Random*) in terms of their duration, the number of sentences, words and words per sentence, the distribution of sentence lengths and the number of various punctuation characters.[2] The *Arctic* sub-corpus by design does not contain sentences of more than 15 words, which is why the average length and the distribution of sentence lengths are so different from the *Full* corpus. The lack of questions might be due to the nature of the text genre (novels).

Table 1: *Textual and duration characteristics of the Full corpus and its sub-corpora.*

	Full	*Arctic*	*Phonbal*	*Random*
seconds	28,591.5	2,914	2,906.25	2,914
sentences	6,579	1,032	728	687
words	79,182	9,196	8,156	8,094
words/sent.	12.0	8.9	11.2	11.8
% sent. with				
1-9 words	37.7	54.9	41.0	38.6
10-15 words	27.6	45.1	18.6	26.9
>15 words	34.8	-	40.4	34.5
’?’	868	1	96	94
’!’	4	-	-	1
’,’	3,977	430	452	410
’;’	30	6	4	3
’:’	17	-	-	-

Table 2: *Unit type coverage in Full corpus and its sub-corpora.*

Unit Types	*Full*	*Arctic*	*Phonbal*	*Random*
diph.(no stress)	1607	1385	1510	1322
lex. diphones	4332	2716	3306	2735
lex. triphones	17032	7945	8716	8144
sil_CV clusters	104	42	46	43
VC_sil clusters	184	84	100	75

Among the two sub-corpora presented in this paper, *Random* is closer to the *Full* corpus than *Phonbal* in terms of average sentence length and the distribution of different sentence lengths. The greedy style set cover algorithm used to select the *Phonbal* seems to result in a greater number of short and long sentences being chosen, at the expense of the average-length ones. It remains to be investigated why this is the case. In terms of punctuation characters, *Phonbal* contains slightly more commas. This might be a side-effect of the presence of more long sentences.

3. Unit type coverage of the corpora

Table 2 shows the unit type coverage in the *Full* corpus and its sub-corpora. The distribution of unit types (diphones, lexical diphones, lexical triphones, silence_CV clusters and VC_silence clusters) is considerably smaller in the *Random* sub-corpus than in the *Phonbal* sub-corpus. In comparison with the *Arctic* speech database the random sub-corpus appears to have a better coverage of lexical diphone and lexical triphone types.

3.1. Coverage with respect to test sentences

400 test sentences provided by the Blizzard Challenge 2007 organizers were used here to objectively evaluate the phonological and phonetic coverage of the *Full* corpus and its sub-

[2] Counts for commas, semi-colons and colons are for sentence-internal ones only.

corpora. The test sentences comprised 100 sentences each from conversational (*conv*), *news* and *novel* text genres and 50 sentences each from modified rhyme tests (*mrt*) and semantically unpredictable sentences (*sus*). Figure 1 shows the coverage of diphone types (without stress consideration), lexical diphone and lexical triphone types in test sentences for each text genre. Occurrences of silence_CV clusters and VC_silence clusters in test sentences per given text genre are poor: less than 10 occurrences for silence_CV clusters types and less than 20 for VC_silence cluster types.

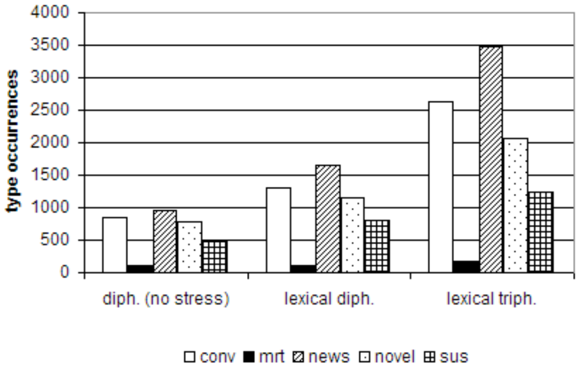


Figure 1: Distribution of diphone and triphone types in test sentences per text genre.

An analysis of unit coverage showed that neither the full corpus nor its sub-corpora contained all the lexical diphone types that were present in the 400 test sentences. Set cover mathematical operations (e.g. difference and intersection) were used on test sentences and (sub-)corpora to ascertain (i) which phonetic/phonological units were covered by both sets (Table 3), (ii) which units appeared in *Full* corpus/sub-corpora but were missing from the test sentences (Figures 2 and 3) and (iii) which units appeared in the test sentences but were missing from the *Full* corpus/sub-corpora (Figures 4 and 5).

Table 3: *Lexical diphone type coverage in the Full corpus and its sub-corpora for 400 test sentences.*

TestSent	*Full*	*Arctic*	*Phonbal*	*Random*
conv	1293	1203	1238	1212
mrt	113	111	111	108
news	1648	1518	1550	1534
novel	1147	1105	1099	1069
sus	799	741	760	739

With regard to the test sentences from novels, the *Arctic* sub-corpus appears to have better coverage than the *Phonbal* and *Random* sub-corpus. The *Phonbal* sub-corpus in comparison with the *Arctic* sub-corpus has better coverage of lexical diphone types with respect to test sentences for three text genres (for *mrt* there is a tie). In comparison with the *Random* sub-corpus, the *Phonbal* sub-corpus appears to have better lexical diphone type coverage for all five text genres.

Figures 2 and 3 show the number of diphone types that exist in the *Full* corpus and its sub-corpora but do not appear in the test sentences. For new and unpredictable test sentences, this figure indicates the phonetic and phonological richness of the given sub-corpus in relation to each text genre.

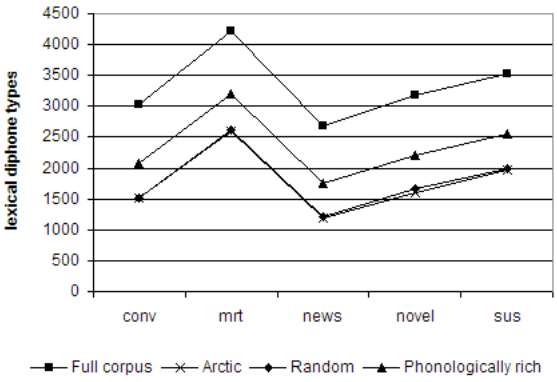


Figure 2: Lexical diphone types that appear in each (full/sub-) corpus but are missing from the test sentences.

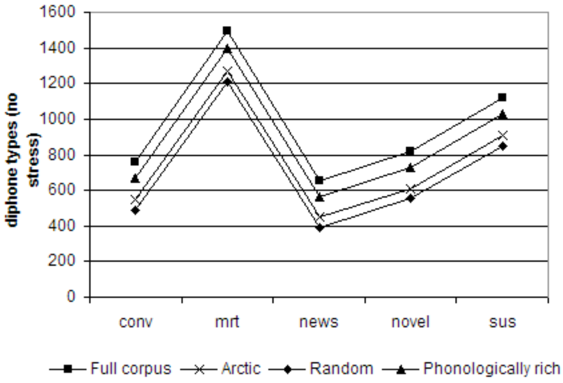


Figure 3: Diphone types that appear in each (full/sub-) corpus but are missing from the test sentences.

4. Objective evaluation

The previous section showed that the diphone and lexical diphone coverage of *Phonbal* is indeed better than that of *Random*. However, the corpus is used not only to derive the half-phones used by the TTS system but also to train its prosodic modules. The Toshiba TTS system contains a pipeline of modules that predict:

- the presence or absence of prosodic phrase breaks (chunk boundaries) [11];
- the presence or absence of pauses [11];
- the length of previously predicted pauses;
- the accent property of each word: deaccented, accented or highly accented;
- the duration of each phone;
- the pitch contour of each word.

The output of the pause, duration and pitch modules is used to restrict the unit selection (together with phonetic context and concatenation cost). If the selected units do not fulfil the target requirements of duration and pitch, they are modified accordingly. Therefore, the quality of the predicted prosody is an important factor in the overall voice quality.

All prosodic components are trained on the corpus. In the context of Blizzard, the corpus did not come with the necessary

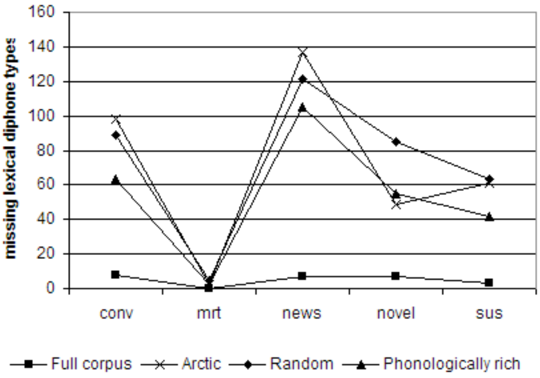


Figure 4: Lexical diphone types that are missing from each (full/sub-) corpus in relation to test sentences.

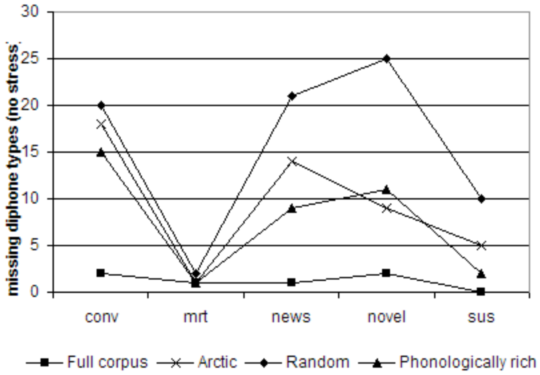


Figure 5: Diphone types that are missing from each (full/sub-) corpus in relation to test sentences.

annotations, and therefore had to be annotated automatically. First, the text was normalized (i.e. numbers, abbreviations etc. expanded). Next, likely phonetic transcriptions for each word were generated through a combination of lexicon-lookup and probabilistic post-lexical effects rules (to account for elision, assimilation etc.).[3] Then, an automatic phone aligner similar to the one described in [12] was used to perform forced phone alignment, choosing between potential pronunciation variants and allowing optional pauses at each word boundary.

As the Aligner did not use a pre-trained alignment model but rather performed a flat start [13] from the given corpus, the distribution of phones in a corpus potentially influences alignment quality. As no gold standard phonetic alignment is given for the *Full* corpus, we cannot directly measure the quality of the alignments in the two sub-corpora (*Phonbal* and *Random*). However, all other things being equal, flat starting on a larger corpus is very likely to result in better alignments than on a smaller one. It is therefore reasonable to assume that the alignments for the *Full* corpus are closer to the truth than those for the smaller sub-corpora. We therefore estimate the alignment quality of the sub-corpora by comparing them to the alignment of the *Full* corpus. We computed several metrics that have been suggested in the literature: overlap rate[4] [15], RMSE of phone boundaries,[5] and percentage of boundaries that are within certain tolerance margins of the "true" boundary. Table 4 shows the results.

Table 4: *Comparison of phone alignments in the Phonbal and Random sub-corpora against those in the Full corpus.*

Metric	*Phonbal*	*Random*
Overlap Rate	95.26	96.35
RMSE of boundaries	6.3 ms	3.3 ms
boundaries within 5ms	86.6 %	91.8 %
boundaries within 10ms	97.1 %	99.1 %
boundaries within 20ms	99.1 %	99.9 %

According to all metrics, the alignment of the *Random* sub-corpus is slightly better than that of the *Phonbal* one. When comparing the overlap rate of individual phones, a similar picture emerges. The overlap rate of most phones is better for *Random* than for *Phonbal*; in particular, the overlap rate of *all* higher frequency phones (occurring more than 800 times in the two sub-corpora) is better for *Random*. Conversely, there are only 10 phones for which *Phonbal* has a better overlap rate, all of them of lower frequency. In *all* of these cases, *Phonbal* actually contains more instances of these phones than *Random*. In general, *Phonbal* contains more instances of rarer phones than *Random*, at the expense of more frequent phones. These figures suggest that a greater phonetic coverage of a sub-corpus has a detrimental effect on alignment accuracy. Interestingly, *Random* contains fewer sentence-internal pauses (and also fewer sentence-initial and sentence-final pauses because it generally contains fewer sentences than *Phonbal*) but the overlap rate of these pauses is much better than for *Phonbal* (97.72% vs. 86.95%; for sentence-initial/final pauses: 99.73% vs. 97.27%). More investigation is needed to explain this effect. Given that the phone duration and pause models are trained using the Aligner output, we can hypothesize that training on the *Random* corpus would result in slightly better pause and duration models. In addition, units derived from the *Random* corpus should generally have better boundaries, and might give rise to fewer bad joins during synthesis.

After forced alignment, the Prosodizer [16] is used to predict ToBI markup [14] based on the phone alignments, the previously predicted syntactic annotation and F0 contours extracted from the audio files using *get_f0* from the ESPS/waves toolkit [17]. The ToBI labels are then mapped to the more coarse-grained annotation on which the chunker and the accent module can be trained. The Prosodizer operates on the sentence level, which means that the *accuracy* of this annotation should be the same for both sub-corpora (contrary to what we saw for the Aligner). However, for training prosodic modules, it is also important that the training material contains a variety of prosodic contexts. Given that this concept is difficult to define,

[3] Words that did not occur in the pronunciation lexicon were mostly transcribed manually without reference to the audio. In some cases this was not possible because the transcriber did not know the word. Sentences containing these truly unknown words were excluded from the selection sub-corpora (in accordance with the Blizzard guidelines which forbid reference to the audio for corpus selection). For the *Full* corpus, the audio was consulted to transcribe those words. This means that neither the *Full* corpus nor the sub-corpora contain words whose pronunciations had to be derived by letter-to-sound rules. Therefore the quality of the transcriptions should be relatively high.

[4] The overlap rate "is the ratio between the number of frames that belong to that segment in both segmentations and the number of frames that belong to the segment in one segmentation".

[5] excluding boundaries where the sub-corpus and the *Full* corpus have non-identical phone labels

Table 5: *Precision and recall of pauses, prosodic chunk boundaries, and accented (acc) and highly accented (high) words predicted by the prosodic modules trained either on the Phonbal or on the Random sub-corpus against the automatic markup of 1000 sentences not belonging to either sub-corpus.*

		Phonbal	*Random*
Chunks	Precision	58.9	56.3
	Recall	34.2	38.7
Pauses	Precision	63.1	63.4
	Recall	34.1	38.0
acc	Precision	69.7	69.5
	Recall	78.4	78.9
high	Precision	54.7	57.1
	Recall	38.6	41.1

we decided instead to measure the performance of the prosodic modules trained on both sub-corpora by comparing their predictions with the (automatic) annotations for 1000 sentences from the *Full* corpus which are neither in the *Random* nor in the *Phonbal* selection. Remember that the automatic annotation tools (Aligner and Prosodizer) heavily rely on the audio files, whereas the trained prosodic modules have to make their prediction from text-derived features only. It is therefore reasonable to assume that the more the predictions of a prosodic modules coincide with the automatic annotation, the better its performance.

Table 5 shows the precision and recall of (presence of) pauses, chunk boundaries, accented and highly accented words for the prosodic modules trained on each sub-corpus. For pauses and highly accented words, *Random* clearly has better performance: precision as well as recall are higher than for *Phonbal*. For chunks and normally accented words, *Random* has lower precision but higher recall than *Phonbal*. In these cases, it is unclear what the best balance between the two is. If one weighs both equally ($\beta = 1$) and computes the F-measure, *Random* has better performance (45.9 vs. 43.3 for chunks, 73.9 vs. 73.8 for accented). However, spurious chunk boundaries and accents are likely to have a bigger negative effect than missing ones, so $\beta = 1$ does probably not define the optimal trade-off. In any case, it is fair to say that some of the modules trained on the *Random* sub-corpus have a quantitatively better performance than those trained on *Phonbal*, whereas other modules are at least not clearly worse.

5. Subjective evaluation

The previous sections have shown which objective advantages and disadvantages the two sub-corpora have. However, objective metrics cannot yet replace subjective listening tests. We therefore conducted preference tests to determine which sub-corpus resulted in overall better voice quality. These tests included 53 sentences (25 relatively short declarative sentences, 11 longer sentences, 5 commands, 6 wh-questions, 6 yes/no-questions) which had been used in earlier listening tests independent of Blizzard. The sentences were synthesized with both systems and for each sentence, both samples were played one after the other. Subjects could listen to the stimuli repeatedly but were encouraged to give their answer after the first time. The order of sentences and the order of systems for each sentence were randomized for each listener. Subjects had to make a forced choice whether they preferred the first or the second sample.

Table 6: *Result of preference test comparing 53 test sentences synthesized with voice Phonbal or voice Random. Columns 2 and 3 show the number of times each subject preferred each voice.*

Subject	*Phonbal*	*Random*
Non-American Listeners		
1	20	33
2	21	32
3	24	29
4	25	28
All	90	122
American English Listeners		
1	21	32
2	21	32
3	16	37
4	23	30
5	25	28
All	106	159

In a preliminary test, 3 British and one German speech expert took part. A later, more formal test involved 5 American English speakers without specific speech technology knowledge. In the latter test, we also asked subjects to briefly write down theirs reason (if any) for each preference decision. Table 6 shows the quantitative results of both tests. Each of the 9 subjects preferred the Random over the Phonbal voice.

When comparing the preference scores for those sentences where either only *Phonbal* or only *Random* was missing a (non-lexical) diphone (6 and 9 sentences, respectively), we do observe that in general, the voice which has the diphone is preferred. However, as this effect concerns only a minority of sentences, and in any case *Phonbal* has only slightly fewer missing diphone tokens than *Random*, it does not change the overall picture.

In the future, we plan to analyze the comments by the American subjects in more detail, identify the points in the speech signals that caused them to prefer one version or the other and check whether we can trace them back to bad alignments or wrong prosody predictions.

6. Conclusions and future research

We have described the creation of two sub-corpora, a phonologically balanced (*Phonbal*) and a randomly selected one (*Random*), and have shown that listeners consistently prefer the TTS voice built with our system from the *Random* corpus. We have investigated the differences between the two sub-corpora and shown that although *Phonbal* has better diphone and lexical diphone coverage, the automatic phone alignment of the *Random* corpus is more accurate than that of the *Phonbal* one. In addition, the prosody predicted by the models trained on the *Random* corpus seems to be slightly better. We assume that these factors are at least part, if not all, of the explanation for the observed preference results.

The experiment described in this paper used a specific corpus, a specific (automatic) annotation method, and a specific TTS system. However, it is likely that other corpus-based unit-selection systems would also suffer quality losses when trained on worse alignments. This means that for the very fast creation of TTS voices, where one cannot manually correct the corpus annotations, one should seriously consider how to select the set of sentences to be recorded.

In the future, we would like to explore the following questions in more details:

- Is the better prosody prediction performance only due to better automatic prosody annotation which is due to better phonetic alignment, or is the *Random* selection inherently better suited to train prosody models on, e.g. because its distribution of sentence lengths is not as skewed as the *Phonbal* one? This question can be answered by re-doing the automatic prosody annotation of the sub-corpora, but this time using the phone alignments of the *Full* corpus as input to the prosody annotation tool, thereby eliminating any difference in alignment quality, and then re-training the prosodic modules on the two sub-corpora. If the prosody predicted by the modules trained on the *Random* corpus is then still slightly better, the difference has to be inherent to the selection method. This would mean that a *Random* selection has advantages even when manual annotation is used, as long as the TTS prosody is trained on the corpus and not rule-based.
- What exactly is the relation between phone frequency and alignment accuracy?
- Why does the *Random* corpus have so much better pause alignment when it contains fewer pauses?
- Is it worth trying to construct some kind of prosodically balanced corpus to boost the performance of the trained prosody modules, or would that result in a similar detrimental effect on alignment accuracy?

7. Acknowledgement

We gratefully acknowledge the use of the "ATR American English Speech Corpus for Speech Synthesis", kindly supplied by ATR for the 2007 Blizzard Challenge. We would also like to thank all participants in the listening tests.

8. References

[1] François, H., and Boëffard, O., "Design of an Optimal Continuous Speech Database for Text-to-Speech Synthesis Considered as a Set Covering Problem", in *Proc. of Eurospeech'01*, Aalborg, Denmark, 2001, pp. 829–832.

[2] Beutnagel, M., Conkie, A., and Syrdall, A. K., "Diphone Synthesis Using Unit Selection", in *Proc. of the Third ESCA/COCOSDA Workshop on Speech Synthesis*, Jenolan Caves, Blue Mountains, Australia, 1998, pp. 185–190.

[3] Lambert, T., "Databases for Concatenative Text-to-Speech Synthesis Systems - Unit Selection and Knowledge-Based Approach", Ph.D. dissertation, Univ. of East Anglia, 2005.

[4] Lambert, T., and Breen, A., "A Database Design for a TTS Synthesis System Using Lexical Diphones", *8th International Conference on Spoken Language Processing (ICSLP)*, Korea, 2004, pp. 1381–1384.

[5] *ATR American English Speech Corpus for Speech Synthesis*, Advanced Telecommunications Research Institute International (ATR), 2005–2007.

[6] Black, A., Tokuda, K., and King, S., "Blizzard Challenge 2007", in *Proc. of The 6th ISCA Speech Synthesis Workshop*, Bonn, Germany, 2007.

[7] Kominek, J., and Black, A. W., "CMU Arctic Databases for Speech Synthesis", Carnegie Mellon University, Pittsburgh, Pennsylvania, 2003

[8] Buchholz, S., Braunschweiler, N., Morita, M., and Webster, G., "The Toshiba entry for the Blizzard Challenge 2007", in *Proc. of The 6th ISCA Speech Synthesis Workshop*, Bonn, Germany, 2007.

[9] Syrdal, A. K., "Phonetic Effects on Listener Detection of Vowel Concatenation", in *Proc. of Eurospeech'01*, Aalborg, Denmark, 2001, pp. 979–982.

[10] Klabbers, E., and Veldhuis, R., "Reducing Audible Spectral Discontinuities", *IEEE Transactions on Speech and Audio Processing*, 9(1), pp. 39–51, 2001.

[11] Burrows, T., Jackson, P., Knill, K. and Sityaev, D., "Combining Models of Prosodic Phrasing and Pausing", in *Proc. of Interspeech*, 9th International Conference on Speech Communication and Technology, Lisboa, Portugal, 2005, pp. 1829–1832.

[12] Talkin, D., and Wightman, C. W., "The Aligner: Text to speech alignment using Markov models and a pronunciation dictionary", in *Proc. of 2nd ESCA/IEEE Workshop on Speech Synthesis*, Mohonk, New Paltz, NY, USA, 1994, pp. 89–92.

[13] Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woddland, P., "The HTK Book", (for HTK Version 3.4), Cambridge, United Kingdom, 2006.

[14] Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., and Hirschberg, J.,. "ToBI: A Standard for Labeling English Prosody", in *Proc. of the International Conference on Spoken Language Systems*, Banff, Canada, 1992, pp. 867–870.

[15] Paulo, S., and Oliveira, L. C., "Automatic Phone Alignment and its Confidence Measures", in *Proc. of Advances in Natural Language Processing*, 4th International Conference, EsTAL, Alicante, Spain, 2004, pp. 36–45.

[16] Braunschweiler, N., "The Prosodizer - Automatic Prosodic Annotations of Speech Synthesis Databases", in*Proc. of Speech Prosody*, 3rd International Conference, Dresden, Germany, 2006, PS5-27-76.

[17] Talkin, D., "A robust algorithm for pitch tracking (RAPT)", in *Speech Coding and Synthesis*, W.B. Kleijn and K. K. Paliwal, Eds., Amsterdam, The Netherlands: Elsevier Science, pp. 495–518, 1995.

Unit Selection Synthesis Using Long Non-Uniform Units and Phonemic Identity Matching

Lukas Latacz, Yuk On Kong, Werner Verhelst

Department of Electronics and Informatics (ETRO)
Vrije Universiteit Brussel
Pleinlaan 2, B-1050 Brussels, Belgium
{llatacz, ykong, wverhels}@etro.vub.ac.be

Abstract

This paper investigates two ways of improving synthesis quality: to maximise the length of selected units or to capitalise on phonemic context. For the former, it compares a synthesiser using a novel way of target specification and unit search with a standard unit selection synthesiser. For the latter, weights for phonemic context are set differently according to the distance of the phoneme concerned from the target diphone, and according to the class (consonant/vowel) to which the phoneme in question belongs. Both ways lead to improvements, at least when the speech database is small in size.

1. Introduction

Concatenative synthesis has been the mainstream way of speech synthesis for about two decades. Many speech synthesizers are based on the unit selection paradigm, e.g. [1]. In such systems, units are first selected from a reasonably large speech database, based on target specifications. A search algorithm, e.g. the Viterbi algorithm, selects afterwards the best combination of units. Optionally, one could modify the units in order to have a closer match to their target specification. Typically, the speech database contains many candidate units for a given target specification. By searching for small candidate units, the maximum number of combinations of units can be achieved. Those small units could represent phones, diphones, demiphones, etc. This is a bottom-up approach. Longer units could occur when two or more units which are adjacent to one another in the speech database are selected. We express the length of a unit as the number of diphones represented by the unit. Longer units are preferred because fewer joins are required. A join can be problematic if there is any noticeable artifact or if the two associated units are obviously different in voice quality.

Both the linguistic and prosodic contexts of the unit are important in the selection process. Due to the sheer amount of candidate units, we must be able to distinguish suitable candidates from the others. Using more context could lead to the selection of longer units. Of course, the length of a unit selected is not the only criterion in determining synthesis quality and various factors play a role.

In this paper, we propose a new target cost to capture how well a unit matches the phonemic context of the target. Instead of using only the direct neighboring phonemes of the target and the unit, we look at the bigger "picture". However, even if these wider contexts are used, this does not always result in the selection of long units. This is illustrated by an experiment in the paper.

Therefore some speech synthesizers use completely different ways of target specification and unit search, and bias longer units, e.g. [2], [3] and [4]. This results in fewer units for the same combinations compared to the bottom-up approach and much faster synthesis. Reasonably good results have been reported using these methods in so-called "limited" domains. In such domains, the text to be synthesized is limited to one particular type. Yet, the vocabulary involved could still be unrestricted. To our knowledge, the quality of those approaches has not yet been investigated in the open domain.

In this paper, we present a new way of target specification and unit search, which is also a top-down approach. It is different from the other approaches because we explicitly search for longer units based on their phonemic identity. By doing so, we aim at finding the best units efficiently.

Section 2 contains an overview of our new unit selection synthesis framework. Section 3 explains the new way of target specification and unit search and section 4 gives more details about the new target cost based on phonemic identity. We investigated the effect of incorporating a broader phonemic context in a standard unit selection synthesizer based on diphones and compared this to the experimental synthesizer which uses the new way of target specification and unit search. These are explained in section 5 and the results are discussed in section 6. Finally, we present our conclusion and possible improvements in section 7.

2. The SPACE synthesizer

The SPACE synthesizer is new and developed as part of the SPACE project. SPACE stands for "SPeech Algorithms for Clinical and Educational applications". Part of the aim of this research project is to build a Dutch speech synthesizer with high-quality output and extra synthesis options to be incorporated into a reading tutor for treating dyslexic children. The SPACE synthesizer is corpus-based. It features a unit selection framework, which allows the implementation and evaluation of different unit selection algorithms. These can be implemented in either Scheme, the scripting language used by the Festival environment [5], or C++. The linguistic and prosodic processing of the input text is currently provided by NeXTeNS [6], which is an open source Dutch synthesizer based on Festival.

As the application is meant for children's therapy, it is a limited-domain synthesizer for children's stories. Although the vocabulary size of the domain is unlimited, certain words or phrases could occur more frequently than in another domain, e.g., news. Therefore, the speech database contains story material at different complexity levels (about 3 hours of speech) in addition to all Dutch diphones (about 2000), which serve as the back-up. AVI Levels [7], the complexity scale used, vary from one to nine, and are based on the average sentence length, the average word length, word types, etc., and the suitability of a text for a particular child. For the experiment in this paper, only the AVI1 part of our story database and diphones are used. Some utterances in the AVI1 part of our database are:

- *met die kam en die zeep*. (English: *with that comb and that soap*)
- *er zit een buis in mijn haar*. (English: *there is a tube in my hair*)
- *maar in dat oor van suus wil ik ook wel zijn.* (English: *but I would also like to be in that ear of suus*)
- *dat is juist leuk.* (English: *that is what makes it fun*)

2.1. Unit selection framework

Different unit selection algorithms are implemented as different synthesis options in the SPACE synthesizer. The following options are currently available: diphone synthesis [8], "standard" unit selection synthesis (explained below), and our unit selection synthesis algorithm (experimental option) which is explained later. The diphone synthesis option synthesizes an input text by combining single diphone candidates as required and there is no selection involved. The standard unit selection synthesis option evaluates possible combinations of candidate units which are either diphones or phones and selects the best combination using a cost function based on both target and join costs. Within this framework, the different synthesis options can share part of or the whole speech database, and also the selection cost function and the associated implementation if necessary.

In general, unit selection synthesis constructs so called “targets” based on the linguistic and prosodic analysis of the input text. Selection is based on the features of each target. The unit selection framework allows the use of heterogeneous targets, i.e. targets based on linguistic units of different lengths or targets with a different set of features.

The cost function c(u_1, u_2,…, u_n,, t_1, t_2, …, t_n) is used to calculate the cost for selecting a sequence of n candidate units u_i, with their corresponding targets being t_i, based on k target costs c_j^{target} and m join costs c_j^{join} :

$$\mathrm{c}(u_1,u_2,\ldots u_n,t_1,t_2,\ldots t_n) =$$

$$\alpha * \sum_{i=1}^{n} \frac{\sum_{j=1}^{k} w_j^{\text{target}} c_j^{\text{target}}(u_i,t_i)}{\sum_{j=1}^{k} w_j^{\text{target}}} + \sum_{i=1}^{n-1} \frac{\sum_{j=1}^{m} w_j^{\text{join}} c_j^{\text{join}}(u_i,u_{i+1})}{\sum_{j=1}^{m} w_j^{\text{join}}} \qquad (1)$$

The weight α allows the fine-tuning between join and target costs. Weights w_j^{target} and w_j^{join} are set manually. The cost function is minimized by applying the Viterbi algorithm. Notably, if two candidate units happen to be from neighboring units in the database, all join costs would be zero.

3. Searching units using phonemic identity matching

We propose a new unit selection algorithm based on phonemic identity matching which favors longer units (as implemented in our experimental synthesis option). This results in the selection of non-uniform units from our database. The explicit selection of longer units reduces the number of joins and hence probably that of bad joins.

But, of course, the prosody of the units and the quality of the joins are also important. Selection is therefore still based on a target and join cost formulation as in a standard unit selection synthesizer.

Our system could be considered a "pure" unit selection synthesizer since the prosody of the selected unit is not modified. Modification is applied only at boundaries when units are joined by the pitch-synchronous concatenation algorithm described in [8]. The natural prosody from the speaker is maintained within a unit.

In our case, the smallest unit possible is a diphone. Since we have recorded all Dutch diphones in carrier phrases, we can always find a particular diphone in the database as the last resort. If this is not the case, we could opt for a back-off procedure as, for example, in the Multisyn synthesizer [9]. We choose diphone as the basic unit to capture phone transitions. However, the algorithm can easily be adapted for other small basic units, such as phones and demiphones.

3.1. Biasing long units

The idea of biasing longer units is not new, as mentioned before. Even in a standard unit selection synthesizer, long units can easily be favored by the use of an *adjacency* cost. Such a join cost measures whether two units are consecutive in the speech database:

$$\mathrm{c}_{adjacency}(u_1,u_2) = \begin{cases} 0, \text{ if } \mathrm{u}_1 \text{ and } \mathrm{u}_2 \text{ are adjacent in the speech database} \\ 1, \text{ otherwise} \end{cases}$$

It is a join cost since it gives an estimate as to how well consecutive candidate units match each other. It is used in many speech synthesizers. By setting a high weight to this cost compared to the weights of other costs in the system, the selected sequence of units would often show smaller number of joins. However, the costs for all possible combinations still have to be calculated although many of these combinations will not be selected anyway due to the high weight of the adjacency cost. More importantly, we do not know for sure if the selected unit sequence is indeed one with fewer joins. Weights are relative to one another after all.

Several other approaches were proposed featuring longer units. In [2], Taylor and Black constructed a phonological tree. Units have to match part of the tree to be selected. Another approach is to build a so-called multi-level tree as in [3]. Most approaches, however, do not consider the fact that co-articulation does not stop at word or syllable boundaries. This sets our approach apart from them. Another difference is that we do not explicitly search for individual linguistic units such as words or syllables, but achieve this implicitly by searching for the phonemic representation of the text instead. This contrasts with, e.g. [3]. We opt to use canonical phonemic transcription to label our database. In this way, we can by-pass problems caused by reduced speech at high speech rate, etc.

The approach most related to ours is described by Yang et al. in [4]. Their approach selects long non-uniform units consisting of one or more (adjacent) phoneme units. In our case, these units consist of one or more (adjacent) diphone units. Other differences are that units are not clustered and that there is no maximum unit length in our system.

3.2. Unit selection algorithm

As mentioned before, in our experimental synthesis option, we wish to select long sequences of diphones consecutive to each other in the database because this results in the selection of long units. The only criterion used in selection is phoneme identity.

Based on the linguistic and prosodic processing of the input text, our system generates a sequence of target diphones. Each phone of the target diphones is labeled with features required for target cost calculation and selection. Although other features than phoneme identity could be used, such as stress/unstressed, we opt to use phoneme identity only so as to maximize the number of possible candidates.

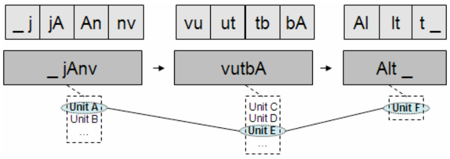


Figure 1: Selecting the longest sequence of diphones starting from the left. The utterance “jan voetbalt” (English: “Jan plays football”) is synthesized. Note that units could correspond to more than one target diphones.

The next step involves the selection of candidate units from the speech database. The complete inventory of units is used. As we intend to select longer units explicitly, each candidate unit corresponds to one or more target diphones, as can be seen in figure 1. The selection process is illustrated in figure 2. First, we search in the database for units matching the first target diphone. This results usually in a very large number of possible candidate units. Next, we prune these results and keep only the units which have a neighboring diphone in the database corresponding to the second target diphone. This results in longer units matching two adjacent target diphones. This process continues until the longest possible unit is found. If there is still any unmatched target diphone, the search starts again to select candidate unit/units matching the unmatched diphone/diphones.

The algorithm described above can lead to the minimum number of joins. However, longer candidate units tend to be fewer in supply. This could lower the number of possible combinations for selection. Potentially, this could lead to poor join quality or prosody. Therefore, we propose not to use the longest possible candidate unit but to use slightly shorter ones. Each time after finding the longest possible matching unit, we backtrack and select units which match a smaller number of target diphones. In most cases, this should result in more candidate units since probably more units would match the shorter target diphone sequence. We choose to stop the target unit sequence right after reaching the last syllable boundary of the longest possible candidate unit. This means that the last diphone of the target unit contains this particular syllable boundary. (Note that syllable boundaries are given by the target specification.) If the longest possible candidate unit does not contain any syllable boundary, we do not reduce the length of the unit. By stopping after the first syllable boundary, the risk of getting noticeable artifacts is lower as this keeps syllables together as far as possible. An alternative could be to always use a fixed number of diphones less than the number of target diphones matching the longest possible units found.

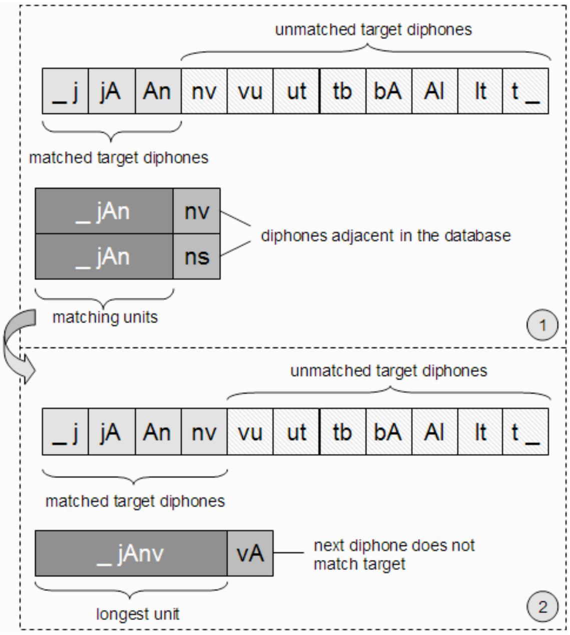


Figure 2: Illustration of the unit selection procedure

After all the target diphones of the input text have been covered by at least one unit, the best unit sequence is selected. This is illustrated in figure 1.

Sample syntheses can be found on our website http://www.etro.vub.ac.be/Research/DSSP/Demo/SSW6.htm.

3.3. Cost functions

To test the performance of our unit selection algorithm, we use only a limited set of target and join costs for both the standard unit selection and experimental synthesis options. More advanced costs can, of course, be used. They probably would improve synthesis quality but could also make it harder to compare algorithms as these could minimize the differences amongst syntheses from different algorithms. Only one target cost is employed in order to highlight differences, namely the one for phonemic context described below (section 5). As for join costs, these are used in our experiment:

- Euclidean distance between MFCCs (12 coefficients including the first one)
- Absolute difference in F0 (logarithmic). If the phone at the join position is voiceless, this cost is 0.
- Absolute difference in energy on either side of a join.

- Adjacency cost, as explained above.

4. Target cost based on phonemic identity matching

Diphones are often used as the basic unit for speech synthesis because they capture the transition at the boundaries between neighboring phonemes. Phonemes are not static re-usable templates of speech. Instead, depending on the identity of its neighbors, a particular phoneme is modified slightly. But such a process, or co-articulation, may last further than just the immediate neighbor.

While investigating the effect of a wider context of phonemic identity, actually the exact neighboring syllables, words and phrases are implied. As a result, the prosody associated with them is implied as well. Since prosody is difficult to model, this potential additional benefit could be crucial to quality.

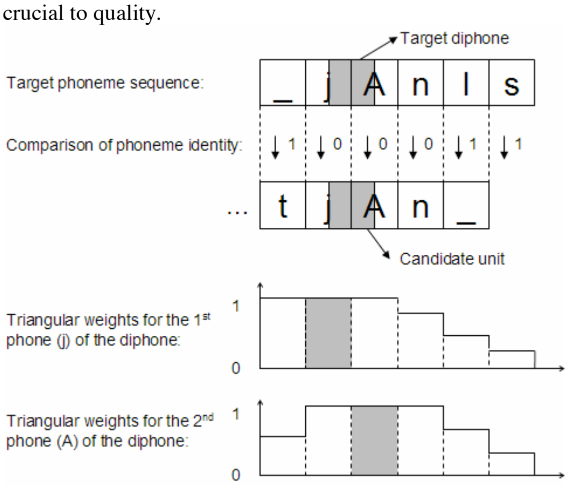

Target diphone
Target phoneme sequence:
_ j A n I s
Comparison of phoneme identity:
1 0 0 0 1 1
... t j A n _
Candidate unit
Triangular weights for the 1st phone (j) of the diphone:
1
0
Triangular weights for the 2nd phone (A) of the diphone:
1
0

Figure 3: Illustration of the use of an extended phonemic context with "triangular weights" (weights decreasing with the distance from the target diphone).

5. Experiment

The only target cost used in our experiment deals with the extended phonemic context of a target diphone. A pilot experiment is conducted to investigate how important the phonemic context at different distances from the target diphone is to synthesis. To do so, we assign either the same weight or different weights for the extended phonemic context cost to phonemes at different distance from the target diphone. In our design, we have 3 cases. In the first case, the same non-zero weight is assigned to the phonemes immediately next to the target diphone on either side only. Zero weight is assigned to all other phonemes within the same utterance. In the second case, the same non-zero weight is assigned to all phonemes within the utterance. In the last case, the further away a phoneme is from the target diphone, the lower its assigned non-zero weight is (Figure 3).

To investigate whether the class of a phoneme (consonant/vowel) would affect the importance of the phonemic context to synthesis, the weights for the extended phonemic context are manipulated depending on the nature of the phoneme concerned. In our design, we have 3 cases. In the first case, the same baseline as above is used for comparison. In the second case, non-zero weights are assigned to all phonemes within the utterance and there is no difference whether the phoneme in question is a consonant or a vowel. In the last case, non-zero weights are also assigned to all phonemes within the utterance but the weight is doubled if the phoneme in question is a consonant. The weight for silence remains the same for all cases.

While the above two independent variables are separate theoretically, in practice there is a shared baseline and the different phonemic context target cost settings are derived by crossing these two independent variables. The details would be explained (section 5.1.4).

To compare syntheses from the above phonemic context target cost settings, and to compare the experimental synthesis option with the standard unit selection synthesis option of the SPACE synthesiser, the same set of sentences are synthesised in each case while other parameters are kept the same.

5.1. Procedures

5.1.1. Subjects

As this is a pilot experiment, there are 5 subjects altogether, all working in our department. They all appear to have normal hearing, good general health and normal intelligence. They are also native Dutch-speakers and naive in the sense that they do not know what has been manipulated & what exactly we are investigating.

5.1.2. Environment and Equipment

The experiment is carried out inside a quiet office. The sound files are stored in a computer. Stimuli are listened through headphones of the same model (Sennheiser HD555).

5.1.3. Presentation

The sound files are imported to a word document in the form of a table. Each row contains files synthesised from the same sentence and each column files from the same synthesis option or under the same phonemic context target cost setting from the standard unit selection synthesis option. However, columns are labeled only alphabetically instead of with the respective synthesiser or phonemic context target cost setting. Also, the columns are not arranged sensibly according to the types of synthesis option or phonemic context target cost setting. Instead, they have been randomised. Therefore, the subjects do not know anything about the source of the files other than that they are syntheses. They do not know whether files in each column share the same source either. All subjects respond to the same document.

The subjects can click to listen to each synthesis file as many times as they like. They can adjust the volume to a level which is loud enough and comfortable. The subjects are asked which synthesis version they prefer and instructed to score each with an integer from 0 to 10, with 0 being the worst and 10 being the best. There can be ties between versions.

There are two anchors in this experiment, namely files from diphone synthesis [8] and natural recording. Sound files from these sources have pre-assigned ratings of 3 and 9 respectively and serve as references for getting more reliable ratings.

Each subject should finish rating all files within a single session, with a short break in the middle if needed. There is no time limit for the session.

5.1.4. Stimuli

Most synthesised speech comes from the standard unit selection synthesis option of the SPACE synthesiser. For this synthesis option, the weight for the phonemic context target cost is manipulated so to have the following 5 phonemic context target cost settings (by crossing the two independent variables described above):

1. baseline
2. fixed weight for all
3. weight decreases with the distance from the target diphone
4. as in (2) but weights for consonants are doubled
5. as in (3) but weights for consonants are doubled

Comparison between (2) and (3), and between (4) and (5) should shed light on whether the weight should decrease with the distance from the target diphone. Similarly, comparison between (2) and (4), and between (3) and (5) should tell us if consonants should be given higher weights than vowels.

In order to make sure that we have perceivable differences among the stimuli from the different phonemic context target cost settings, we performed some pre-trials and set weights to balance the effects from costs that were inherently large in value.

Altogether 10 sentences are selected randomly from AVI1 story material for synthesis in each case. None of them is in the speech database of the synthesiser. Otherwise, unusually long units or even the whole utterance can get "selected" by some synthesis option or phonemic context target cost setting and this would obviously affect comparison. Some of these 10 sentences are:

- *'waar doet het pijn?' zegt mam*. (English: *'where does it hurt?' says mom*)
- *dat haar is niet goed voor je*. (English: *that hair is not good for you*)
- *in die hoek ligt een pop*. (English: *a doll lies at that corner*)
- *of ik schuil in haar oor*. (English: *or I could hide in her ear*)
- *hij rent van hier naar daar*. (English: *he runs from here to there*)

Sentence lengths are limited to 6-10 words. They should not be too short because there has to be enough to listen to for making a judgement and should not be too long because otherwise the listener cannot remember and compare them.

Besides, the same 10 sentences are also synthesised with the experimental synthesis option (our new unit selection algorithm based on phonemic identity matching, which favors longer units) under the same conditions (for features, weights, etc.) and under the baseline condition (phonemic context target cost setting) in order to compare that option with the standard unit selection synthesis option. This is our stimulus (6)

The same is also performed using the diphone synthesis option. These syntheses, together with the corresponding natural recordings, serve as anchors (stimuli (7) and (8)). Altogether 60 stimuli need to be scored. With the anchors, each subject has to listen to 80 utterances.

6. Results and Discussion

The results of the listening test are presented in Table 1. One-way ANOVA (Analysis of Variance) is conducted to test for differences in the perceived synthesis quality among the synthesis options and phonemic context target cost settings (Table 2). The results do not show any significant difference among phonemic context target cost settings 2-5. The perceived synthesis quality from these 4 settings is not different statistically from the experimental option either. However, both settings 2-5 and "experimental" are different significantly from the baseline setting.

listener	1	2	3	4	5	mean
setting 1	5.4	5.0	5.6	5.1	5.4	5.30
setting 2	6.5	5.3	6.6	6.0	6.2	6.12
setting 3	6.3	5.4	6.0	6.0	6.2	5.98
setting 4	6.7	5.2	6.7	6.0	6.0	6.12
setting 5	6.3	5.4	6.2	6.1	6.4	6.08
experimental	7.0	5.5	6.6	6.4	7.0	6.50

Table 1: Results of the listening experiment. Values are mean rating scores on 10 synthesized sentences

Comparison	F
settings 1-5, experimental	3.382083*
settings 2-5	0.093605
settings 1 (baseline) & 2-5	3.106747*
settings 1 (baseline) & 2	10.28135*
settings 1 (baseline) & 3	12.7033**
settings 1 (baseline) & 4	7.521253*
settings 1 (baseline) & 5	14.01843**
setting 1 (baseline) & experimental	16.36364**
settings 2-5 & experimental	0.75019
setting 2 & experimental	1.11592
setting 3 & experimental	2.693227
setting 4 & experimental	0.94133
setting 5 & experimental	1.642458

Table 2: ANOVA on listening test results. Note that * means significant difference (p=0.05) while ** means significant difference (p=0.01). Other apparent differences are not significant statistically.

In other words, the various phonemic context target cost settings of the standard unit selection synthesis option perform better than the baseline. Widening phonemic context does bring about improvement. But giving extra weights to consonants does not cause any noticeable change. Setting uniform weights gives about the same performance as decreasing weights with distance from the target diphone. The results also show that the experimental algorithm and widening phonemic context lead to the same extent of improvement, given the other conditions that we have. It is worth noting that all mean ratings lie around the mid-point between the two anchors.

To further investigate, we calculate the mean unit lengths of different types of syntheses as shown in table 3. As expected, the mean unit length found in the syntheses from the experimental synthesis option is almost double that from the standard unit selection synthesis option (phonemic context target cost setting 1) while the same measurements found in the syntheses from other phonemic context target cost settings are only slightly longer than that from the latter and are about the same in values among themselves. In fact, when the selected units of the latter 4 settings were compared, they showed high levels of overlap. Therefore, these settings do not cause many differences among themselves.

As 10 sentences is a small number, we synthesised 30 additional sentences under the same conditions. The same pattern emerged (table 3).

	10 sentences for listening test	30 additional sentences
setting 1	1.65	1.55
setting 2	1.93	1.68
setting 3	1.94	1.69
setting 4	1.91	1.65
setting 5	1.93	1.66
experimental	3.12	3.06

Table 3: Mean length of units found in syntheses (in number of diphones)

By assigning weights to all phonemes within the utterance being synthesised is like targeting not just for a diphone but one which is surrounded by exactly the required phoneme sequences on either side. It is like targeting for the diphone within the right syllable, the right word, or even the right phrase or utterance.

The results suggest that consonants and vowels are equally important in terms of their contribution to the wider phonemic context for higher synthesis quality.

They also suggest that as long as the phonemic context is widened, there would be improvement. It does not matter if weights stay the same or taper off along the utterance. This seems against intuition and deserves further investigation.

7. Conclusion

Our new way of target specification and unit search, as implemented in our experimental synthesis option, was found to select units which are longer on average for synthesis. It also performs better than standard unit selection as implemented in our standard unit selection synthesis option, probably as a result of the longer mean unit length of syntheses and the potentially more natural prosody which may come along with that.

Widening phonemic context in some way can also lead to synthesis quality improvement. But the conditions that we investigated into, namely uniform/tapering weights along the utterance and differential weights based on phoneme identity (consonant/vowel), do not cause any difference.

It should be noted that searching for wider contexts is not the same as searching explicitly for long target strings. In our experimental option, consecutive targets in the string also represent consecutive diphones in a natural utterance of the database, while this is not guaranteed in the case of searching for targets with a wider context match. In that case, consecutive diphones in synthesis could each find the wider phonemic context in different candidate units from the database, resulting in a join.

A lot of research effort has been devoted to improve synthesis within the existing framework of unit selection. However, this paper shows that a change in the way of target specification and unit search in itself can lead to better quality. This suggests that a simple strategy targeting at longer units can perform as well as standard unit selection with its dependence on different contexts and features, if not even better.

We would investigate other features for specifying phonemic contexts, e.g. by matching the place of articulation, voicing, etc. instead of the actual phoneme identity. We would also scale up our synthesiser in terms of the database size, the number of costs, etc., and investigate their effects on quality.

8. Acknowledgements

The research in this paper was supported by the IWT project SPACE (SBO/040102): SPeech Algorithms for Clinical and Educational applications (home page: http://www.esat.kuleuven.be/psi/spraak/projects/SPACE).
The authors would like to thank the colleagues at ETRO who participated in the listening experiment.

9. References

[1] Hunt, A. and Black A., “Unit selection in a concatenative speech synthesis system using a large speech database”, *ICASSP-96,* Atlanta, GA, vol. 1, pp. 373-376, 1996.

[2] Taylor, P. and Black, A. W., "Speech synthesis by phonological structure matching", *EUROSPEECH ‘99*, Budapest, Hungary, pp. 623-626, 1999

[3] Schweitzer, A., Braunschweiler, N., Klankert, T., Möbius, B., and Säuberlich, B. “Restricted unlimited domain synthesis”, *EUROSPEECH 2003*, Geneva, Switzerland, pp. 1321-1324, 2003

[4] Yang, J.-H., Zhao, Z.-W., Jiang, Y., Hu, G.-P., and Wu, X.-R., “Multi-tier Non-uniform Unit Selection for Corpus-based Speech Synthesis", *Blizzard Challenge 2006*

[5] Clark, R. A. J., Richmond, K., and King, S. “Festival 2: build your own general purpose unit selection speech synthesizer”, *5th ISCA Workshop on Speech Synthesis*, pp. 173-178, 2004

[6] Kerkhoff, J. and Marsi, E. “NeXTeNS: a New Open Source Text-to-speech System for Dutch”, *13th meeting of Computational Linguistics in the Netherlands*, 2002

[7] Visser, J., Van Laarhoven, A. and Ter Beek, A. *AVI-toetsenpakket. Handleiding*, ’s-Hertogenbosch: Katholiek Pedagogisch Centrum (KPC), 1994

[8] Mattheyses, W., Latacz, L., Kong, Y. O., and Verhelst, W. "A Flemish Voice for the Nextens Text-To-Speech System", *IS-LTC-06*, Lublijana, Slovenia, 2006.

[9] Clark, R. A. J, Richmond, K., and King, S. “Multisyn: Open-domain unit selection for the Festival speech synthesis system”, *Speech Communication*, vol49, no. 4, pp. 317-330, 2007.

Evaluation of various unit types in the unit selection approach for the Czech language using the Festival system

Martin Grůber, Daniel Tihelka, Jindřich Matoušek

Department of Cybernetics, University of West Bohemia, Pilsen, Czech Republic

gruber@kky.zcu.cz, dtihelka@kky.zcu.cz, jmatouse@kky.zcu.cz

Abstract

The present paper focuses on the utilization of concatenative speech synthesis, aiming to determine and compare the influence on the synthesized speech quality when various unit types are used in the unit selection approach. There are several unit types which can be used for this purpose. This work deals with those most widely used, i.e. *halfphones*, *diphones*, *phones*, *triphones* and *syllables*. Speech was synthesized using these unit types and the outcome was listened to a by number of listeners, whose task was to evaluate the quality of synthetic speech. The result of the listening test performed for the Czech language is presented. However, it can be assumed that the results would be probably equal for other languages with similar structure, as we made no language-dependent modification in the Festival system. No research of a similar character has been conducted yet, so this unique evaluation should suggest what unit types are appropriate for general TTS systems.

Index Terms: speech synthesis, unit selection, various unit types

1. Introduction

The unit selection approach is one of the possibilities of the concatenative speech synthesis. Today, the method is extensively used due to its simplicity and the increasing quality of the speech produced.

The main principle of concatenative speech synthesis is the concatenation of segments of natural speech signal, which is stored in a *speech corpus* in the form of utterances. It is assumed that speech is composed of *acoustical (speech) units*. The real speech signal is by means of automatic or hand-made segmentation divided into *segments* which correspond to the speech units. These segments are stored in a *unit inventory* as a list of all units, which can be used for synthesis. The synthesized speech is produced as a concatenation of appropriate units from this inventory. It is evident that the synthetic speech, generated in this way, reproduces the voice of the speaker who recorded the speech corpus.

As was mentioned above, the cornerstone of speech is a speech unit. It is an absolute term for marking the same type of speech sound. The specific realization of the specific unit is called *candidate of the speech unit*. However, there is an issue of what the length of the unit should be. The maximum coverage of coarticulation effects and trouble-free concatenation (neither spectral nor prosody discontinuities) are the requirements to meet in this task. In this respect, we would like to choose long units, e.g. words or sentences. On the other hand, we need to keep the unit inventory as small as possible, i.e. to use only a limited number of different units. This requirement makes us use shorter units. In the course of choosing unit type, a trade-off has to be made.

Although we have our own system for speech synthesis [1], *The Festival Speech Synthesis System*[1] [2] was used in order to compare the speech synthesized by various unit types. It would be more difficult to implement the application of various unit types into our system than into the Festival system, which is used for experiments like this. Afterwards, we are planning to apply the achieved results and findings in our system as well.

The Festival system is an environment which was developed at The Centre for Speech Technology Research at The University of Edinburgh. One of its purposes is to allow the researcher to focus on his own problem in terms of speech synthesis instead of developing a whole complex system. Festival is composed of modules which can be modified independently. We adapted those that were originally used for standard diphone unit selection speech synthesis in such a way that it allows the application of four more unit types.

First of all, in section 2, a brief description of the Festival system is stated. Section 3 is dedicated to the application and implementation of various unit types (diphones, phones, triphones, halfphones and syllables) in the Festival system. There are described modifications which were needed to be performed in order to use these units in Festival and the achieved results are also shown. In section 4, the synthesized speech quality using different unit types is evaluated and compared by means of a listening test.

All of the units in the present paper are named according to the Czech version of SAMPA phonetic alphabet.

2. The Festival system

2.1. Introduction

The Festival system is an environment which is suitable for the development of speech synthesizers. It is being used for synthesis in a number of languages, but the basic version contains only data for English and Spanish. The system is intended for 3 groups of users:

- Users who want to generate high quality speech from general text without any knowledge of speech synthesis and without a need to intervene in the process.
- Users who design dialogue systems or any other systems and need to use the output of the speech synthesis. In this case, some changes need to be performed, e.g. particular voice or phrasing selection.
- Researchers developing new methods and approaches to speech synthesis. Indeed, we are among these users, aiming at improving speech synthesis quality. We modified the Festival system so that we could reveal features

[1] free download at http://www.cstr.ed.ac.uk/downloads/

which affect speech quality and make changes to the process of synthesis in order to be able to test various unit types for the purposes of this paper.

2.2. Unit selection

In the unit selection approach, synthetic speech is produced by concatenating speech units selected from a unit inventory.

Each target speech unit has its own list of candidate units. The naturalness of the synthetic speech is then affected by both unit types chosen and candidates selected to build speech. However, once a unit type is chosen it cannot be varied (except for the use of hybrid units, which is not our case), so the only way of controlling speech quality is the criterion of candidate selection. Usually, it consists of two costs.

The first, called *target cost* reflects how each candidate meets the requirements for communication function (what the synthesized phrase is supposed to express or communicate), which also includes the prosodic and phonetic context. The differences between the desired target unit and the real features of a candidate are crucial. In the Festival system, the following features were chosen to describe the communication function: emphasis, position in a syllable, position in a word, position in a phrase, left and right context. Each of these features has a different weight (weights are determined ad hoc) and an overall cost is calculated. It is clear that the application of various unit types requires various features. Some of those mentioned above cannot be used for all of the unit types. For example, the determination of the feature 'position in syllable' is absurd for syllable units and, therefore, it is useless. Other unit types need other modifications in the unit selection algorithm, so we had to make changes to the Festival system in order to be able to use all of them, as described in section 3.

The second one, *join cost*, means how the candidate unit meets the requirements for perceptual smoothness. The differences between the following features of two successive units affect the join cost in the Festival system: F0 and spectral discontinuity (computed as Euclidean distance of vectors composed of z-score normalized 12 MFCC coefficients and energy). These features are also weighted unlikely. The spectral characteristics are determined in instants of time when the first unit of the concatenation ends and the second one begins in their original utterances. It is not guaranteed that the MFCC coefficients are appropriate for the characterization of a unit or computing the join cost; however, they are still widely used for speech synthesis. There is no proof of which features currently examined are the best ones and could be used instead of these coefficients. Thus, we also used them for all unit types in order to be able to compare results correctly.

The best sequence of units is then found using the Viterbi algorithm through the whole unit inventory. It attempts to minimize a cost function which combines the two costs mentioned above.

2.3. Unit inventory

In order to use the unit inventory, it is necessary to create it in such a form that the Festival system needs. In our approach, automatic segmentation (see [1] and [3]) is made by using HTK tools. Moreover, we are also trying to improve it by new methods so that it is able to determine the boundaries of phones more accurately [3]. The current segmentation process produces a file that is not directly usable in Festival. As its output are segments in the form of triphones, several modifications have to be made, and new files (one file for one utterance in a database) are created. For the testing of various unit types, it is easier to adapt these files for all the desired types rather than to make significant modifications in Festival, but some changes in the unit handling modules in the system are still required.

Each file with an utterance has a specific structure and contains the following items: phrases, words, syllables and segments from the utterance, and the relations between these items are also saved there (e.g. which syllable is contained in which word, etc.). These files are then used as a part of the unit inventory. Exactly in this form they can only be used for triphones; for the other unit types they have to be modified. Especially segments need to be renamed and times of their beginning and end have to be determined according to the unit type.

As mentioned above, the MFCC coefficients are used for join cost calculation, so they have to be included in the unit inventory. For the synthesis, the LPC coefficients and residual signal are used. Therefore, it is essential to store these coefficients as well. This is a standard setting in the Festival system; however, the application of different coefficients for join cost computation as well as different coefficients for storing the waveform could be used.

The unit inventory for every single unit type contains all the items mentioned and it is loaded by the Festival system before the synthesis.

3. Application of various unit types in the Festival system

The effort to improve the quality of synthesized speech leads us to the question which unit type is suitable for speech synthesis and under what conditions. Nowadays, there are debates regarding the best unit type selection. It is difficult to determine what type is best for speech synthesis in a TTS system, each having its own advantages and disadvantages. In this paper we attempt to conduct some experiments and establish the strengths and weaknesses, thus contributing to answering this question. By comparing the results achieved we should draw a conclusion what unit type appears to be the best one. Eventually, we could take advantage of every particular unit type and suggest the use of this type in a special system, e.g. any speech synthesizer in a limited domain, which is also as very current topic. This research is unique in comparing the unit types under the same conditions. For all unit types, there is used the same speech corpus, segmentation, features for cost computation, etc.

In order to use all the below-mentioned unit types, we had to make some additional changes in Festival. One of the major modifications consists in the integration of our system of phonetic transcription for the Czech language. The other considerable modification was adding a syllabification algorithm [4]. Both these changes were needed to be performed in order to be able to process any Czech text incoming into the Festival system.

In the following subsections, the application of diphones, phones, triphones, halfphones and syllables is subsequently presented.

3.1. Diphones

We used diphones in this experiment as it is a commonly used unit type in speech synthesizers. Diphones are also the basic units which are used in the Festival system, requiring minimum amount of effort to implement them.

A diphone is a unit beginning in the middle of one phone and ending in the middle of the subsequent phone. The boundary of the diphone is then in the area with stationary signal, which should improve the quality of concatenation. Each unit contains a transition between phones, thereby also including a coarticulation effect which is very important for the naturalness of the synthetic speech.

As mentioned in section 2.3, modification was needed to be performed in the files with utterances for this unit type. It consisted in the renaming of the segments from triphone form to the phone form, because Festival is ready for working with diphones implicitly on the basis of phone names. Festival is able to generate diphone names within the system.

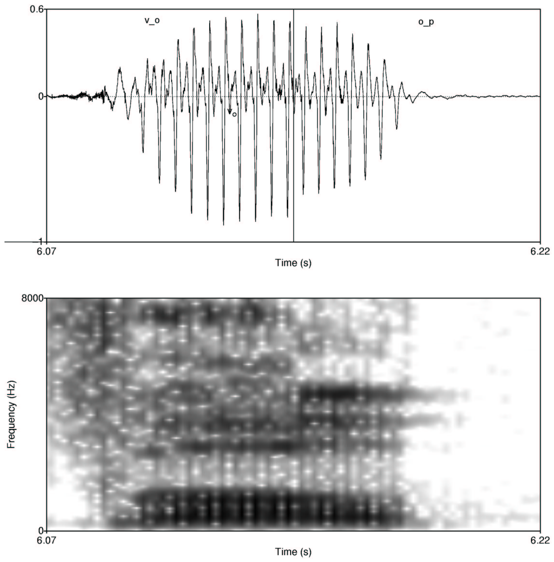


Figure 1: Concatenation of two diphones, originally non consecutive, but continuous in synthesized speech. Waveform and spectrogram.

In fig. 1 the waveform of two concatenated diphones [v_o] and [o_p] is shown. This concatenation was produced as a result of synthesis. It seems to be almost smooth, in spite of the fact that the units were selected from different utterances and they were not originally consecutive. In the spectrogram, the point of concatenation is still visible in the area of higher frequencies (about 4-5 kHZ), but it was not perceived at all.

In fig. 2, there is presented another concatenation of two diphones, [h_a] and [a_#] ([#] denotes pause). Again, they were selected from different original utterances and were non-consecutive. This time, the point of concatenation is extremely visible in the waveform as well as in the spectrogram and it was reported to cause speech degradation in the middle of the phone [a]. For solving this problem, there should be some correction (e.g. some type of normalization) to ensure that there will be at least approximately the same amplitude level. But there is no simple solution because by amplifying the signal of one diphone, we could need to amplify another, and energy accumulation could occur.

One of the advantages of diphones is their relatively small amount. Taking into account the fact that Czech language has 43 different phones, plus 3 types of pauses (loud breath, break and boundary break) and glottal stop, in the sum we have 47 different phone units, it means we have $47^2 \approx 2200$ different diphone units. In addition, some of them don't practically appear in the common text, see table 1 in section 4.

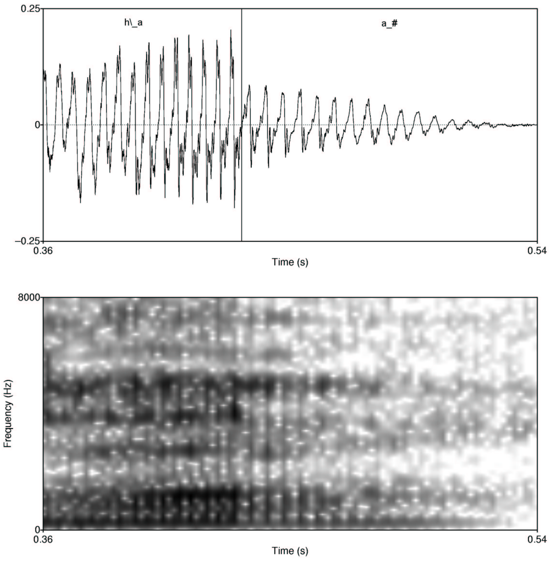


Figure 2: Concatenation of two diphones, originally non consecutive, visibly non continuous in synthesized speech. Waveform and spectrogram.

We made no changes in the target cost and join cost computation algorithm for diphones since the Festival system implicitly treats them in the desired way.

3.2. Phones

A phone is considered to be one of the fundamental phonetic units of speech. The application of this unit type then could seem to be very natural. However, as the boundaries of a phone unit are determined directly by segmentation, it is necessary for the segmentation to be made very accurately. Otherwise, one phone is likely to contain a part of another, which is absolutely undesirable and affects the synthetic speech quality.

Since the triphone segmentation was used, as described in 2.3, triphone labels needed to be renamed to phones which were then stored in Festival's utterance files. This time, changes were made also in the Festival system because otherwise it wouldn't be able to interpret the segment names properly. We had to edit the part of unit handling module that stores the units in Festival's unit inventory.

To illustrate the effect of segmentation inaccuracy, there is shown a concatenation of two units, [v] and [a], that were non-consecutive in the original utterance in fig. 3. The first one ([v]) has a different right context in the original utterance. It is phone [o] and it is easy to see that this phone affects the unit chosen for synthesis. The quality of the synthesized speech is worsen by this effect. Apparently, in this particular case, the cost penalizing incorrect right context was outweighed by other costs.

The seeming advantage is the count of the phone units. For the Czech language we have 47 phones, as was mentioned in the previous section. However, this means that there is a huge amount of candidates for the target unit. Therefore, the enumeration of the best candidate sequence is computationally very exacting and time-consuming. On the other hand, in a very specialized limited domain speech synthesizer (e.g. on the basis of

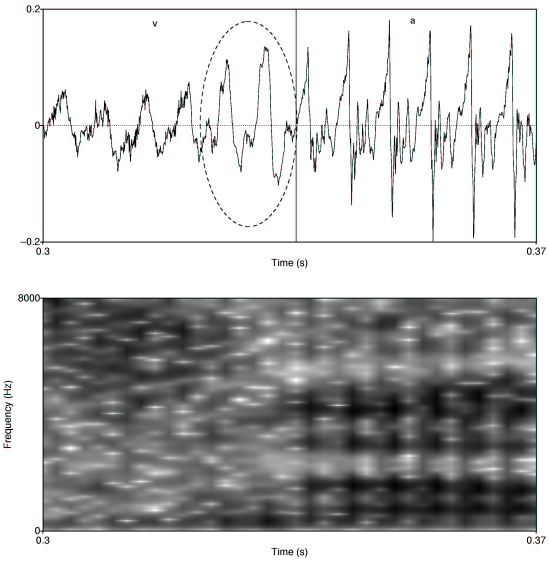


Figure 3: Transition between non-consecutive phones. Right context of the [v] unit was [o] in the original utterance, whereas in the sythesized speech it is phone [a]. This is also visible in the waveform.

sentence unit type), the phones may be advantageous to be used for connecting the sentences in a meaningful way. In that case, diphones would be inappropriate due to their quantity.

We made again no changes in the target cost computation algorithm but a modification was made in the join cost computation. Measurement of the difference between F0 in the joint of two units has sense only in such a case, that we concatenate two voiced or contrariwise two unvoiced units (where the difference should be zero as well as the values of F0). So the algorithm was edited accordingly. When there is a concatenation of a voiced unit with an unvoiced one, this cost is set to zero.

3.3. Triphones

On the basis of good experience with this unit type in the ARTIC speech synthesizer [1], [5], we included it in this research as well. By its principle, it should suppress the disadvantages of phones regarding the transition between units; however, there is still the segmentation problem.

Boundaries of a triphone are the same as for a phone, but the unit includes information about its context. Thus, instead of considering e.g. phone [o], we consider triphone [l-o+r], which means the phone [o] is preceded by phone [l] and followed by phone [r].

For this unit type, no modifications were made in the files with utterances. These files already contain the names of the segments as it is required for the application of this unit type. On the other hand, a modification had to be implemented in the Festival system in order to be able to read the unit inventory from the files correctly.

It is clear that by using this approach we have a large number of different triphone units. In the place of 47 phone units we have $47^3 \approx 100000$ triphone units. As well as in the case of diphones, not all the units appear in the real utterance. However, it is still a great deal of triphones and it is almost impossible to have such a unit inventory that would contain all of them. To avoid this problem, there is an algorithm that groups together the units with similar context. It is made by virtue of the acoustic similarity.

In order to find out which units should be in the same group, we need to take a look at their acoustic signal and a phonetic similarity. Well-suited combination of these two aspects divides the phones for potential left context into 15 groups, and for right context into 14 groups.

For example, phones [p], [t] and [k] are in the same group for the left context. For the right context, these phones are also in the same group, but, in addition, there are also phones [t_s], [t_S] and all 3 types of pauses along with them.

Using this grouping we have only approximately 10.000 units, but it is still possible that during the synthesis there will be a missing unit. In the Festival system, so-called backoff rules (see [2]) can be used. These rules enable the replacement of a unit which is not in the inventory by another unit which is similar to the missing one. It is obvious that this method can be applied mainly to triphones. For other unit types, this kind of replacement could change a sense of synthesized utterances.

The target cost computation did not need to be modified. In the join cost computation algorithm, the same changes as described in previous section for determination of F0 difference were performed.

3.4. Halfphones

The application of halfphones was presented by AT&T Labs in [6] and the results shown there are very promising. Thus, we attempted to compare also this unit type with the others in order to prove or disprove their qualities.

Halfphones are units which start at the beginning of a phone (or in the middle of it) and end in the middle of the same phone (or at the end of it) - they are created by cutting a phone into two halves. Thus, the phone [a] is divided into a sequence of two halfphones, [a1] and [a2].

For the application of this unit type, we had to adapt both the unit inventory in the form of files with utterances and also the Festival system. The main modification was renaming of segments in the unit inventory and editing Festival so that it was able to use this unit type.

The tendency to use the halfphone units could partially replace the application of hybrid diphone-phone (diphone-triphone) unit types. When the halfphones, which are selected during synthesis time, were originally consecutive in an utterance, it means that they could be concatenated into phones, diphones or even longer units. The point of concatenation is sometimes in the middle of a phone and sometimes on the border. As it is noted in [6], the halfphones should be promising units because they could maintain the advantages of phones and diphones. However, they also have disadvantages. One of them is the fact that they are very short, so in a synthesized utterance there is a large number of concatenations. As it is known, at the point of concatenation there could arise many problems which, however, were not reported in [6].

The number of halfphone units should be doubled as compared to the number of phone units, but we didn't cut into halves the units representing pauses. It means that there are 91 different units. This simplification shouldn't affect the final quality of the synthetic speech.

At computation of target cost for these units, there is an anomalous situation. One of the costs which penalizes different left or right context will always be zero (except pause units, because they are treated as phone units). The unit [a1] will always have the unit [a2] as its right context and vice versa, [a2] will always have [a1] as its left context. The algorithm computing this cost could also be modified in such a way that it would consider as a context one more unit following (preceding) the immediate neighbouring unit. The other features affecting the target cost remained the same as for previous unit types. In the join cost computation algorithm, there was made a modification in order to measure F0 difference meaningfully, as described in section 3.2.

3.5. Syllables

Syllables are taken in this experiment as the only representative of longer unit types. It is interesting to confront the previous phone-like unit type with syllables, which include more than one phone (a typical Czech syllable has 2-3 phones).

Syllables are often considered the phonological building blocks of words with boundaries aligned to phones. There again can arise the problem of segmentation inaccuracy.

For this unit type, the files with utterances didn't need to be edited. Segments were ignored and only syllables were used. The modification of the Festival system was in this case more extensive than before. Firstly, we needed to adapt the system, so that it could accept the correct names for syllable units. It was performed the same way as it was performed for previous unit types, by editing the unit handling module.

In addition, some changes in the target cost computation were needed to be carried out, especially the left and right context penalization. It is not necessary to take into account the whole syllable adjacent to the target unit. It is assumed that the whole syllable which forms the context doesn't affect it. Thus, only the last phone of the preceding syllable was treated as the left context and the first phone of the following syllable was treated as the right context. Moreover, these phones were divided into groups in the same way as was done for left and right part of triphone name in section 3.3. The reason is the high number of different syllable units.

The next thing to change in the target cost computation was the feature called position in a syllable. It was removed because it is pointless to use this feature.

As well as for the previous unit type, the join cost computation algorithm was modified. The F0 difference was measured only in such cases when it was meaningful, i.e. when the concatenation occurred in the transition between two voiced or two unvoiced phones.

The problem of the application of syllables is the amount of units. It is not easy even to make a list of all syllables in the Czech language. We use an automatic syllabification [4], which is performed for the phonetically transcribed text, and some syllables are thereby different from the case when it would be implemented for orthographical form of the same text. In addition, the syllabification is not always unambiguous in the Czech language.

In spite of these problems, the list containing about 14.000 syllables which should be included in the unit inventory was generated. There have to be all possible units, and this requirement is almost impossible to achieve. In the application of the syllable units, the backoff rules included with the Festival system are unusable. So in a real TTS system, there has to be another way of synthesizing utterances containing unavailable syllables, e.g. some combination of shorter units. However, like phones, a limited domain synthesis can profit from the advantages that syllables have.

4. Conclusion

In order to compare the results of application of various unit types, we used our speech corpus for synthesizing a listening test. The corpus, recorded in a consistent news-like style by a semi-professional female speaker with some radio-broadcasting experience, contains approximately 12.5 hours of natural speech, stored in 5000 utterances. During synthesis, statistical data about units were collected and are presented here.

Units	Number of different units
Diphones	1528
Phones	47
Triphones	3023
Halfphones	91
Syllables	5684

Table 1: Number of different units in unit inventory for each unit type

In table 1, there is the number of different units in the unit inventory for each unit type. It can be seen that in our fairly large corpus, we covered only 70% of diphones, 30% of triphones and 40% of syllables. phones and halfphones were covered completely, because the number of different units is very low for these unit types. When synthesizing the sentences, we encountered a problem with missing units for triphones and syllables. Therefore, we had to choose such sentences to synthesize which contain only the units we have. For this experiment it is conceivable as we aimed to prove the behaviour of units, not to build a real TTS system where this would have to be solved by another way. For example, in the Festival system the backoff rules could be more adapted to this problem when using triphones or any type of hybrid synthesizer [4] could be used for syllables.

Units	Maximum number of candidates	Minimum number of candidates	Average number of candidates
Diphones	5004	3	1519
Phones	38451	309	17618
Triphones	9994	15	552
Halfphones	38451	309	17693
Syllables	3317	1	788

Table 2: Statistics about units used during the synthesis of utterances for the listening test

In table 2, there is stated maximum, minimum and average number of candidates for each unit type used during the synthesis. You can see, that phones and halfphones have the highest maximum and minimum number of candidates, and these numbers are the same for both of them. The average number differs because in our approach we used the same pause units for phones as well as for halfphones, we didn't cut them into halves. Although the results display the statistics obtained for units used for synthesis of the testing sentences, the results for whole corpus will be very similar.

Taking into account the number of units in a synthesized sentence, which was approximately 150, the number of possible concatenations for phones, diphones and triphones is about n^{150}, where n is the average number of candidates for particular unit types. For halfphones, it is approximately n^{300} because the number of units in the synthesized utterance is doubled. Finally, for syllables it is about n^{60}. It is evident that for phones and halfphones, the algorithm computing the best units sequence needs to perform lots of operations and the whole process of synthesizing is highly computationally exacting. The synthesis of one utterance for the listening test using phones and halfphones takes approximately 24 hours. The fact that it takes the same time for both unit types, in spite of there being more possible concatenations for halfphones, may be explained by any kind of optimalization used by the Festival system, which needs to be more verified. The synthesis using the other unit types takes only a few minutes, but it was still out of real time. However, it does not matter for our experiment because we examined qualities of unit types rather than possibilities of speech synthesis acceleration.

The same corpus, as described earlier, was used to synthesize a listening test. It consists of 5 sentences, each of them was synthesized in 5 various versions. The versions were different in the unit type that was used for synthesis. The sentences were not originally in the corpus and they were selected from newspaper articles.

The listeners were asked to evaluate the synthesized sentences in all versions by marks 1 to 5 (optimally to sort them by quality from the worst one to the best one), where the 5 means the best, this mark always having to be used for the best sentence in terms of naturalness, fluency, intelligibility and prosodic consistency. Sentences which seemed to be equal could be evaluated by equal mark. Afterwards, normalization was performed in order to take advantage of the whole scale. The resulting average marks and standard deviations are shown in table 3.

Units	Average mark	Standard deviation
Diphones	3.61	1.22
Phones	1.88	0.91
Triphones	3.57	1.40
Halfphones	3.81	1.35
Syllables	2.24	1.33

Table 3: The average marks and standard deviations for various unit types

Halfphones with the average mark 3.81 were evaluated as the best unit type. Diphones and triphones have more or less equal marks, as when compared in [5]. However, after performing a statictical one-way analysis of variance (ANOVA), it was proved that there is no significant difference between triphones, diphones and halfphones. During statistical comparison of the results of these three unit types, the p-value for the null hypothesis, that there is no difference among means, reached the value 0.65.

Syllables with the average mark 2.24 were rated a little better than phones, which were identified as the worst ones with the average mark 1.88. This occurs even though the algorithm looking for the best phone sequence theoretically had the best opportunity to select the most appropriate units due to the highest number of candidates. However, in this case as well, the difference between the means of the marks for these two unit types is not statistically significant, which was proved by the ANOVA test. The p-value was determined as 0.094.

On the other hand, between these two groups (diphones, triphones and halfphones on one side, and phones and syllables on the other side) a significant difference was detected. The p-values were equal or near-equal to zero when comparing unit types from one group with those from the other group.

There are further factors which affect unit selection and which can be changed. One of them are weights, used for computation of the target cost and the join cost. In this experiment, Festival implicit setting of these weights was applied. The balancing of the weights should influence the final synthetic speech quality and this setting might be dissimilar for each unit type. However, we attempted to maintain equal conditions for all the tested unit types and in that way achieve a consistent result.

The conclusion may suggest that halfphones, diphones and triphones are comparable regarding the synthetic speech quality. However, taking into account the fact that the synthesis using halfphones was multiple with respect to computational complexity, the application of diphones or triphones seems to be more profitable.

5. Acknowledgements

We thank all those that have supported our research work and development of the synthesis of the Czech language.

This paper has been funded by GAČR 102/06/P205, the Academy of Sciences of the Czech Republic project No. 1ET101470416 and the EU 6th Framework Programme IST-034434.

We also greatly appreciate the help of the computation centre available at our university, which allows us to perform this computationally exacting and time-consuming speech synthesis experiments.

6. References

[1] Matoušek J., Romportl J., Tihelka D., Tychtl Z.: ”Recent Improvements on ARTIC: Czech Text-to-Speech System”, *In Proceedings of INTERSPEECH 2004 - ICSLP, 8th International Conference on Spoken Language Processing. Jeju, Korea, vol. III, pp. 1933-1936. ISSN 1225-441x*

[2] Clark R. A. J., Richmond K., King S.: ”Festival 2 - Build Your Own General Purpose Unit Selection Speech Synthesiser”, *CSTR, The University of Edinburgh*

[3] Matoušek J., Tihelka D., Psutka J.: ”Automatic Segmentation for Czech Concatenative Speech Synthesis Using Statistical Approach with Boundary-Specific Correction”, *In Proceedings of the 8th European Conference on Speech Communication and Technology EUROSPEECH 2003. Geneva, Switzerland, pp. 301-304, ISSN 1018-4074*

[4] Matoušek J., Hanzlíček Z., Tihelka D.: ”Hybrid Syllable/Triphone Speech Synthesis”, *In Proceedings of Interspeech 2005 - Eurospeech, Lisbon, Portugal, s. 2529-2532, ISSN 1018-4074, 2005*

[5] Tihelka D., Matoušek J.: ”Diphones vs. Triphones in Czech Unit Selection TTS”, *TSD 2006. Lecture Notes in Artificial Intelligence 4188, Springer-Verlag, Berlin, Hiedelberg, 2006, pp.531-538., 2006*

[6] Conkie A.: ”Robust Unit Selection System for Speech Synthesis”, *In Proceedings of the Eurospeech ’99 Conference, Budapest, Hungary, 1999.*

Assessing the Adequate Treatment of Fast Speech in Unit Selection Speech Synthesis Systems for the Visually Impaired

Donata Moers, Petra Wagner, Stefan Breuer

Institut für Kommunikationswissenschaften, Abteilung Sprachliche Kommunikation
Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany
{dmo,pwa,sbr}@ifk.uni-bonn.de

Abstract

This paper describes work in progress concerning the adequate modeling of fast speech in unit selection speech synthesis systems – mostly having in mind blind and visually impaired users. Initially, a survey of the main phonetic characteristics of fast speech will be given. From this, certain conclusions concerning an adequate modeling of fast speech in unit selection synthesis will be drawn. Subsequently, a questionnaire assessing synthetic speech related preferences of visually impaired users will be presented. The last section deals with future experiments aiming at a definition of criteria for the development of synthesis corpora modeling fast speech within the unit selection paradigm.

1. Introduction

The option of making a synthesizer “talk fast” is elementary for users who are crucially dependent on their synthesis system in many everyday tasks such as browsing the web, reading emails, reading newspapers etc. and who hardly have any alternative to synthetic speech, i.e. visually impaired or blind users. While reading a web page, the – not visually impaired – user will usually concentrate on certain text passages, e.g. headlines and skip everything that appears to him/her as less interesting. This selective attention leads to a fast reading of least important parts or while a decision is being made whether the text passage currently read is interesting at all. The visually impaired user may want to have a similar option – the possibility to “skim through”. An optionally fast, or even very fast synthesis system is therefore often preferred by this user group.

The phonetic characteristics of fast speech are found to be very different from those of speech produced at “normal” speed. In order to model fast speech during synthesis, the engineer has several options. It is possible to either accelerate the “normal” speech linearly with the help of duration manipulation, to mimic certain prosodic features typical for fast speech such as pauses, intonation and strength of prosodic boundaries or to create an independent inventory inherently showing all segmental and suprasegmental characteristics of fast speech. Previous studies indicate that the different approaches lead to different results in perception experiments. E.g. artificially produced fast spoken words whose temporal pattern was equivalent to natural fast speech were judged to be less intelligible than artificially produced fast spoken words which were simply linearly compressed. The less the stimulus deviated from the canonical form of the word in normal speech the better the word was understood by the listeners [1]. This indicates that a clear pronunciation is still preferred over a synthesis that includes typical phonetic characteristics of natural fast speech such as reductions, elisions and strong coarticulation.

Furthermore, in a comparison of two synthesis architectures where a linear tempo manipulation is easily performed, i.e. formant synthesis and diphone synthesis, blind listeners preferred the less natural sounding formant over diphone synthesis with regards to intelligibility in very fast speech [2]. This indicates that the fast and smooth acoustic transitions in natural fast speech are important for the intelligibility of synthetic speech. Such transitions are not treated adequately by traditional diphone concatenation synthesis but can be modeled by a formant synthesis. Since discontinuities pose a problem for concatenative synthesis in general and unit selection in particular, Breuer [3] suggested to simply treat certain phone sequences which are prone to heavy coarticulation as atomic in the sense that they are regarded as two or more phones, but one indivisable synthesis unit. This approach might lead to a possible solution to model fast synthetic speech both naturally – by using prerecorded concatenation units – and intelligibly – by including typical smooth transitions in heavily coarticulated contexts.

However, a lot of questions concerning the proper treatment of fast speech in unit selection synthesis remain. Taking into account the aforementioned preconditions, the main focus of the – ongoing – project presented here is the definition of robust directives which should be obeyed when building a unit selection synthesis for the visually impaired which can produce fast or very fast speech in an acceptable quality.

2. Phonetic Characteristics of Natural Fast Speech

As stated in the introduction, the characteristics of fast speech differ from those produced at “normal” tempo. Hence, in this section a short overview of the general phonetic characteristics of naturally fast speech is given.

Fast speech differs from “normal” speech both in quality/quantity of vowels and in quality/quantity of consonants. Suprasegmental features like accents, phrase boundaries and the pause durations are also affected by a change in speaking rate. The course of the fundamental frequency is strongly influenced by tempo acceleration. How these differences come about and whether speakers are able to avoid them – because this might be an important option for a synthesizer as well – will shortly be described in the following paragraphs.

2.1. Vowels

Vowels can roughly be described as to consist of three parts: the onset at the beginning of a vowel, which includes the formant movements (transitions) from the preceding sound, the so called steady state almost covering the greatest part in the middle of the vowel, where the formant frequencies stay stable, and the offset, which includes the transitions to the following sound. These transitions from and to another sound are characteristic for certain combinations of sounds and thus important for their correct identification [4].

When speaking faster, vowels are shortened in duration. This process mostly affects the steady state, which is logical since the transitions are very important for the vowel's perceptual identification and may therefore not be curtailed or even left out.

There is not only pure vowel shortening when speaking faster. Another important effect is vowel reduction. Here, reduction refers to a shift of the formant frequencies towards the neutral vowel in the middle of the vowel space [5]. One can assume that this reduction is the consequence of the limited movement velocity of the articulators and/or increasing coarticulation of segments. It is still a matter of ongoing discussion whether the shift of the formant frequencies is a directed movement towards the neutral vowel or simply a consequence of mutual influence between neighboring segments. However, it is questionable whether these phenomena can or should be regarded separately at all. Nevertheless, both of them affect the produced vowel quality and consequently have an impact on the listeners' perception.

2.2. Consonants

Like vowels, consonants are influenced by the acceleration of speaking rate. Like vowels, they are shortened in duration. However, due to the fact that most consonants do not possess a steady state which can be compressed without losing the segment's main characteristics, consonantal shortening is much less pronounced compared to vowels.

Hence, different types of consonants are affected differently by speech rate acceleration. E.g., plosives *become weaker*, which means, that the closures are not complete resulting in a lack of pressure. This leads to plosive bursts performed with less intensity. In consequence, the acoustic characteristics of plosives are more similar to approximants in fast speech [5]. A similar kind of weakening happens to fricatives too: The centers of gravity in their noise spectra show less intensity. Being a combination of plosive and fricative, affricates turn into pure (reduced) fricatives when speaking faster [6].

Another phenomenon occurring in fast speech is the syllabification of consonants. Due to reduction and finally elision of vocalic segments, consonants may become the syllable nucleus. This is accompanied by a duration prolongation of the respective syllabified consonant [7].

Furthermore, the phonetic distinction between voiced and unvoiced consonants is influenced by an increase in speech rate. Since voice onset time (VOT) is decreased, its function as a perceptual cue to distinguish between voiced and unvoiced plosives is neutralized [8].

The effects accumulated above are partly a result of an increasing gestural overlap between subsequent segments in fast speech. The segments have to be articulated in a smaller temporal frame and are therefore produced with more interference, often referred to as coarticulation. Another factor is – similar to vowels – reduction. Due to the fact that the articulators are limited in their movement velocity, the articulators do not reach the optimal target position for each segment. Therefore, the segments as well as the transitions from one to another are not produced as clearly as in speech uttered at normal speed. On the segmental level, these phenomena lead to elision, reduction and assimilation processes, but it is highly context dependent whether or not the phenomena do occur or not.

2.3. Suprasegmental Duration

Apart from phone-specific effects, it has been shown that larger entities, such as the syllable, also behave differently under variations of speaking rate. E.g., unstressed syllables show a stronger shortening in fast speech than stressed ones [9], [10] which actually increases the difference in duration between stressed and unstressed syllables [11], [12]. An investigation in American English indicated that the proportion of stressed syllables decreased from nearly 75 % in normal speech tempo to less than 50 % in fast speech [13]. Anyway, the duration of stressed syllables or even stressed vowels in a stress group stayed stable, despite the increasing number of unstressed syllables.

Nooteboom [14] stated that the vocalic part of a syllable is more variable in fast speech than the consonantal part. But it was also shown, that the syllable internal proportion into 1/3 consonantal and 2/3 vocalic part stays almost stable across different speech tempos [15]. The average number of phones per syllable decreases as speaking rate increases. In addition, the elasticity hypothesis of Campbell and Isard [16] states that the relative duration of the syllable constituents is adjusted to the temporal frame of the syllable by scaling the intrinsic duration according to the temporal demands. Different factors have an influence on this scaling, among them the number of phones in the syllable, the position of the syllable in the phrase, the stress assigned to the syllable and the content of its parent word [ibid.].

2.4. Prosodic Organization

2.4.1. Pauses and Phrase Boundaries

When speaking faster, one of the first and easiest things to do in order to minimize the time for speech production is to decrease or even delete the pauses between utterances or phrases. Thus, there are fewer and shorter pauses in fast speech. The number of phrases decreases as well as prosodic boundaries are omitted or at least reduced [17], [18]. Monaghan [18] also showed also that in fast speech accents are left out and only the most important information remains accented.

2.4.2. Fundamental frequency

In fast speech, fundamental frequency excursions are less pronounced, the intonation contour becomes flatter and the pitch range is reduced. Due to its monotony, this speaking style can give the listener the impression of tediousness [17].

2.5. Semantic and Pragmatic Influences on Rate

As already mentioned, stressed syllables are shortened less than unstressed syllables in fast speech. They remain nearly stable concerning their degree of accentuation if the information they carry is important for comprehension.

Therefore accentuated syllables in content words, that tend to have a higher information content compared to function words, remain stable with an accelerating speech rate [20]. Consequently, content words are less reduced than function words as well.

Similar to tempo changes in a musical piece, speakers vary their tempo within an utterance relative to the linguistic context [21]. Quené [22] found that the Just Noticeable Difference (JND)[1] for human speech adds up to 2.5 % to 5 % difference in speech rate relative to the fundamental rate. Professional speakers produced a variation up to 4 % depending on the degree of novelty of the information in the relevant utterance. Tempo changes which are above the JND threshold are obviously relevant for communication. A speaker may express the relevance of an utterance in a greater context simply by changing the tempo and listeners can interpret a change of speaking tempo as a sign for the importance of what is said.

2.6. Speaking Strategies

Despite the continuous speech flow accompanied by coarticulation, a sufficient contrast between neighboring segments is both necessary and achievable in successful human communication. According to Lindblom's theory of hyper- and hypoarticulation (H&H theory) [23] a contrast is sufficient if it allows the listener to discriminate the signal to the extent necessary to identify the intended item in his mental lexicon. In contrast, the speaker produces speech earmarked and future-oriented. This causes a dilemma because on the one hand the speaker tries to communicate with as little effort as possible. *Hypospeech*, a somewhat more slurry pronunciation style, is the result of this economic constraint. On the other hand the speaker wants to reach a communicative goal, he therefore needs to maintain the phonetic contrast necessary for comprehension. Thus, in situations where comprehension might be more difficult (e.g. in a loud environment) or absolutely essential (e.g. when giving driving instructions) speakers tend to use *hyperspeech*, a very exact pronunciation style. Lindblom describes this phenomenon as follows: „speakers are expected to vary their output along a continuum of hyper- and hypospeech". To be understood by a listener the speaker's (speech)-signals need to feature a sufficient contrast for the listeners' lexical access. For fast speech, we would normally expect speakers to use hypospeech while speaking fast – due to economy. However, speakers may be well able to speak both fast and clear (hyperspeech) if the situation requires this – within certain articulatory constraints.

2.7. Perception

As explained above, the main problem during the perception of natural fast speech is the omission of several acoustic characteristics which are necessary for the correct identification of what has been said. In contrast, it has been shown that if natural speech was compressed up to 65 % of its original duration it was still "perfectly intelligible" [1]. Obviously, the natural acoustic transitions keep the speech intelligible even at fast tempo but the content needs to be semantically or pragmatically predictable to be understood. Even if the temporal compression is further intensified and the compressed utterances have only 35 % of their original duration, they remain comprehensible in the majority of cases (53 %) [24].

2.8. Conclusions and Implications for Fast Synthetic Speech

Speakers follow certain strategies when speaking fast, they reduce vowels and consonants, flatten the fundamental frequency contour and try to minimize duration of pauses and of segments that can be contracted best, i.e. vowels. This process may lead to a loss of distinctiveness and consequently comprehension. However, speakers obey certain rules in order to keep the communication chain working: Semantically important elements of speech are compressed/reduced less than unimportant ones. Nevertheless, with a lot of effort, speakers are well able to speak both clear and fast.

It is possible that a modeling of these speaker strategies may increase naturalness of synthetic speech. Furthermore, it is possible that a stronger contrast between clearly spoken, semantically important and slurrily spoken, less important elements may even increase comprehension of fast synthetic speech, since it draws the attention to the main content of an utterance.

Furthermore, we know that the acoustic transitions of subsequent segments play a vital role in the intelligibility of (fast) speech. The discontinuities added to the speech chain during concatenation must therefore be minimized. This can be achieved straightforwardly by combining phones which are prone to heavy coarticulation into indivisable synthesis units.

We therefore aim to integrate the insights of H&H theory and flexible approaches to inventory creation for unit selection synthesis in order to achieve synthetic speech that is both maximally natural and maximally fast.

3. Preliminary Evaluation

The goal of our present study is to determine an optimal strategy for modeling fast synthetic speech for the visually impaired user. A fundamental problem is the circumstance that preferences – especially of the blind or otherwise visually impaired people – are not investigated as much as it would be necessary for designing an optimal inventory for a fast unit selection speech synthesis.

When starting work for the project some questions came up: What do the blind or visually impaired people aim for concerning speech synthesis? Do they really prefer a monotonous fast synthesis being prosodically relatively close to natural fast speech as suggested in [19]? Or do they not mind a lack in naturalness as long as acoustic transitions important for segment identification are adequately modeled as in formant synthesis [2]? Is it important that the information bearing units are less compressed/reduced than the words carrying less semantic load? What kind of speech quality do they prefer?

The literature concerning these problems proved to be very poor and so it was decided to start a survey among the prospective users. A questionnaire was designed which includes questions about the users'

- fields of synthesis applications
- used or preferred speech synthesis devices
- global preferences concerning speech tempo
- preferred speech rate when listening to synthetic speech

[1] Just noticeable difference is the smallest difference in a specified modality of sensory input that is detectable by a human being. [27]

A second part of the questionnaire deals with several detailed questions related to

- the preferred or desired intelligibility
- the preferred intonation and prosody of fast speech
- the users' desire for an even faster output than what is currently possible
- preferences concerning the tradeoff between naturalness, liveliness and the possibility to have a synthesizer talk very fast.

3.1. (Expected) Results

Due to the fact that at the time of writing this paper the questionnaire has just been released to the public, there are no results available. Nevertheless the following section contains some information concerning the expected outcome. During the workshop, detailed results of the survey will be presented.

4. Further experiments

Based on our previous investigations (cf. 2.) and the outcome of the questionnaire (cf. 3.), we are currently setting up a series of perception experiments aimed to determine an optimal strategy for building a unit inventory that enables us to model fast synthetic speech. The synthetic quality should be especially suited for applications used by the visually impaired. Below we describe the different steps currently undertaken to gather stimuli containing the different articulatory and acoustic features under examination. Then, the anticipated experimental setup is explained. Of course, these are still subject to amendments based on the prospective survey's results.

4.1. Recordings of Synthesis Units

According to the H&H theory, speakers are able to speak both fast and clear if they increase effort. In order to build a useful synthesis inventory that models fast speech, a speaker needed to be found who was able to realize this speaking style best. To determine a competent inventory speaker, preliminary recordings of 9 volunteers were carried out. These recordings were rated by 12 phonetically trained people. They assessed the individual speakers fastest possible articulation rate, their perceptual clarity during fast speech and their individual voice characteristics. Based on these parameters, the presumably most suitable speaker for a fast inventory of a unit selection speech synthesis system was determined.

During inventory creating, the selected speaker read a subset (400 sentences) of the language material contained in the BITS-Corpus [25]. The BITS-Corpus was simply chosen due to its availability and its phonologically balanced design fulfilling the general criteria of unit selection speech synthesis systems.

The sentences are recorded in 2 conditions:

- "normal" speech rate (4 to 5 syllables per second)
- maximum "clear" speech rate (6 to 8 syllables per second)

All recordings were conducted in a sound treated recording studio of our institute. Due to the fact that not all recordings can be done in only one session a strict monitoring of speaking rate, phrasing and intensity is necessary. Prior to each session and within each session, several reference sentences are presented to the speaker in order to (re)adjust her performance and speaking style. The reference sentences are recordings of the first recording session. Special attention is paid to an adjustment of speaking rate, phrasing and accentuation style and intensity. To reach the fastest rate of speech possible it has proven useful to guide the speaker to the designated tempo gradually [26].

All recordings are labeled automatically and corrected manually. Thus, we create two unit selection inventories: one in normal speech rate and one in fast speech rate. In order to assess the general quality of the normal rate inventory and make sure it fulfills the baseline criteria of an acceptable unit selection corpus, the normal rate inventory will be compared with the performance of the existing BITS-inventory. This assessment will be performed by generating and comparing identical sentences from the two different inventories.

4.2. Stimuli and Experimental Setup

As stimuli, different sentences will be generated from the two inventories recorded previously. The stimulus sentences have also been recorded but have not been included in the inventory. Thus, we have templates for further manipulations and comparisons. The first sentence will be generated from normal rate units, the second from fast rate units. A third sentence will be mixed: content words generated from the normal rate units and function words generated from the fast rate units. The motivation for these three groups is that it is still unclear whether listeners prefer fast synthetic speech generated from fast units (most natural?), compressed normal units (most intelligible?) or a mixture of both, trying to mimic the speaking strategies explained by the H&H-theory.

The sentences which are partly or completely generated from the normal rate units presumably will have to be largely manipulated concerning their duration and f0 based on the prerecorded template. It is expected that the sentences which have been generated from the fast rate units will require a comparatively marginal manipulation. This manipulation may create another variable influencing the results of the perception experiments.

There are three groups of stimulus sentences which will be evaluated pairwise in preference tests:

Stimulus Group 1:

- Generated from normal rate units
- Presumably little coarticulation
- Presumably massive prosodic manipulation

Stimulus Group 2:

- Generated from fast rate units
- Presumably massive, but typical coarticulation
- Presumably little prosody manipulation

Stimulus Group 3:

- Generated from normal and fast rate units
- Presumably little coarticulation in content words and massive coarticulation in function words
- Presumably some prosody manipulation

Additionally, stimuli representing a normal speech rate will be generated from the two inventories. These sentences represent a crosscheck. Here, we expect that the sentences generated from the normal rate units are judged much better than that generated from the fast rate units. On the one hand, the fast rate units will have to be massively manipulated, on the other hand they will cause intelligibility problems for the listeners due to their strong pertinent coarticulation and reduction.

The tests shall be conducted with different listener groups. The first group shall consist of people who are not or only slightly visually impaired (e.g. their impairment can be corrected by wearing glasses or contact lenses). In this group, we expect that the preferred sentences will be the ones generated from the normal rate inventory and that the overall preferred tempo of speech is moderate. A second listener group consists of blind or heavily visually impaired people who are reliant on using a speech synthesis system in daily life. Here we expect that these people prefer a fast speech rate, maybe even not intelligible for the visually unimpaired. Furthermore, it is assumed that the fast versions of the sentences where the content words are synthesized from the normal rate units are preferred because the important information is more intelligible and easy to understand.

5. Conclusions

Our paper comprises phonetic knowledge concerning fast speech, discusses implications for its most adequate modeling in concatenation based synthesis applications aimed at visually impaired users and presents a research strategy to investigate this problem further. If the approach chosen in this investigation proves not to be appropriate to synthesize fast speech in an adequate and acceptable quality other ways of producing fast speech in concatenation based synthesis systems have to be considered.

Since our paper described work in progress, only very preliminary results are presented, but first results with regards to the – formerly poorly investigated – tempo related synthesis preferences of visually impaired users will be reported during the workshop.

6. Acknowledgements

The authors would like to thank our speakers, the IfK staff for technical help and David Frauenkron for support during the recordings. Special thanks go to the people who encouraged the work which was presented here.

7. References

[1] Janse, E. (2003): *Word perception in natural-fast and artificially time-compressed speech.* Proceedings 15th ICPhS. Barcelona. pp. 3001 - 3004.

[2] Trouvain, J. (2006): *Subjektive Verständlichkeit von Computerstimmen bei verschiedenen Geschwindigkeiten. Eine Pilotstudie mit zwei Benutzergruppen.* Saarbrücken 2006.

[3] Breuer, S.; Abresch, J. (2004): *Phoxsy: Multi-phone Segments for Unit Selection Speech Synthesis.* In: Proceedings ICSLP. Jeju.

[4] Martínez, F.; Tapias, D.; Alvarez, J.; León, P. (1997): *Characteristics of slow, average and fast speech and their effects in large vocabulary continuous speech recognition.* Proceedings Eurospeech. Rhodes, Greece.

[5] Kohler, K.J. (1990): *Segmental reduction in connected speech in German: Phonological facts and phonetic explanations.* In: Hardcastle, W.J.; Marchal, A. (eds.): Speech Production and Speech Modelling. Dordrecht : Kluwer. pp. 69 - 92.

[6] van Son, R. J. J. H.; Pols, L. C. W. (1996): *An acoustic profile of consonant reduction.* Proceedings ICSLP. Philadelphia. pp. 1529 – 1532.

[7] Roach, P.; Sergeant, P.; Miller, D. (1992): *Syllabic consonants at different speaking rates: A problem for automatic speech recognition.* Speech Communication. Vol. 11, pp. 475 - 479.

[8] Kessinger, R.H.; Blumstein, S.E. (1998): *Effects of speaking rate on voice-onset time and vowel production: Some implications for perception studies.* Journal of Phonetics. Vol. 26, pp. 117-128.

[9] Peterson, G.E.; Lehiste, I. (1960): *Duration of syllable nuclei in English.* JASA. Vol. 32, S. 693 - 703.

[10] Gopal, H.S. (1990): *Effects of speaking rate on the behaviour of tense and lax vowel durations.* Journal of Phonetics. Vol. 18, pp. 497 - 518.

[11] Delattre, P.C. (1966): A *comparison of syllable length conditioning among languages.* Int. Review of Applied Linguistics. Vol. 4, pp. 183 - 198.

[12] Hoequist, C.E. (1983): *Syllable duration in stress-, syllable- and mora-timed languages*. Phonetica. Vol. 40, S. 203 - 237.

[13] Crystal, T.H.; House, A.S. (1990): *Articulation rate and the duration of syllables and stress groups in connected speech.* JASA. Vol. 88, pp. 101 - 112.

[14] Nooteboom, S. (1972): *Production and perception of vowel duration: A study of durational properties of vowels in Dutch.* PhD thesis. Rijksuniversiteit Utrecht.

[15] Kuwabara, H. (1997): *Acoustic and Perceptual Properties of Phonemes in Continuous Speech as a Function of Speaking Rate.* In: Proceedings Eurospeech. Rhodes, Greece.

[16] Campbell, W.N.; Isard, S.D. (1991): *Segment durations in a syllable frame.* Journal of Phonetics. Vol. 19, pp. 37 - 47.

[17] Fougeron, C.; Jun, S. (1998): *Rate effects on French intonation: prosodic organization and phonetic realization.* Journal of Phonetics. Vol. 26, pp. 45 - 69.

[18] Monaghan, A. (2001): *An Auditory Analysis of the Prosody of Fast and Slow Speech Styles in English, Dutch and German.* In: Keller, E.; Bailly, G.; Monaghan, A. et al. (eds.): Improvements in Speech Synthesis. Chichester. pp. 204 - 217.

[19] Fellbaum, K. (1996): *Einsatz der Sprachsynthese im Behindertenbereich.* In: Fortschritte der Akustik. DAGA'96, Oldenburg : DEGA. pp. 78-81.

[20] Schindler, F. (1975): *Faktoren phonetischer Performanz. Instrumentalphonetische Versuche zur akustischen Bestimmung des Ausprägungsgrades von Eigenschaften des lautsprachlichen Signals.* Zeitschrift für Dialektologie und Linguistik. Beihefte. Neue Folge Nr. 14 der Zeitschrift für Mundartforschung. Franz Steiner, Wiesbaden.

[21] Nooteboom, S.; Eefting, W. (1994): *Evidence for the adaptive nature of speech on the phrase level and below.* Phonetica. Vol. 51, pp. 92 – 98.

[22] Quené, H. (2006): *On the just noticeable difference for tempo in speech.* Utrecht 2006.

[23] Lindblom, B. (1990): *Explaining phonetic variation: A sketch of the H&H-Theory.* In: Hardcastle, W.J.; Marchal,

A.: Speech Production and Speech Modelling. Dordrecht: Kluwer. pp. 403 - 439.

[24] Janse, E.; Nooteboom, S.; Quené, H. (2003): *Word-level intelligibility of time-compressed speech: prosodic and segmental factors*. Speech Communication, Vol. 41. pp. 287–301.

[25] Schiel, F.; Draxler, C.; Ellbogen, T.; Jänsch, K.; Schmidt, S. (2006): Die BITS Sprachsynthesekorpora - Diphon- und Unit Selection-Synthesekorpora für das Deutsche.

[26] Greisbach, R. (1992): Reading aloud at maximal speed. Speech Communication. Vol. 11, pp. 469 - 473.

[27] Eefting, W.; Rietveld, A. (1989): Just noticeable differences of articulation rate at sentence level. Speech Communication, Vol. 8. pp. 355–361.

Making Speech Synthesis More Accessible to Older People

Maria Wolters[1], Pauline Campbell[2], Christine DePlacido[2], Amy Liddell[2], David Owens[2]

[1]Centre for Speech Technology Research, University of Edinburgh, Edinburgh, UK
[2]Audiology Division, Queen Margaret University, Edinburgh, UK

mwolters@inf.ed.ac.uk, (pcampbell|cdeplacido|06006484|06005471@qmu.ac.uk)

Abstract

In this paper, we report on an experiment that tested users' ability to understand the content of spoken auditory reminders. Users heard meeting reminders and medication reminders spoken in both a natural and a synthetic voice. Our results show that older users can understand synthetic speech as well as younger users provided that the prompt texts are well-designed, using familiar words and contextual cues. As soon as unfamiliar and complex words are introduced, users' hearing affects how well they can understand the synthetic voice, even if their hearing would pass common screening tests for speech synthesis experiments. Although hearing thresholds correlate best with users' performance, central auditory processing may also influence performance, especially when complex errors are made.

1. Introduction

Older people are a key user group for speech synthesisers. Not only is the percentage of older people in the population increasing, but there are also many groups of older people who will clearly benefit from voice interfaces. Take for example people whose arthritis restricts the motion of their arms and hands: This user group will find it very difficult to navigate traditional graphical user interfaces. Moreover, as the baby boomer generation enters old age, older people are becoming more familiar with and amenable to using computer technology. But there is a fly in the ointment: Older people are also far more likely to have hearing problems than younger users. However, we should be able to optimise our synthetic voices to help compensate for these problems. To achieve this, we need to understand what makes synthetic speech more difficult to understand for older people. In this paper, we report a detailed error analysis of an intelligibility experiment that potentially hints at the direction to take. After a short review of the literature (Section 2), we describe the assessment battery each participant underwent (Sections 3.2 and 3.3) and the experiment itself (Section 3.4). In Section 4, we relate error patterns to selected aspects of participants' hearing, participants' cognitive ability, and problems with the synthetic stimuli. Finally, in Section 5, we suggest how synthesis systems might address the issues found.

2. Background

Older listeners have problems understanding synthetic speech, in particular if they have hearing problems [1], and if there are no contextual cues to compensate for the diminished acoustic cues [2]. Unfortunately, most of the research investigating potential reasons for these problems has not been carried out on unit-selection voices, but on formant synthesisers. The two major problems with formant synthesisers are the dearth of acoustic information in the signal [3] and incorrect prosody [4]. These problems with decoding the signal may place a higher cognitive load on listeners [5]. This increased load may affect older listeners more than younger ones [6]. Since concatenative approaches preserve far more of the acoustic signal than formant synthesisers, dearth of information should not be a problem anymore. Instead, we have problems with spectral mismatches at joins between units, spectral distortion due to signal processing, and temporal distortion due to wrong durations. It is central auditory processing mechanisms that are responsible for tasks such as detecting gaps or compensating for spectral and temporal distortions. Problems with central auditory processing are not picked up by standard pure-tone audiometry. Therefore, we need to expand our range of measures.

The results of Roring et al. [2] may suggest that we need to be particularly careful not to introduce distortions due to signal processing. Their stimuli were generated using an American English diphone voice as supplied with the open source version of Festival [7]. Stimuli were presented at two rates, normal (210 words-per-minute (wpm), duration parameter 1.0), and slow (150 wpm, duration parameter 1.5). The slow rate was chosen based on a 1995 study of DECtalk [8]. Older adults performed significantly worse at the slower rate, which was generated by setting Festival's duration parameter to 1.5 instead of 1.0. Not having heard the original stimuli, we can only speculate that this result was due to increased distortions introduced by PSOLA. As older adults are less able to compensate for those distortions than younger adults, this may partly explain the finding. Langner and Black [1] compared, among other options, speech that was recorded while the speaker was listening to time-varying noise (speech-in-noise) and synthetic speech that was post-filtered to mimic the spectral characteristics speech-in-noise. The original speech-in-noise had a positive effect on performance, the filtered version did not.

Although both Roring et al. [2] and Langner and Black [1] examine the role of hearing problems, neither was able to perform a comprehensive hearing assessment of their participants. Langner and Black relied on self-reports of hearing problems, while Roring et al. used pure-tone audiometry to determine participants' hearing threshold, averaging thresholds for 0.25, 0.5, 1, 2, 3, 4, 6, and 8 kHz.

Roring et al. concluded from their study of the Festival diphone voice, published in 2007, that "[s]ynthetic speech fidelity must be improved significantly before becoming truly useful for the older adult population." [2, p. 25]. One of the aims of this paper and its companion paper [9] is to assess whether unit selection has delivered this significant improvement. In our previous analyses of the data set reported on here [9], we examined correlations between pure-tone thresholds and intelligibility in more detail. We found that the most important threshold to consider is the average threshold for 1, 2, and 3 kHz, corresponding to the range of F2. We also noticed that extended high frequency (UHF) thresholds above 9 kHz correlated well with participants' performance. UHF thresholds are a potential indicator of the general health of the cochlea, since hearing loss begins at the highest frequencies of 20 kHz and propagates down with age. These correlations were not due to a subset of participants with particularly pathological hearing—we can see these trends even in participants who would pass standard screening tests where 0.5, 1, 2, and 4 kHz pure tones are presented at 20dB.

From this brief review of the literature, we see that we know very little about the way in which age-related changes in hearing affect the intelligibility of synthetic speech, in particular unit selection. These age-related changes do not necessarily have to be pathological to affect a person's performance. Furthermore, the role of central auditory processing has barely been explored, even though it is key to compensating for artefacts introduced during the synthesis process.

3. Experiment

3.1. Participants

44 participants took part in our experiment. 12 were aged between 20 and 30, 20 between 50 and 60, and 12 between 60 and 70. The 20-30 group served as controls who showed very few signs of auditory ageing. The 50-60 group were included because they are more likely to show clear evidence of auditory ageing, but less likely to have complex pathologies or require a hearing aid. Finally, the 60-70 group fits with the type of participants that are typically labelled "older". We pooled the participants aged between 50-70 into a generic "older" group because chronological age is notoriously bad at predicting changes in ability [10].

3.2. Cognitive Assessments

We used the Prospective and Retrospective Memory Questionnaire [11] to screen for major memory problems. All scores were well within the normal range. In addition, all participants completed a working memory span (WMS) test [12] that was scored from an answer sheet. The test was presented visually because auditory presentation might affect scores [13]. We used WMS because the experimental task involved remembering the information presented in reminders (cf. Section 3.4 for more detail), and because WMS is highly correlated with other measures of cognitive functioning [10]. Older participants had a significantly lower WMS than younger participants (t-test,t=5.33,df=29.606,p<0.00001). The 20-30's scored on average 38 points out of 42, the 50-70's scored 27. The spread of scores in our test is considerable, with 25% of all participants scoring 24 of 42 possible points or less.

3.3. Audiological Assessments

3.3.1. Pure-Tone Audiometry

Pure-tone (PTA) and ultra high-frequency (UHF) audiometry was measured on a recently calibrated audiometer (Grason-Stadler, Milford, NH; model GSI 61) in a double-walled soundproofed room (Industrial Acoustics Corporation, Staines, Middlesex, UK). Air-conduction thresholds were measured for each ear at 0.25, 0.5, 1, 2, 3, 4, 6, and 8 kHz following the procedure recommended by the British Society of Audiology [14]. UHF thresholds were established at 9, 10, 11.2, 12.5, 14, 16, 18, and 20 kHz. If a participant was unable to detect a tone at the loudest setting `IntMax` for that particular frequency, their threshold for that frequency was recorded as `IntMax` + 5 dB. Testing always began with the better ear in all subjects. Since there are significant differences between the two ears, data from the right and the left ear will be reported separately in this analysis. In this paper, we use the following thresholds:

Trad: Average of 0.5, 1, 2, and 4 kHz, the frequencies conventionally used for screening participants in speech synthesis experiments

F2: Average of 1, 2, and 3 kHz, the frequency range of F2, which has been found to correlate with participants' ability to understand synthetic speech [9]

UHF: Average of 9, 10, 11.2, 12.5, 14, 16, 18, and 20 kHz

3.3.2. Gap Detection

The aim of the gap detection test is to establish the smallest gap between two carrier stimuli that participants can detect. Instead of psychoacoustic testing procedures, we used the Random Gap Detection Test [15], which samples gap detection ability at a fixed set of seven intervals, namely 0, 2, 5, 10, 15, 20, 25, 30, and 40 ms. The sequence in which these intervals are presented is randomised. The stimuli consisted of a 1000 Hz calibration tone and two subtests, the first covering the four frequencies 0.5, 1, 2, and 4 kHz, the second covering clicks. In this paper, we only report results for clicks, because we did not find any correlations between participants' performance on the synthetic speech test and their ability to detect a gap between two tones [16]. This finding is mirrored by studies which found that people's ability to detect gaps between tones does not correlate well with their ability to understand speech in noise, while their ability to detect gaps in noise does [17]. All test items were presented binaurally through a GSI 61 audiometer (model GSI 61; Grason-Stadler, Milford, NH) and a high fidelity Sony cassette with calibrated TDH-49 headphones.

3.3.3. Speech Audiometry

The speech audiometry test used a set of 20 standard CVC word lists [18]. Each list was 10 words long. After each word, participants need to repeat what they heard. The score is the number of phonemes that were repeated correctly, with the maximum score 30 (10 words $\times$ 3 phonemes). Word lists were initially presented at a comfortable intensity derived from participants' PTA scores. That intensity was increased until participants scored 30 out of 30 phonemes correct, and then lowered again until participants' score dropped to 3 out of 30 phonemes (10%) or worse. Intensity was changed by 5 dB at a time.

3.4. Synthesis Experiment

For this study, we used stimuli that are closely modelled on a real-life application—task reminders. Task reminders were chosen because they are an integral part of many relevant applications, ranging from electronic diaries to cognitive prosthetics [19]. Since our research focusses on adapting speech technology to the home care domain, we investigated two relevant types of reminders: reminders to meet a specific person at a given time, and reminders to take a specific medication at a given time. 32 reminders were generated, 16 meeting reminders and 16 medication reminders. Time preceded person or medication in half the sentences, person/medication preceded time in the other half. Table 1 shows the sentence templates that were used. Each template was used eight times.

Table 1: *Reminder Templates*

Reminder	Template
Meeting	At TIME, you are meeting PERSON. You are meeting PERSON at TIME.
Medication	At TIME, you need to take your MEDICATION. You need to take your MEDICATION at TIME.

3.4.1. *Stimuli*

There were three categories of target stimuli, times (easiest), person names (medium difficulty), and medication names (most difficult). Since *temporal expressions* are relatively distinct from each other, it is difficult to elicit errors. We addressed this problem by focussing on two sets of phonologically similar hours: "seven", "eleven" and "twelve" and "one", "nine", and "ten". We added further complexity by adding complex expressions such as X to HOUR and X past HOUR, where X was one of "ten", "twenty", and "a quarter". We chose *proper names* that matched the pattern C_1VC_2, where both consonants were oral or nasal stops, because stops are more easily confundable than other consonant types [20, 21]. For each proper name (except for "Dan"), we ensured that there was at least one other proper name that differed from the name by just one consonant. *Medication names* were constructed by recombining morphemes taken from actual medication names. Care was taken to ensure that the medication names did not resemble any existing or commonly used medication to avoid familiarity effects. All names are 3-4 syllables long; seven contain at least one consonant cluster. Table 2 lists all targets used in the experiment.

3.4.2. *Voices*

For the *synthetic speech* condition, all 32 reminders were synthesised using Scottish female voice "Heather" of the unit selection speech synthesis system Cerevoice [22]. Medication names were added to the lexicon before synthesis to eliminate problems due to letter-to-sound rules. The transcriptions were adjusted to render them maximally intelligible. No other aspects of the synthetic speech were adjusted.

For the *natural speech* condition, the reminders were read by the same speaker who provided the source material for the synthetic voice. The natural speech was then postprocessed using the procedures used for creating synthetic speech: high-pass filtering with a cut-off frequency of 70 Hz, then downsampling to 16kHz, and finally encoding and decoding with the tools `speexenc` and `speexdec`. This procedure ensures an exceptionally close matching between human and synthetic speech.

3.4.3. *Experiment Design and Procedure*

Four stimulus lists were created, each comprising 32 reminders. Each reminder was followed by a short question, recorded using the same natural voice as that used for the reminders. Each participant only heard one of the four lists. Each reminder was presented using the synthetic voice in two lists, and using natural speech in the remaining two. In two lists (one synthetic, one natural), participants were asked for the first item of a given reminder, while in the other two conditions, participants were asked for the second item.

The sequence of reminders was randomised once and then kept constant for all four lists. Each participant had to correctly remember 32 targets: 8 times presented in a natural voice, 8 times presented in a synthetic voice, 4 medication names presented in a synthetic voice, 4 medication names presented using a human voice, 4 person names presented using a human voice, and 4 person names using a synthetic voice.

Participants replied verbally with the information which they had been asked to recall. All responses were written down during the experiment and recorded using a minidisc recorder for further transcription and scoring. The total number of responses collected was 1408, with 352 times, 352 person names, and 704 times. For each category, half the responses are to the natural version, half to the synthesised version.

3.4.4. *Scoring*

Participants' pronunciations were scored by a phonetician (MW) based on whether their response was an acceptable pronunciation of the orthographical form of the target. This allows us to adjust for effects of the participants' dialect, such as rhoticity or differences in vowel quality. Deviant pronunciations that could not be accounted for by dialect were classified into three categories:

phoneme errors: Insertion, deletion or replacement of one consonant or vowel in a syllable. Example: Propanodryl → Prop**r**anodryl, Beclotor → Beclo**d**or. Phoneme errors occur in person names and medication names.

syllable errors: More than one phoneme error in the pronunciation of a syllable. Syllable errors only occur in medication names. Example: Propanodryl → Propano**lol**

word errors: One of the target words is replaced by a different word. Medication names were scored as wrong words if all of the word's syllables were affected by syllable errors. Word errors occurred in all three stimulus categories. Example: eleven → **seven**

Responses were scored as **correct** if they contained no errors.

Table 2: *Target Stimuli*

Item type	Items
Person	Ben, Bob, Dan, Don, Dick, Ned, Nick, Rick, Rob, Ron, Ken, Kim, Jim, Tim, Ted, Tom
Medication	Accumycin, Beclotor, Dexozine, Erytozole, Fosinarol, Kisinolol, Levapril, Mevacycline, Pravaclor, Propanodryl, Sulfacillin, Streptostatin, Tetradine, Trovalide
Times	one, four, five, seven, nine, ten, eleven, twelve ten past ten, ten past three, ten past twelve, ten past two, ten to eight, ten to eleven, ten to one, ten to ten twenty past ten, twenty past three, twenty past twelve, twenty past two, twenty to eight, twenty to eleven, twenty to one, twenty to ten quarter past ten, quarter past three, quarter past twelve, quarter past two, quarter to eight, quarter to eleven, quarter to one, quarter to ten

4. Results

Results are presented in three stages. First, we examine whether some stimuli were more difficult to process than others and present results of a detailed inspection of the synthetic speech signals that caused particular problems (Section 4.1). Next, we examine the effect of ageing. Instead of testing chronological age, we focus on measures of cognitive ability (Section 4.2) and hearing loss(Section 4.3, both of which are linked to ageing.

4.1. The Effect of the Stimuli

We determined the effects of three independent variables characterising the nature of the stimuli, category (person, time, or medication), voice (synthetic or human) and position in the reminder (first or second), on participants' ability to remember the stimulus correctly. A three-way ANOVA shows main effects of the category (df=2,F=278.66,p<0.00001), voice (df=1,F=26.66,p<0.0001) and position (df=1,F=5.58,p<0.05). Tukey's HSD post-hoc tests reveal that synthetic stimuli are more difficult to remember than those spoken by the natural voice, items in second place are easier to remember than items in first place, and persons and times are easier to remember than medications (cf. Table 3). This validates our decision to test all three types of responses. The reasons for this result are clear: Times and person names are frequent, familiar, and phonologically simple, whereas medication names are unfamiliar and phonologically complex. We also find a clear interaction between stimulus category and voice (df=2, F=33.06, p<0.0000001). Our post-hoc tests reveal that in fact, participants remember times and person names well *no matter what the voice*—it is the complex, unfamiliar medication names that make the difference: Performance doubles for the natural voice compared to the synthetic voice. Therefore, when messages are restricted to stimuli using familiar words in familiar contexts, older users may be able to cope perfectly well with modern synthetic voices.

Although average scores for person names and times are similar, performance on the two categories is not correlated (ρ=-0.09,df=42,p>0.5). Neither is there a correlation between the number of correct person names and the number of correct medication names (ρ=0.19,p>0.2), nor between the number of correct times and the number of correct medication names (ρ=0.13,p>0.4). If participants' performance for the three response categories is uncorrelated, then performance on each category is potentially determined by different factors.

For six targets, the performance difference between natural and synthetic versions was 30% or worse. These were the medication names "Accumycin", "Beclotor", "Erytozole", "Mevacycline", "Pravaclor", and "Sulfacillin". In two of these, "Accumycin" and "Sulfacillin", there are clear bad joins. The second syllable of "Accumycin" is often misheard as "clu" or "cru". This could be due to a bad join in the /m/ of "mycin", where a nasal with relatively weak intensity meets a nasalised /a/. Likewise, "Sulfacillin" is affected by a bad join in the first vowel /ᴜ/, and "-lin" is rendered as /lɪnɪn/. As a consequence, 33% of participants misheard the suffix, and 50% confused the initial /ᴜ/ with an initial /ɪ/. In the remaining four, "Erytozole", "Mevacycline", "Pravaclor", and "Beclotor", the problem lies elsewhere. With "Mevacycline", "meva-" is often misheard as "neva-". This could be due to a tricky transition between the final /r/ of "your" and the initial /m/ of "meva". With "Pravaclor", the third syllable is affected most, with participants omitting the /l/, which is very short, or changing the nucleus to /a/, which may be due to the almost vocalic final /r/. For "Erytozole", the suffix "zole" is often confused with a similar sounding suffix. This could be due to the relatively rapid transition to the following preposition "at". "Beclotor" was affected worst. This is not due to bad joins, but to very short nuclei whose identity is difficult to identify. Moreover, the final /r/ is very short and segues quickly into the initial vowel of the following "at". As a result, none of the participants identifies the suffix correctly. Most misinterpret "-tor" as "-tin", and only seven correctly identify the /l/ in "-clo".

The picture sketched above for the six medications with the biggest performance difference between the natural and the synthetic version holds for the other medications as well: Bad joins are less of a problem than transitions that are too fast and durations, in particular of second consonants in consonant clusters, that are too short.

Table 3: *% correct by voice and stimulus category*

Category	**Voice**		Total
	Natural	Synthetic	
Medication	65.91%	35.23%	50.57%
Person	96.59%	90.91%	93.75%
Time	94.60%	96.02%	95.31%
Total	87.93%	79.55%	83.75%

4.2. The Effect of Memory

Working memory score is highly correlated with participants' performance on natural stimuli (ρ=0.42, 95% confidence interval [0.14,0.64], p<0.01), but not with performance on synthetic stimuli in general (ρ=0.23, 95% CI [-0.07,0.49], p>0.1). Looking at the effect of working memory span on the kinds of errors made, we find a significant correlation with words substituted (ρ=-0.36, 95% CI [-0.59,-0.07], p<0.05), but not with altered phonemes or syllables.

4.3. The Effect of Hearing

After examining potential confounders such as particularly difficult items and working memory, we turn to the central aspect of our study, the influence of hearing. We are looking for aspects of hearing that are highly correlated with participants' performance: the number of correct responses, the amount of phoneme errors, the number of syllable errors, and the number of word errors. The audiological measures included in our analysis (cf. Sec. 3.3) are:

Pure Tone Audiometry: TRADL, TRADR, F2L, F2R, UHFL, UHFR

Central Auditory Processing: MAXR, MAXL (Speech audiometry); GAP (gap detection in noise)

It would be very convenient if most of the results obtained were due to participants with abnormal hearing that would have been eliminated automatically by the traditional screening test, with the average threshold TRAD for 0.5, 1, 2, and 4 kHz at 20dB or lower for both. For this reason, we present results for two groups of participants:

Full: the complete group of 44 participants

Screened: the subgroup of 35 (79.55%) participants who would have passed the traditional screening test

Of the group SCREENED, 5 (14%) had a gap detection threshold in noise of 20 ms or higher. 8 (23%) had to hear the speech audiometry word lists at 60dB or louder to obtain a perfect score. This is well above the dynamic range of normal speech, which varies between 20 and 50 dB.

Tables 4–7 summarise the audiological measures which correlate with participants' performance on synthetic versus natural speech. Measures for which correlations are significant at a level of p<0.005 are presented in **bold**, correlations with p<0.01 are in normal type, and measures for which correlations are significant at p<0.05 in *italics*.

All correlations are in the expected direction: the higher audiometric thresholds, the higher the gap detection threshold, and the higher the maximum intensity at which participants correctly repeated all words, the worse their performance. The first key result to note is that for both full group and screened group, aspects of hearing clearly influence performance. This is a powerful argument for including at least some simple hearing thresholds as covariates when analysing results of intelligibility tests. Even though our population was significantly older than the usual undergraduate testers, age does not imply healthy ears: We excluded two younger subjects from our initial pool of 15 younger participants because of low-frequency hearing loss.

The key *hearing threshold* is not one of the traditional screening values TradR and TradL, but F2L. This is the one threshold that correlates well with our error measures, no matter what the group. This is good news, because like TradL, it is relatively quick and easy to measure. We also find strong correlations with the ultra-high frequency hearing thresholds UHFR and UHFL, which confirms our earlier findings [9]. The correlations between UHFR and UHFL and participant performance are stronger for the subgroup that would have passed screening than for the full group. This is interesting, since losses at ultra-high frequencies precede losses further down the basilar membrane.

Our measurements of *central auditory function*, MaxR, MaxL, and Gap are mainly correlated with participants' performance on natural speech - they play a far smaller role in predicting errors on synthetic speech. In particular, MaxR correlates well with the number of correct responses, and the number of word errors. This reflects the design of this particular test, which looks at the ability to correctly understand monosyllables. Gap is only relevant in accounting for syllable errors made when repeating synthetic stimuli (cf. Table 6): The less participants are able to detect small gaps in noise, the more likely they are elide, substitute, or insert two or more phonemes in a syllable of a complex multisyllabic stimulus.

Finally, the evidence shows very clearly that hearing problems affect natural and synthetic speech differently, even though the underlying speaker was the same. The key differences are:

- Speech audiometry correlates far better with people's ability to understand natural speech than with their ability to understand synthetic speech.
- Performance for synthetic stimuli on the other hand is predicted mostly by pure tone audiometry thresholds.
- No audiological measures correlate significantly with the number of phoneme errors made on synthetic speech, and no measures correlate significantly with the number of syllable errors made on natural speech.

5. Discussion

Our results indicate that older people can remember and process synthetic stimuli just as well as those produced by natural speech if the text consists of familiar words and phrases. We can exploit this finding by ensuring that prompts are redundant and contain frequent and familiar words. Since quite a few problems with the synthetic stimuli occurred at transitions between the target words and the surrounding sentence matrix, a quick hack to avoid these problems would be to delimit the key content words by very short pauses. These general design guidelines can be implemented almost immediately and benefit all users regardless of age.

Table 4: *Correlation of audiological measures with performance on reminder task*

	Full (n=44)	
	Natural	Synthetic
Audiometry	**F2L**	F2L, UHFR
	TradL	*UHFL, TradL*
Central	**MaxR**, MaxL	MaxL
	Screened (n=35)	
	Natural	Synthetic
Audiometry	(none)	**F2L, UHFR, UHFL**
		F2R, TradL, *TradR*
Central	(none)	(none)

Table 5: *Correlation of audiological measures with phoneme errors*

	Full (n=44)	
	Natural	Synthetic
Audiometry	**F2L, TradL**	(none)
	F2R, TradR, UHFL, UHFR	
Central	*MaxR*	(none)
	Screened (n=35)	
	Natural	Synthetic
Audiometry	*TradR, F2R*	(none)
Central	(none)	(none)

Considerable differences emerge only when the text to be synthesised contains phonologically complex, unfamiliar stimuli. This result needs to be investigated further in a more systematic study where phonological complexity and familiarity are both varied systematically.

Our results also demonstrate that factors which will affect the ability to understand natural speech do not necessarily affect the ability to understand synthetic speech. Hence, we cannot just extrapolate from the literature on human speech recognition, but need to reevaluate all findings carefully.

A more detailed analysis of the results shows that people's ability to understand synthetic speech is greatly influenced by pure-tone audiometric thresholds. Central auditory processing has a small, but decisive influence. For example, when remembering phonologically complex syllables, the ability to detect small gaps in the signal becomes important. This indicates that users' ability to understand synthetic speech may depend mainly on aspects of auditory function that affect the general processing of auditory stimuli, and less on users' ability to understand speech.

The natural response to this result might be to apply preemphasis to relevant frequency ranges. However, the benefits of any signal processing need to be weighed against the distortions it introduces. Furthermore, detailed post-hoc error analyses show that the main source of errors are not bad joins, but segments that are too short and transitions that move too quickly. Hence, it might be more effective to use units for important content words that are longer and contain clearer auditory cues. We hope to investigate this hypothesis in future work.

Table 6: *Correlation of audiological measures with syllable errors*

	Full (n=44)	
	Natural	Synthetic
Audiometry	(none)	F2L, F2R, *UHFL, UHFR*
Central	(none)	**MaxR**, *Gap*
	Screened (n=35)	
	Natural	Synthetic
Audiometry	(none)	*F2L, UHFR, UHFL*
Central	(none)	*Gap*, *MaxL*

Table 7: *Correlation of audiological measures with word errors*

	Full (n=44)	
	Natural	Synthetic
Audiometry	(none)	*F2L, TradL*
Central	*MaxR*	(none)
	Screened (n=35)	
	Natural	Synthetic
Audiometry	(none)	TradR, *TradL, UHFL* *UHFR, F2R, F2L*
Central	MaxR, *MaxL,GapNoise*	(none)

6. Acknowledgements

This research was funded by the EPSRC/BBSRC initiative SPARC and by the SFC grant MATCH (grant no. HR04016). We would like to thank our participants for their patience, Matthew Aylett and Christopher Pidcock for their invaluable help with generating the stimuli, and Ravi Chander Vipperla for his generous help with digitising minidiscs.

7. References

[1] B. Langner and A. W. Black, "Using Speech In Noise to Improve Understandability for Elderly Listeners," in *Proceedings of ASRU, San Juan, Puerto Rico*, 2005.

[2] R. W. Roring, F. G. Hines, and N. Charness, "Age differences in identifying words in synthetic speech," *Hum Factors*, vol. 49, pp. 25–31, 2007.

[3] S. Duffy and D. Pisoni, "Comprehension of synthetic speech produced by rule: A review and theoretical interpretation," *Language and Speech*, vol. 35, pp. 351–389, 1992.

[4] C. R. Paris, M. H. Thomas, R. D. Gilson, and J. P. Kincaid, "Linguistic cues and memory for synthetic and natural speech," *Human Factors*, vol. 42, pp. 421–431, 2000.

[5] P. Luce, T. Feustel, and D. Pisoni, "Capacity demands in short-term memory for synthetic and nautral speech," *Human Factors*, vol. 25, pp. 17–32, 1983.

[6] J. Al-Awar Smither, "The processing of synthetic speech by older and younger adults," in *Proceedings of the Human Factors Society 36th Annual Meeting. Innovations for Interactions, 12-16 Oct. 1992*. Atlanta, GA, USA: Human Factors Soc, 1992, pp. 190–192.

[7] A. Black and P. Taylor, "The festival speech synthesis system," Human Communication Research Centre, Tech. Rep. TR-83, 1997.

[8] B. Sutton, J. King, K. Hux, and D. Beukelman, "Younger and older adults' rate performance when listening to synthetic speech," *Augmentative and Alternative Communication*, vol. 11, no. 3, pp. 147–153, 1995.

[9] M. Wolters, P. Campbell, C. DePlacido, A. Liddell, and D. Owens, "The effect of hearing loss on the intelligibility of synthetic speech," in *Proc. Intl. Conf. Phon. Sci.*, Aug. 2007.

[10] T. A. Salthouse, "Where in an ordered sequence of variables do independent age-related effects occur?" *J.Gerontol.B Psychol.Sci.Soc.Sci.*, vol. 51, pp. 166–178, 1996.

[11] J. R. Crawford, G. Smith, E. A. Maylor, S. della Sala, and R. H. Logie, "The Prospective and Retrospective Memory Questionnaire (PRMQ): Normative data and latent structure in a large non-clinical sample," *Memory*, vol. 11, pp. 261–275, ,Psychological physiopathology 2003.

[12] N. Unsworth and R. Engle, "Simple and complex memory spans and their relation to fluid abilities: Evidence from list-length effects," *Journal of Memory and Language*, vol. 54, pp. 68–80, 2006.

[13] P. Rabbitt, "Mild hearing loss can cause apparent memory failures which increase with age and reduce with IQ," *Acta Otolaryngol.Suppl*, vol. 476, pp. 167–175, 1990.

[14] British Society of Audiology, "Pure tone air and bone conduction threshold audiometry with and without masking and determination of uncomfortable loudness levels," 2004.

[15] R. W. Keith, *The Random Gap Detection Test*. St. Louis: AUDITEC, 2000.

[16] D. Owens, P. Campbell, A. Liddell, C. DePlacido, and M. Wolters, "Random gap detection threshold: A useful measure of auditory ageing?" in *Proc. Europ. Cong. Fed. Audiol. Heidelberg, Germany*, Jun.

[17] K. B. Snell, F. M. Mapes, E. D. Hickman, and D. R. Frisina, "Word recognition in competing babble and the effects of age, temporal processing, and absolute sensitivity," *Journal of the Acoustical Society of America*, vol. 112, pp. 720–727, 2002.

[18] A. Boothroyd, "Developments in speech audiometry," *British Journal of Audiometry*, vol. 2, pp. 3–10, 1968.

[19] M. Pollack, "Intelligent Technology for an Aging Population: The Use of AI to Assist Elders with Cognitive Impairment," *AI Magazine*, vol. 26, pp. 9–24, 2005.

[20] J. R. Dubno and H. Levitt, "Predicting Consonant Confusions from Acoustic Analysis," *Journal of the Acoustical Society of America*, vol. 69, pp. 249–261, 1981.

[21] S. Gordon-Salant, "Recognition of natural and time/intensity altered CVs by young and elderly subjects with normal hearing," *J.Acoust.Soc.Am.*, vol. 80, pp. 1599–1607, 1986.

[22] M. A. Aylett, C. J. Pidcock, and M. E. Fraser, "The cerevoice blizzard entry 2006: A prototype database unit selection engine," in *Proceedings of Blizzard Challenge Workshop, Pittsburgh, PA*, 2006.

The HMM-based Speech Synthesis System (HTS) Version 2.0

Heiga Zen[1], *Takashi Nose*[2], *Junichi Yamagishi*[23], *Shinji Sako*[14],
Takashi Masuko[2], *Alan W. Black*[5], *Keiichi Tokuda*[1]

[1]Nagoya Institute of Technology, [2]Tokyo Institute of Technology, [3]University of Edinburgh,
[4]Tokyo University, [5]Carnegie Mellon University

zen@sp.nitech.ac.jp, Takashi.Nose@ip.titech.ac.jp, jyamagis@inf.ed.ac.uk, sako@mmsp.nitech.ac.jp
awb@cs.cmu.edu, tokuda@nitech.ac.jp

Abstract

A statistical parametric speech synthesis system based on hidden Markov models (HMMs) has grown in popularity over the last few years. This system simultaneously models spectrum, excitation, and duration of speech using context-dependent HMMs and generates speech waveforms from the HMMs themselves. Since December 2002, we have publicly released an open-source software toolkit named HMM-based speech synthesis system (HTS) to provide a research and development platform for the speech synthesis community. In December 2006, HTS version 2.0 was released. This version includes a number of new features which are useful for both speech synthesis researchers and developers. This paper describes HTS version 2.0 in detail, as well as future release plans.

1. Introduction

Currently the most popular speech synthesis technique is unit selection [1–3], where appropriate sub-word units are selected from large speech databases. Over the last decade, this technique has been shown to synthesize high quality speech and is used for many applications. Although it is very hard to surpass the quality of the best examples of unit selection, it does have a limitation that the synthesized speech will strongly resemble the style of the speech recorded in the database. As we require speech which is more varied in voice characteristics, speaking styles, and emotions, we need to record larger and larger databases with these variations to achieve the synthesis we desire without degrading the quality [4]. However, recording such a large database is very difficult and costly [5].

Over the last few years, a statistical parametric speech synthesis system based on hidden Markov models (HMMs) has grown in popularity [6–10]. In this system, context-dependent HMMs are trained from databases of natural speech, and we can generate speech waveforms from the HMMs themselves. This system offers the ability to model different styles without requiring the recording of very large databases.

Figure 1 is an overview of this system. It consists of training and synthesis parts. The training part is similar to that used in speech recognition systems. The main difference is that both spectrum (mel-cepstral coefficients [11], and their dynamic features) and excitation (logarithmic fundamental frequencies ($\log F_0$) and its dynamic features) parameters are extracted from a speech database and modeled by context-dependent HMMs (phonetic, linguistic, and prosodic contexts are taken into account). To model variable dimensional parameter sequence such as $\log F_0$ with unvoiced regions properly, multi-space probability distributions (MSD) [12] are used. Each HMM has state duration probability density functions (PDFs) to capture the temporal structure of speech [13, 14]. As a result, the system models spectrum, excitation, and durations in a unified HMM framework [6]. The synthesis part does the inverse operation of speech recognition. First, an arbitrarily given text to be synthesized is converted to a context-dependent label sequence, and then an utterance HMM is constructed by concatenating the context-dependent HMMs according to the label sequence. Second, state durations of the utterance HMM are determined based on the state duration PDFs. Third, the speech parameter generation algorithm (typically, the Case 1 algorithm in [15] is used, please refer to Section 2.4 for detail) generates the sequence of spectral and excitation parameters that maximize their output probabilities. Finally, a speech waveform is synthesized directly from the generated spectral and excitation parameters using the corresponding speech synthesis filter (mel log spectrum approximation (MLSA) filter [16] for mel-cepstral coefficients).

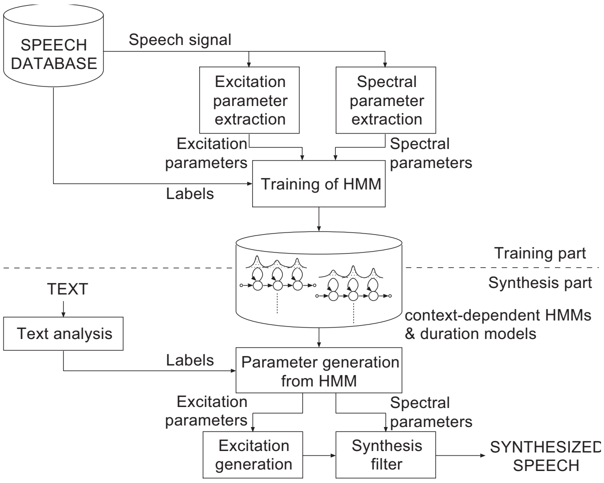


Figure 1: Overview of a typical HMM-based speech synthesis system.

The most attractive part of this system is that its voice characteristics, speaking styles, or emotions can easily be modified by transforming HMM parameters using various techniques such as adaptation [17, 18], interpolation [19, 20], eigenvoice [21], or multiple regression [22].

Since December 2002, we have publicly released an open-source software toolkit named HMM-based speech synthesis system (HTS) [23] to provide a research and development platform for speech synthesis community. Currently various organizations use it to conduct their own research projects, and we believe that it has contributed significantly to the success of HMM-based synthesis today. In December 2006, HTS version 2.0 was released. This version includes a number of new features which are useful for both speech synthesis researchers and developers. This paper describes relevant details of this system, and future release plans.

2. HTS: A toolkit for HMM-based speech synthesis system

2.1. Outline

The HMM-based speech synthesis system (HTS) has been being developed by the HTS working group as an extension of the HMM toolkit (HTK) [24]. The history of the main modifications we have made are listed below:

- Version 1.0 (December 2002)
  - Based on HTK-3.2.
  - Context clustering based on the minimum description length (MDL) criterion [25].
  - Stream-dependent context clustering [6].
  - Multi-space probability distributions (MSD) as state output PDFs [12].
  - State duration modeling and clustering [13].
  - Speech parameter generation algorithm (Case 1 in [15] only).
  - Demo using the CMU Communicator database.
- Version 1.1 (May 2003)
  - Based on HTK-3.2.
  - Small run-time synthesis engine.
  - Demo using the CSTR TIMIT database.
  - HTS voices for the Festival speech synthesis system [26].
- Version 1.1.1 (December 2003)
  - Based on HTK-3.2.1.
  - Demo using the CMU ARCTIC database [27].
  - Demo using the Nitech Japanese database.
  - Variance flooring for MSD-HMMs.
  - Post-filtering [28].
  - HTS voice for the Galatea toolkit [29].

The source code of HTS is released as a patch for HTK. Although the patch is released under a free software license similar to the MIT license, once the patch is applied users must obey the license of HTK.[1] Since version 1.1, a small run-time synthesis engine named `hts_engine` has been included. It works without the HTK libraries, hence it is free from the HTK license. Users can develop their own open or proprietary software based on the run-time synthesis engine. In fact, it has been integrated into ATR XIMERA [30] and Festival as an spectrum and prosody prediction modules and one of the speech synthesis modules, respectively. Although no text analyzers have been included, Festival (general) or the Galatea toolkit (Japanese) can be used. Of course users can use their own text analyzers. For example, Krstulovic et al. [31] used the text analysis provided by the MARY software [32] instead of the Festival. This toolkit has been used in various research groups to develop their own HMM-based speech synthesis systems [33–46].

There have been a variety of functional restrictions in HTS version 1.x releases. However, HTS version 2.0 has more flexibility and a number of new functions which we have proposed. The next section describes the detail of HTS version 2.0.

2.2. New features in version 2.0

After an interval of three years, HTS version 2.0 was released in December 2006. This is a major update and includes a number of new features and fixes, such as

- Based on HTK-3.4.
- Support GCC-4.
- Compilation without signal processing toolkit (SPTK).
- Terms about redistributions in binary form are added to the HTS license.
- `HCompV` (global mean and variance calculation tool) accumulates statistics in double precision. For large databases the previous version often suffered from numerical errors.
- `HRest` (Baum-Welch re-estimation tool for a single HMM) can generate state duration PDFs [13, 14] with the `-g` option.
- Phoneme boundaries can be given to `HERest` (embedded Baum-Welch re-estimation tool) using the `-e` option. This can reduce computational cost and improve phoneme segmentation accuracy [47]. We may also specify subset of boundaries (e.g, pause positions).
- Reduced-memory implementation of decision tree-based context clustering in `HHEd` (a tool for manipulating HMM definitions) with the `-r` option. For large databases the previous versions sometimes consumed huge memory.
- Each decision tree can have a name with regular expressions (`HHEd` with the `-p` option).
  e.g.,

  ```
  TB 000 {(*-a+*,*-i+*).state[2]}
  TB 000 {(*-t+*,*-d+*).state[3]}
  ```

  As a result, two different trees can be constructed for consonants and vowels respectively.
- Flexible model structures in `HMGenS` (speech parameter generation tool). In the previous versions, we assumed that the first HMM stream is mel-cepstral coefficients and the others are for $\log F_0$. Now we can specify model structures using the configuration variables `PDFSTRSIZE` and `PDFSTRORDER` Non-left-to-right model topologies (e.g., ergodic HMM), Gaussian mixtures, and full covariance matrices are also supported.
- Speech parameter generation algorithm based on the expectation-maximization (EM) algorithm (the Case 3 algorithm in [15], please refer to Section 2.4 for detail) in `HMGenS`. Users can select generation algorithms using the `-c` option.

[1] The HTK license prohibits redistribution and commercial use.

- Random generation algorithm [48] in HMGenS. Users can turn on this function by setting a configuration variable RNDPG=TRUE.
- State or phoneme-level alignments can be given to HMGenS.
- The interface of HMGenS has been switched from HHEd-style to HERest-style.
- Various kinds of linear transformations for MSD-HMMs in HERest.
  - Constrained and unconstrained maximum likelihood linear regression (MLLR) based adaptation [49].
  - Adaptive training based on constrained MLLR [49].
  - Precision matrix modeling based on semi-tied covariance matrices [50].
  - Heteroscedastic linear discriminant analysis (HLDA) based feature transform [51].
  - Phonetic decision trees can be used to define regression classes for adaptation [52,53].
  - Adapted HMMs can be converted to the run-time synthesis engine format.
- Maximum a posteriori (MAP) adaptation [54] for MSD-HMMs in HERest.
- Speed improvements in many parts.
- Many bug fixes.

The most significant new features are speaker adaptation for MSD-HMMs and the speech parameter generation algorithm based on the EM algorithm. In the following section, we describe these features in more detail.

2.3. Adaptation and adaptive training

As discussed in Section 1, one of the major advantages of the HMM-based speech synthesis approach over the unit-selection approach is its flexibility: we can easily modify its voice characteristics, speaking style, or emotions by transforming HMM parameters appropriately.

Speaker adaptation is the most successful example. By adapting HMMs with only a small number of utterances, we can synthesize speech with voice characteristics of a target speaker [17, 18]. MLLR and MAP-based speaker adaptation for single-stream HMMs have been supported since HTK-2.2. However, we could not support them in the official HTS releases because our internal implementation of adaptation for multi-stream MSD-HMMs was not portable. In HTK-3.4 alpha, most of adaptation-related parts in HTK were rewritten. This change made porting adaptation for multi-stream MSD-HMMs straightforward.

In HTS version 2.0, MLLR mean (MLLRMEAN), diagonal variance (MLLRVAR), full variance (MLLRCOV), and constrained mean and variance (CMLLR) adaptations for MSD-HMMs are implemented. Unfortunately, adaptation of state duration PDFs [55, 56] is not supported yet. MAP estimation for mixture weights, means, variances, and transition probabilities are also supported. In addition, HTS version 2.0 includes adaptive training (CMLLR) [49], semi-tied covariance matrices [50], and HLDA, which have recently been implemented in HTK. The use of adaptive training enables us to estimate better canonical models for speaker adaptation and improves the performance of the average voice-based speech synthesis system [57]. Recently semi-tied covariance models were applied to HMM-based speech synthesis and we have achieved some improvement over diagonal covariance models if it is used with the speech parameter generation algorithm considering global variance [58]. These efficient full covariance modeling methods (would) become essential when we want to model highly correlated features such as articulatory movements. The use of HLDA enables us to derive a linear projection that best decorrelates training data associated with each particular class [51]. Although HLDA may not be effective in speech synthesis, it would be beneficial in recognition tasks.

Usually, MLLR transforms are shared across similar Gaussian distributions clustered by a regression class tree [59]. However, this method has a disadvantage: we can adapt segment level features only [60]. This is because the regression class tree is constructed based on a distribution distance in a bottom-up fashion and does not reflect connections between distributions on the time axis. To address this problem, phonetic decision trees have been applied to define regression classes [52, 53]. This enables us to adapt both segmental and suprasegmental features, and in this way significant improvements over the regression class trees have been reported. In HTS version 2.0, HHEd has a command DT for converting phonetic decision trees into a regression class tree. Converted decision trees can be used as a regression class tree to estimate MLLR transforms.[2]

To use adaptation transforms in synthesis, we can use both HMGenS and hts_engine. HMGenS can load and apply adaptation transforms in the same way used in HERest. For hts_engine, first model sets are transformed by adaptation transforms using the AX command of HHEd. Then adapted model sets are converted into the hts_engine format using the CT and CM commands.[3]

2.4. Speech parameter generation algorithm based on the EM algorithm

In [15], three types of speech parameter generation algorithms are described. These algorithms aim to solve the following three problems

Case 1. Maximize $P(\boldsymbol{o} \mid \boldsymbol{q}, \boldsymbol{i}, \lambda)$ w.r.t. $\boldsymbol{o}$,

Case 2. Maximize $P(\boldsymbol{o}, \boldsymbol{q}, \boldsymbol{i} \mid \lambda)$ w.r.t. $\boldsymbol{q}$, $\boldsymbol{i}$, and $\boldsymbol{o}$,

Case 3. Maximize $P(\boldsymbol{o} \mid \lambda)$ w.r.t. $\boldsymbol{o}$,

under the constraints between static and dynamic features ($\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$), where λ is an utterance HMM and corresponding state duration models, $\boldsymbol{o} = \left[\boldsymbol{o}_1^\top, \dots, \boldsymbol{o}_T^\top\right]^\top$ is a speech parameter trajectory including both static and dynamic features, $\boldsymbol{c} = \left[\boldsymbol{c}_1^\top, \dots, \boldsymbol{c}_T^\top\right]^\top$ is a static feature vector sequence, $\boldsymbol{W}$ is a window matrix to calculate dynamic features from static features, $\boldsymbol{q} = \{q_1, \dots, q_T\}$ is a state sequence, $\boldsymbol{i} = \{i_1, \dots, i_T\}$ is a mixture component sequence, and T is the number of frames. For Case 1, it is simply required to solve a set of linear equations. However, recursive search and EM algorithm-based iterative optimization are required for Cases 2 and 3 respectively.

In the previous versions, only the algorithm for Case 1 was implemented: state and mixture component sequences were as-

[2] In the speaker adaptation demo script released with HTS version 2.0, this function is turned off to reduce computational complexity.

[3] Covariance matrices of adapted model sets are approximated by their diagonal elements.

sumed to be provided. In HTS version 2.0, we have additionally implemented the algorithm for Case 3,[4] in which we assume that the state and mixture component sequences or a part of them are hidden. We can select the algorithm to be used using the `-c` option. If the `-c 0` option is specified, the Case 1 algorithm is used (both $\boldsymbol{q}$ and $\boldsymbol{i}$ are given). If `-c 1`, the Case 3 algorithm with a fixed state sequence is used ($\boldsymbol{q}$ is given but $\boldsymbol{i}$ is hidden). With the `-c 2` option, the Case 3 algorithm is used (both $\boldsymbol{q}$ and $\boldsymbol{i}$ are hidden). It should be noted that although the Case 1 algorithm cannot use Gaussian mixtures, it is much more computationally efficient than the Case 2 and Case 3 algorithms.

2.5. Demonstrations and documentation

HTS version 2.0 comes with two demo scripts for training speaker-dependent systems (English and Japanese) and a demo script for a speaker-adaptation system (English). The English demo scripts use the CMU ARCTIC databases and generate model files for Festival and `hts_engine`. The Japanese demo script uses the Nitech database and generates model files for the Galatea toolkit. These scripts demonstrate the training processes and the functions of HTS. We recommend that users first try to run these demos and read the scripts themselves. Six voices for Festival trained by the CMU ARCTIC databases have also been released. Each HTS voice consists of model files trained by the demo script, and can be used as a voice for Festival without any other HTS tools.

Currently no documentation for HTS is available. However, the interface and functions of HTS are almost the same as those of HTK. Therefore, users who are familiar with HTK can easily understand how to use HTS. The manual of HTK [24] is also very useful. Most of questions we have been asked have their answers in this manual. There is an open mailing list for the discussion of HTS (`hts-users@sp.nitech.ac.jp`). If you have any questions or trouble with HTS, please first search the mailing list archive and read the HTK manual, and then ask on the mailing list.

3. Other applications

Although HTS has been developed to provide a research platform for HMM-based speech synthesis, it has also been used in various other ways, such as

- Human motion synthesis [61–63],
- Face animation synthesis [64],
- Audio-visual synthesis and recognition [65–67],
- Acoustic-articulatory inversion mapping [68],
- Prosodic event recognition [69,70],
- Very low-bitrate speech coder [71],
- Acoustic model adaptation for coded speech [72],
- Training data generation for ASR systems to obtain domain-specific acoustic models [73].
- Automatic evaluation of ASR systems [74].
- Online handwriting recognition [75].

We hope that HTS will contribute to progress in other research fields as well as speech synthesis.

4. Conclusions and future release plans

This paper described the details of the HMM-based speech synthesis system (HTS) version 2.0. This version includes a number of new features and fixes such as adaptation, adaptive training, and the speech parameter generation algorithm based on the EM algorithm.

Internally, we have developed a number of variants of HTS, e.g.,

- Hidden semi-Markov models (HSMMs) [76].
- Speech parameter generation algorithm considering global variance [58].
- Variational Bayes [77].
- Trajectory HMMs [78].
- Interpolation [19,20].
- Shared tree construction [79].
- Advanced adaptation and adaptive training [80,81].
- Eigenvoice [21].
- Multiple linear regression HMMs [22].

Some of these have been applied to our Blizzard Challenge systems and achieved successful results [7]. Hopefully, we can integrate valuable features of these variants into future HTS releases. The current plan for future releases is as follows:

- Version 2.0.1 (August 2007)
  - Bug fixes.
  - C/C++ API for `hts_engine`.
  - Speaker interpolation.
- Version 2.1 (March 2008)
  - HSMM training and adaptation.
  - Speech parameter generation algorithm considering global variance.
  - Advanced adaptation.

HTS version 2.1, with the STRAIGHT analysis/synthesis technique [82], will provide the ability to construct the state-of-the-art HMM-based speech synthesis systems developed for the past Blizzard Challenge events [7,83].

5. Acknowledgments

The authors are grateful to Dr. Korin Richmond of the university of Edinburgh for his helpful comments on earlier drafts of this paper. We also thank people who contributed to the development of HTS version 2.0. This work was partly supported by the MEXT e-Society project, the Japan Society for the Promotion of Science (No.18800019), and the Hori information science promotion foundation.

6. References

[1] A.W. Black and P. Taylor, "CHATR: a generic speech synthesis system," in *Proc. COLING94*, 1994.

[2] A. Hunt and A.W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, 1996, pp. 373–376.

[3] R.E. Donovan and P.C. Woodland, "Automatic speech synthesizer parameter estimation using HMMs," in *Proc. ICASSP*, 1995, pp. 640–643.

[4] Only `HMGenS` provides algorithm for Case 3.

[4] E. Eide, A. Aaron, R. Bakis, W. Hamza, M. Picheny, and J. Pitrelli, "A corpus-based approach to <AHEM/> expressive speech synthesis," in *Proc. ISCA SSW5*, 2004.

[5] A.W. Black, "Unit selection and emotional speech," in *Proc. Eurospeech*, 2003, pp. 1649–1652.

[6] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. Eurospeech*, 1999, pp. 2347–2350.

[7] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 1, pp. 325–333, Jan. 2007.

[8] Z.-H. Ling, Y.-J. Wu, Y.-P. Wang, L. Qin, and R.-H. Wang, "USTC system for Blizzard Challenge 2006 an improved HMM-based speech synthesis method," in *Blizzard Challenge Workshop*, 2006.

[9] A.W. Black, H. Zen, and K. Tokuda, "Statistical parametric speech synthesis," in *Proc. ICASSP*, 2007, pp. 1229–1232.

[10] J. Yu, M. Zhang, J. Tao, and X. Wang, "A novel HMM-based TTS system using both continuous HMMs and discrete HMMs," in *Proc. ICASSP*, 2007, pp. 709–712.

[11] T. Fukada, K. Tokuda, Kobayashi T., and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. ICASSP*, 1992, pp. 137–140.

[12] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Hidden Markov models based on multi-space probability distribution for pitch pattern modeling," in *Proc. ICASSP*, 1999, pp. 229–232.

[13] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration modeling for HMM-based speech synthesis," in *Proc. ICSLP*, 1998, pp. 29–32.

[14] H. Zen, T. Masuko, T. Yoshimura, K. Tokuda, T. Kobayashi, and T. Kitamura, "State duration modeling for HMM-based speech synthesis," *IEICE Trans. on Inf. & Syst.*, vol. E90-D, no. 3, pp. 692–693, 2007.

[15] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, 2000, pp. 1315–1318.

[16] S. Imai, "Cepstral analysis synthesis on the mel frequency scale," in *Proc. ICASSP*, 1983, pp. 93–96.

[17] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Voice characteristics conversion for HMM-based speech synthesis system," in *Proc. ICASSP*, 1997, pp. 1611–1614.

[18] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," in *Proc. ICASSP*, 2001, pp. 805–808.

[19] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker interpolation in HMM-based speech synthesis system," in *Proc. Eurospeech*, 1997, pp. 2523–2526.

[20] M. Tachibana, J. Yamagishi, T. Masuko, and T. Kobayashi, "Speech synthesis with various emotional expressions and speaking styles by style interpolation and morphing," *IEICE Trans. Inf. & Syst.*, vol. E88-D, no. 11, pp. 2484–2491, 2005.

[21] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-based speech synthesis," in *Proc. ICSLP*, 2002, pp. 1269–1272.

[22] T. Nose, J. Yamagishi, and T. Kobayashi, "A style control technique for speech synthesis using multiple regression HSMM," in *Proc. Interspeech*, 2006, pp. 1324–1327.

[23] K. Tokuda, H. Zen, J. Yamagishi, T. Masuko, S. Sako, A.W. Black, and T. Nose, "The HMM-based speech synthesis system (HTS)," `http://hts.sp.nitech.ac.jp/`

[24] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X.-Y. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The Hidden Markov Model Toolkit (HTK) version 3.4*, 2006, `http://htk.eng.cam.ac.uk/`

[25] K. Shinoda and T. Watanabe, "Acoustic modeling based on the MDL criterion for speech recognition," in *Proc. Eurospeech*, 1997, pp. 99–102.

[26] A.W. Black, P. Taylor, and R. Caley, "The festival speech synthesis system," `http://www.festvox.org/festival/`

[27] J. Kominek and A.W. Black, "CMU ARCTIC databases for speech synthesis," Tech. Rep. CMU-LTI-03-177, Carnegie Mellon University, 2003.

[28] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Incorporation of mixed excitation model and postfilter into HMM-based text-to-speech synthesis," *IEICE Trans. Inf. & Syst. (Japanese Edition)*, vol. J87-D-II, no. 8, pp. 1563–1571, Aug. 2004.

[29] Galatea Project, "Galatea – An open-source toolkit for anthropomorphic spoken dialogue agent," `http://hil.t.u-tokyo.ac.jp/galatea/`

[30] H. Kawai, T. Toda, J. Yamagishi, T. Hirai, J. Ni, T. Nishizawa, M. Tsuzaki, and K. Tokuda, "XIMERA: A concatenative speech synthesis system with large scale corpora," *IEICE Trans. Inf. & Syst. (Japanese Edition)*, vol. J89-D, no. 12, pp. 2688–2698, Dec. 2006.

[31] S. Krstulovic, A. Hunecke, and M. Schroeder, "An HMM-based speech synthesis system applied to German and its adaptation to a limited set of expressive football announcements," in *Proc. of Interspeech*, 2007.

[32] M. Schröder and J. Trouvain, "The German text-to-speech synthesis system MARY: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, pp. 365–377, 2003.

[33] Y.-J. Wu and R.H. Wang, "HMM-based trainable speech synthesis for Chinese," *Journal of Chinese Information Processing*, vol. 20, no. 4, pp. 75–81, 2006.

[34] Y. Qian, F. Soong, Y. Chen, and M. Chu, "An HMM-based Mandarin Chinese text-to-speech system," in *Proc. of ISCSLP*, 2006.

[35] S.-J. Kim, J.-J. Kim, and M.-S. Hahn, "Implementation and evaluation of an HMM-based Korean speech synthesis system," *IEICE Trans. Inf. & Syst.*, vol. E89-D, pp. 1116–1119, 2006.

[36] C. Weiss, R. Maia, K. Tokuda, and W. Hess, "Low resource HMM-based speech synthesis applied to German," in *ESSP*, 2005.

[37] M. Barros, R. Maia, K. Tokuda, D. Freitas, and F. Resende Jr., "HMM-based European Portuguese speech synthesis," in *Interspeech*, 2005, pp. 2581–2584.

[38] A. Lundgren, *An HMM-based text-to-speech system applied to Swedish*, Master thesis, Royal Institute of Technology (KTH), 2005.

[39] T. Ojala, *Auditory quality evaluation of present Finnish text-to-speech systems*, Master thesis, Helsinki University of Technology, 2006.

[40] M. Vainio, A. Suni, and P. Sirjola, "Developing a Finnish concept-to-speech system," in *2nd Baltic conference on HLT*, 2005, pp. 201–206.

[41] B. Vesnicer and F. Mihelic, "Evaluation of the Slovenian HMM-based speech synthesis system," in *TSD*, 2004, pp. 513–520.

[42] S. Martincic-Ipsic and I. Ipsic, "Croatian HMM-based speech synthesis," *Journal of Computing and Information Technology*, vol. 14, no. 4, pp. 307–313, 2006.

[43] O. Abdel-Hamid, S. Abdou, and M. Rashwan, "Improving Arabic HMM based speech synthesis quality," in *Interspeech*, 2006, pp. 1332–1335.

[44] J. Latorre, K. Iwano, and S. Furui, "Polyglot synthesis using a mixture of monolingual corpora," in *ICASSP*, 2005, vol. 1, pp. 1–4.

[45] X. Gonzalvo, I. Iriondo, J. Socor, F. Alas, and C. Monzo, "HMM-based Spanish speech synthesis using CBR as F0 estimator," in *ITRW on NOLISP*, 2007.

[46] S. Chomphan and T. Kobayashi, "Implementation and evaluation of an HMM-based Thai speech synthesis system," in *Proc. of Interspeech*, 2007.

[47] D. Huggins-Daines and A. Rudnicky, "A constrained Baum-Welch algorithm for improved phoneme segmentation and efficient training," in *Proc. of Interspeech*, 2006, pp. 1205–1208.

[48] K. Tokuda, H. Zen, and T. Kitamura, "Reformulating the HMM as a trajectory model," in *Proc. Beyond HMM – Workshop on statistical modeling approach for speech recognition*, 2004.

[49] M.J.F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.

[50] M.J.F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[51] M.J.F. Gales, "Maximum likelihood multiple projection schemes for hidden Markov models," *IEEE Trans. Speech & Audio Process.*, vol. 10, no. 2, pp. 37–47, 2002.

[52] J. Yamagishi, M. Tachibana, T. Masuko, and T. Kobayashi, "Speaking style adaptation using context clustering decision tree for HMM-based speech synthesis," in *Proc. ICASSP*, 2004, pp. 5–8.

[53] M. Tachibana, J. Yamagishi, T. Masuko, and T. Kobayashi, "A style adaptation technique for speech synthesis using HSMM and suprasegmental features," *IEICE Trans. Inf. & Syst.*, vol. E89-D, no. 3, pp. 1092–1099, 2006.

[54] J.L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. on Speech & Audio Process.*, vol. 2, no. 2, pp. 291–298, 1994.

[55] J. Yamagishi and T. Kobayashi, "Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 2, pp. 533–543, 2007.

[56] J. Yamagishi, K. Ogata, Y. Nakano, J. Isogai, and T. Kobayashi, "HSMM-based model adaptation algorithms for average-voice-based speech synthesis," in *Proc. ICASSP*, 2006, pp. 77–80.

[57] J. Yamagishi, *Average-Voice-Based Speech Synthesis*, Ph.D. thesis, Tokyo Institute of Technology, 2006.

[58] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 5, pp. 816–824, 2007.

[59] M.J.F. Gales, "The generation and use of regression class trees for MLLR adaptation," Tech. Rep. CUED/F-INFENG/TR263, Cambridge University, 1996.

[60] M. Ostendorf and I. Bulyko, "The impact of speech recognition on speech synthesis," in *Proc. the IEEE Workshop on Speech Synthesis*, 2002, CD-ROM proceeding.

[61] K. Mori, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura, "Motion generation for Japanese finger language based on hidden Markov models," in *Proc. FIT*, 2005, vol. 3, pp. 569–570, (in Japanese).

[62] N. Niwase, J. Yamagishi, and T. Kobayashi, "Human walking motion synthesis with desired pace and stride length based on HSMM," *IEICE Trans. Inf. & Syst.*, vol. E88-D, no. 11, pp. 2492–2499, 2005.

[63] G. Hofer, H. Shimodaira, and J. Yamagishi, "Speech driven head motion synthesis based on a trajectory model," in *Proc. SIGGRAPH*, 2007, (submitted).

[64] O. Govokhina, G. Bailly, G. Breton, and P. Bagshaw, "TDA: a new trainable trajectory formation system for facial animation," in *Proc. Interspeech*, 2006, pp. 1274–1247.

[65] M. Tamura, S. Kondo, T. Masuko, and T. Kobayashi, "Text-to-audio-visual speech synthesis based on parameter generation from HMM," in *Proc. Eurospeech*, 1999, pp. 959–962.

[66] S. Sako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "HMM-based text-to-audio-visual speech synthesis," in *Proc. ICSLP*, 2000, pp. 25–28.

[67] T. Ishikawa, Y. Sawada, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura, "Audio-visual large vocabulary continuous speech recognition based on early integration," in *Proc. FIT*, 2002, pp. 203–204, (in Japanese).

[68] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Proc. of Interspeech*, 2006, pp. 577–580.

[69] K. Emoto, H. Zen, K. Tokuda, and T. Kitamura, "Accent type recognition for automatic prosodic labeling," in *Proc. Autumn Meeting of ASJ*, 2003, vol. I, pp. 225–226, (in Japanese).

[70] H.-L. Wang, Y. Qian, F.K. Soong, J.-L. Zhou, and J.-Q. Han, "A multi-space distribution (MSD) approach to speech recognition of tonal languages," in *Proc. of Interspeech*, 2006, pp. 125–128.

[71] T. Hoshiya, S. Sako, H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Improving the performance of HMM-based very low bitrate speech coding," in *Proc. ICASSP*, 2003, vol. 1, pp. 800–803.

[72] K. Tanaka, S. Kuroiwa, S. Tsuge, and F. Ren, "An acoustic model adaptation using HMM-based speech synthesis," in *Proc. NLPKE*, 2003, vol. 1, pp. 368–373.

[73] M. Ishihara, C. Miyajima, N. Kitaoka, K. Itou, and K. Takeda, "An approach for training acoustic models based on the vocabulary of the target speech recognition task," in *Proc. Spring Meeting of ASJ*, 2007, pp. 153–154, (in Japanese).

[74] R. Terashima, T. Yoshimura, T. Wakita, K. Tokuda, and T. Kitamura, "An evaluation method of ASR performance by HMM-based speech synthesis," in *Proc. Spring Meeting of ASJ*, 2003, pp. 159–160, (in Japanese).

[75] L. Ma, Y.-J. Wu, P. Liu, and F. Soong, "A MSD-HMM approach to pen trajectory modeling for online handwriting recognition," in *Proc. ICDAR*, 2007.

[76] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "A hidden semi-Markov model-based speech synthesis system," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 5, pp. 825–834, 2007.

[77] Y. Nankaku, H. Zen, K. Tokuda, T. Kitamura, and T. Masuko, "A Bayesian approach to HMM-based speech synthesis," in *Tech. rep. of IEICE*, 2003, vol. 103, pp. 19–24, (in Japanese).

[78] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences," *Computer Speech & Language*, vol. 21, no. 1, pp. 153–173, 2006.

[79] J. Yamagishi, M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "A context clustering technique for average voice models," *IEICE Trans. Inf. & Syst.*, vol. E86-D, no. 3, pp. 534–542, 2003.

[80] J. Isogai, J. Yamagishi, and T. Kobayashi, "Model adaptation and adaptive training using ESAT algorithm for HMM-based speech synthesis," in *Proc. Interspeech*, 2005, pp. 2597–2600.

[81] Y. Nakano, M. Tachibana, J. Yamagishi, and T. Kobayashi, "Constrained structural maximum a posteriori linear regression for average-voice-based speech synthesis," in *Proc. Interspeech*, 2006, pp. 2286–2289.

[82] H. Kawahara, I. Masuda-Katsuse, and A.de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F_0 extraction: possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, pp. 187–207, 1999.

[83] H. Zen, T. Toda, and K. Tokuda, "The Nitech-NAIST HMM-based speech synthesis system for the Blizzard Challenge 2006," in *Blizzard Challenge Workshop*, 2006.

ECIRCUS: Building Voices for Autonomous Speaking Agents

Christian Weiss, Luis C. Oliveira, Sergio Paulo, Carlos Mendes, Luis Figueira [1], *Marco Vala, Pedro Sequeira, Ana Paiva*[2], *Thurid Vogt, Elisabeth Andre*[3]

[1] INESC-ID/IST, Spoken Language Systems Laboratory, Lisbon, Portugal
[2] INESC-ID/IST, GAIPS, Lisbon, Portugal
[3] Institute of Computer Science, University of Augsburg, Germany
{christian.weiss, lco}@l2f.inesc-id.pt

Abstract

This paper describes our work integrating automatic speech generation into a virtual environment where autonomous agents are enabled to interact by natural spoken language. The application intents to address bullying problems for children aged 9-12 in the UK and Germany by presenting improvised dramas and by asking the user to act as an "invisible friend" of the victimised character. As we are addressing an elementary school environment one specification of the resulting voice was building age-corresponding young school kids voices. The second specification addresses building a low-resource speech generation system which is capable to run on older school computers but is still fast enough in response time to guaranty a fluent conversation between the agents. Third requirement was integrating the speech-module with the agents. We focus on the speech generation system itself, pointing out possible implementation issues in building non-controlled speech interaction in virtual environments Furthermore we describe the problems arising in building unit-selection based child's' voice TTS and shows alternative methods to child's voice recording by deploying voice transformation methods.

Index Terms: Social learning and education, speech synthesis, spoken interaction

1. Introduction

Virtual animated characters in dramatized scenarios are no longer used only on computer games. Learning and educative environments can benefit from the ever growing familiarity of users with virtual environments.

The eCircus (Education through Characters with Interactive Role-playing Capabilities that Understand Social interaction) project is an ongoing interdisciplinary EU-project focusing on novel conceptual models and innovative technology to support social and emotional learning through role-play and affective engagement for Personal and Social Education. Main aspects are to create a virtual environment for emotional and social learning focusing on the domains of bullying and refugee integration in school [1]. This paper describes our work in integrating an automatic speech generation module into the first showcase of the technology developed in this project, a virtual learning environment on anti-bullying education, called FearNot!. In this application autonomous agents need to communicate with each other in a away understandable by the user. The inter-agent communication using speech acts is converted into either English or German by a language generator engine that is translated into speech using a speech synthesizer. Figure 1 shows a screenshot of a bullying scenario in FearNot!. Although the 3D animated synthetic characters are cartoon like figures, our previous work showed that the users expect them to have naturally sounding voices [2].

As we are addressing an elementary school environment with students at the age between 9 and 12 years old, one specification of the generated voice was building an age-corresponding young school kids voice. The second specification addresses building a low-resource speech generation system which is capable to run on older school computers but is still fast enough in response time to guaranty a fluent conversation between the agents and the user. Third requirement was including audio-visual synchronization with the agents' actions.

This paper is organized as follows. In section 2 we address the problems arising while building a unit-selection based child voice and point out the difficulties and show our solution. In section 3 we describe our implementation of the voice building software and focus on the integration of the various modules usually needed by speech synthesis systems. The next section describes the experiment that was conducted to evaluate the system and its results. The final section presents the conclusions and the planned future work.

Figure 1: Screenshot of a FearNot! scenario

2. Child Voices

When trying to produce voices for child like characters the first approach that comes to mind is to record real children voices. We started by recording a set of 100 English sentences by a 9 year old girl and a boy of age 10. Although these recordings were very useful for our analysis of the acoustics of children's speech it soon became obvious that the recording of a larger set of sentences would be impractical. Children require shorter recording sessions and at slower pace than an adult speaker. It is also more difficult to assure the same speaking style among recording sessions since it depends on the child mood in that specific day. Given this difficulties it was decided to record carefully selected adults and modify their voices to make them sound as children's voices. To select the voice talents and to understand what type of modifications were required, we analysed our own recordings (table 1) and in general confirmed the results published in [3].

Table 1: Parameters from our own recordings.

Boy avg. F0 (Hz)	Boy avg. Formant values	Girl avg. F0 (Hz)	Girl avg. Formant values (Hz)
270	570 1400 2700 3900	280	570 1800 3000 4100

The main characteristic that distinguishes children's voices from adult voices results from the smaller size of their vocal tract. This results in higher pitched voices due to shorter vocal folds and in the scaling of the formants as a result of a shorter vocal tract.

The most significant changes in f0 occur for male speaker from age 12 to 15 resulting in f0 dropping from an average value of 226 Hz at age 12 to a value of 127 Hz at age 15. This drop is much smaller in female speakers with no significant pitch changes after age 12, with an average f0 of 231 Hz. For our target age of 10, the average f0 for boys is around 260 Hz while girls have an average value of around 270 Hz. This suggested the use of female adult voices as a base for the voice of children of both genders.

The analysis of formant frequencies shows a clear linear scaling trend as a result of the axial growth of the vocal tract [3]. The main gender difference is that the scaling factors of male speakers are approximately the same for all formants while each formant of the female speakers evolves differently as a function of age. Since the formant scaling factor from an adult male voice to an adult female voice is, on average, 30%, female voices are also in this respect better for being transformed into children’s voices. This way, using the data in [3], the average scaling factor from an adult female voice to a voice of a 10 years old boy, would be of around 10% for all formants. The average scaling values for a voice of a girl of age 12 would be 20% for F1, 15% for F2 and 10% for F3.

Taking into account these results it was decided to search for voice donors with the following characteristics: females of small stature, corresponding to a small vocal tract, with experience in interacting with children of the target age, without strong social or regional accent and with the ability to produce the required intonation in a regular way. The selected speakers were two English teachers of children of age 10. The recording tests showed that they were able to produce the required intonation patterns and that their voices could be modified by both the PSOLA technique [4] and spectral scaling with little distortion. By applying different small scaling factors to both f0 and formant frequencies, we could produce voices for the different synthetic characters. For the German version a female and a male voice were recorded. As expected, the pitch of the male voice could not be changed to the values usually observed in children’s voices but informal tests showed that the modified voices were acceptable for cartoon like characters.

3. Voice Building Process

The speech corpus for the recordings was built based on the language engine that converts into English or German the speech acts used for the communication between agents. The input text of the synthesizer is thus limited by the variability of the text generated by the language engine. This suggests the use of a limited domain speech synthesizer [5][6].

To create the inventory required to synthesize the utterances spoken by the characters we started by modifying the language engine to generate all the possible sentences. This resulted in a total 7496 sentences, with 1206 distinct words. A greedy algorithm was used to select a subset of these sentences with full word coverage, distinguishing words in the middle of intonational phrases and words close to prosodic boundaries. The greedy algorithm selected 552 sentences for the English inventory. A similar procedure was applied to the German language engine generating a total of 4690 sentences, with 1557 distinct words, from which the greedy algorithm selected an inventory of 622 sentences.

The two selected voice donors for each language recorded all the sentences of the English inventory. The recordings were conducted in the sound proof booth of INESC-ID and the speakers were asked to read the prompts with some, but not excessive, expressiveness. The recordings required four sessions of 2 hours for each speaker.

3.1. Integrated Voice Building

Building a voice for a TTS is a non-trivial task as needs a lot of pre-processing steps. In order to remove errors and repetitions from the utterances' orthographic transcriptions, the text prompts were manually verified. Then, they were automatically split into prosodic phrases by using the MuLAS system [7] so that every single file contains only one prosodic phrase. The resulting 552 phrases were then automatically segmented by our own phonetic segmenter [8] that was specifically adapted for British English. Gender dependent models were trained using the British English WSJ corpus, which reached 85% and 84% of accuracy at 20 milliseconds for female and male speakers, respectively. A speaker adaptation procedure was performed 2 times, by using the canonical word pronunciations for the segmentation stage. At the 3rd iteration, the segmenter was provided with a pronunciation graph accounting for the canonical pronunciations together with some alternative pronunciation raised by the post-lexical rules.

Using a multi-level unit inventory we are able to generated new words which are not occurring in the recorded speech corpus. We call this approach a semi-limited domain synthesis while not all words existing in on language can be reproduced.

Our voice building software is capable of building voice inventories using only the label-files which include the segment start time, the word and syllable boundaries as well as syllable stress information. Furthermore we need the according utterance-files and the recorded audio-files. Once all files are gathered an automatic process starts and builds a context depended voice inventory stored as a XML based representation of each label, utterance and audio- file. Please see section 3.1.1 for a detailed description of the XML representation.

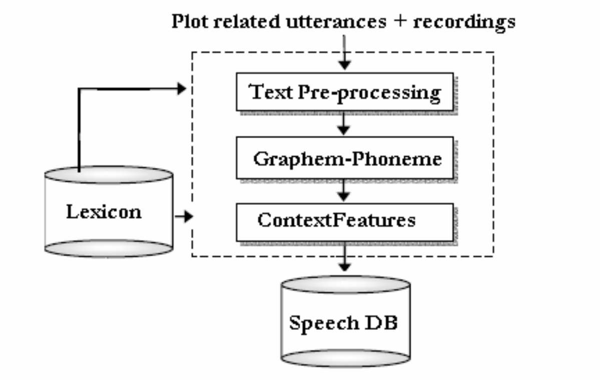


Figure 2: Diagram of the voice building system.

Figure2 shows a schematic flow-chart of the steps which were applied during the voice building process. These steps are:

- Text normalization
- Acoustic and spectral parameter extraction; Duration, F0, MFCC
- Extracting phonological and quantitative features.
- Grapheme-Phoneme conversion

For F0 and spectral feature extraction we use standard tools like the Snack-library and HTK. For dynamic feature prediction we use conditional log-linear models, please see section 3.2.

3.1.1. XML –Representation

The XML based structure consists of the features as listed below:

Table 2: Overview of features

Unit	Feature
Word	Preceding, following word Sentence type Distance left/right in sentence POS Duration, log duration Average F0, log F0 First/last frame MFCCs
Syllable	Preceding, following syllable Distance left/right word, sentence Stress Duration, log duration Average F0, log F0 First/last Frame MFCCs
Phone	preceding, following phone distance left/right word, syllable Duration, log duration average F0, log F0 First/last Frame MFCCs

Once we extracted all features which are describing the segments we build a temporarily XML-based left-right context representation of the utterance and store this information in a voice inventory database.

3.2. Conditional Log-Linear Models for Dynamic Feature Prediction

For Grapheme-Phoneme conversion, Part-of-Speech Tagging syllable boundary detection, as well as for duration and F0 predicting we applied conditional log-linear models also known as Maximum-Entropy models [9], [10].

The conditional log-linear model framework is a well known approach for ambiguities resolution in natural language processing where many problems can be reformulated as a classification problem. The task of such a reformulation is to include a context and to predict a correct class. The objective is to estimate a function $X \to Y$, which predicts an object $x \in X$ to its class $y \in Y$. Y represents the predefined classes for either each task of our prediction problem.

In the field of stress prediction we are dealing with a binary classification where the class is true for stressed syllables and false for non-stressed.

The same binary classification task has to be solved in the domain of syllabification where we have a syllable boundary or not.

X consists of quantitative and phonological features where we include the context and the resulting input for the classification. The classifier $X \to Y$ can be seen as a conditional probability model in the sense of

$$C(x) = \arg\max_{y} p(y|x) \qquad (1)$$

where x is the object to be classified and y is the class. Including the context we get a more complex classifier

$$C(x_1, x_2, ..., x_n) = \arg\max_{y_1 ... y_n} \prod_{i=1}^{n} p(y_i | x_1 ... x_n, y_1 ... y_{i-1}) (2)$$

where $x_1 ... x_n, y_1 ... y_{i-1}$ is the context at the i^{th} decision and y_i is the outcome.

This model we use in all our dynamic feature prediction tasks during the offline voice building process as well as during runtime.

3.3. Acoustic Synthesis with F0 Smoothing

The acoustic synthesis module follows the variable-size unit selection algorithm. We apply a pre-selection strategy while the algorithm tries to find a segment that matches the predefined target structure in a left-right context. If this does not result in any found segment we simplify the structure matching but keep the left-right context. When no segment is found at the word-level, the algorithm searches for syllable segments and, as a last alternative, a phoneme-level segment selection is performed.

Using a predefined structure matching for segment selection we save computational resources in target and join-cost distance calculation. The target distance calculation is done by summing the differences between the values of the features of the selected and of the target segment. Some kind of normalization is needed given the different ranges of the feature values (for example, the log F0 and the duration values). This normalization is done using the following equation:

$$normcost = \frac{x^2}{1 + x^2} \qquad (3).$$

where *x* is the difference between the values of the feature of the selected and of the target segment. The join cost calculation is done by a Euclidian distance measure between the successive frames MFCC's of the segments.

4. Experimental Evaluation

The experimental evaluation was conducted only on the English version of the synthesizer. The evaluation followed a procedure very similar to the one described in [11]. Given that we are not using real children's voices, one of the objectives was to check if the modified voices were acceptable for the FearNot! characters. Two types of tests were conducted: half of the subjects could only listen to the characters voices, while the other half watched movie clips with different animated characters (Figure 2). Although the lip movements were random, they were synchronized with the duration of the utterance making an acceptable illusion of lip synchronization given the small size of the characters mouth.

The subjects were asked to rate the utterances in terms of 6 factors: (1) overall sound quality (2) naturalness of the intonation (3&4) extent to which the utterance sounded like a boy or a girl (5&6) extent to which the utterance sounded like it was pronounced by the bully or the victim.

The stimuli were produced in 8 different versions: the original recordings of both speakers, synthesized speech using unmodified inventories of both speakers, one modified version for each speaker original recordings and synthesized speech using inventories of modified voices. Each subject was asked to rate a total of 48 stimuli. Like in [11] the ratings were on a Likert scale with 1 for very bad and 5 for very good. The test was conducted over the internet and the subjects used headphones. The results showed that the presence of video result in a better rating on the overall perceive quality: 3.42 (with a significance of $p<0.005$) vs 3.70 ($p<0.00001$). Without the video the overall rating of boy, girl, victim and bully was not significant ($p>0.05$). The presence of the animated character made the voices believable especially for the victim (3.68, $p<0.00001$). The modified voices had the same rating in overall quality as the unmodified voices for the audio only test (3.42, $p<0.04$) but were better rated when played in video clips (3.82, $p<0.00001$ vs 3.59, $p<0.009$). The results for the overall quality of both the modified and unmodified recording were above 4 (4.45, $p<0.00001$). The ratings for synthesized speech were not significant and the analysis of the results showed that although the evaluators agreed on some sentences (usually with score above 4) they did not agree on the rating to assign to sentences with noticeable concatenation discontinuities.

Figure 3: Image of one of the video clips used in the audio and video evaluation task.

5. Conclusion and Future Work

Limited domain synthesis allowed us to produce voices for 3D animated characters with almost natural speech quality as expected by the users of virtual learning environments. In order to minimize concatenation mismatches we asked the adult voice donors to refrain their expressiveness during the recordings. This affected mostly the bully character's voice that was found less credible, but with a sufficiently good rating. Although there was no story context in our evaluation, the video of the animated characters influenced positively the perceived overall quality and intonation.

Using the results of this study, we will now generate additional modified voices for the remaining characters of the FearNot! application. We also plan to correct some segmentation and concatenation problems detected during this evaluation, and to improve the voice modification algorithm by using a more robust epoch detector. The German language version of the system is also being developed. The effectiveness of the FearNot! application against bullying in schools will soon be fully investigated when the final version of the system is placed in schools in the UK and Germany for a large scale longitudinal evaluation.

6. Acknowledgements

This work was partially supported by European Community (EC) and is currently funded by the eCIRCUS project IST-4-027656-STP. The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein.

7. References

[1] Zoll, C., Enz, S., Schaub, H., Aylett, R., Paiva, A., "Fighting Bullying with the Help of Autonomous Agents in a Virtual School Environment", *7th International Conference on Cognitive Modelling*, Trieste, Italy, 2006

[2] Cabral, J. and Oliveira, L.C., Guilherme Coelho Barreira Raimundo, Ana Paiva, "What voice do we expect from a synthetic character?", *SPECOM*, pages 536-539, 2006.

[3] Lee, S., Potamianos, A. and Narayanan, S., "Acoustics of children's speech: Developmental changes of temporal and spectral parameters", *JASA*, 105:1455–1468, 1999.

[4] Charpentier, F., Moulines, E., "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones", *Eurospeech*, Paris, 1989.

[5] Black, A. and Lenzo, K., "Limited Domain Synthesis", *ICSLP*, Beijing, China, 2000.

[6] Schweitzer, A., Braunschweiler, N., Klankert, T., Möbius, B., Sauberlich, B., "Restricted unlimited domain synthesis", *Eurospeech*, 1321-1324, 2003.

[7] Paulo, S.G., and Oliveira, L.C., "MuLAS: A Framework For Automatically Building Multi-Tier Corpora", *Interspeech*, Antwerpen, 2007.

[8] Paulo, S.G. and Oliveira, L.C., "Generation of Word Alternative Pronunciations Using Weighted Finite State Transducers", *Interspeech*, pages 1157-1160, 2005.

[9] Berger, A., Della Pietra, S.A., Della Pietra, V.J., "A Maximum Entropy Approach to Natural Language Processing", *Computational Linguistics*, 22(1), 1996.

[10] Ratnarparkhi, A., *Maximum Entropy Models for Natural Language Ambiguity Resolution*, PhD Dissertation, University of Pennsylvania, 1998.

[11] Johnson, W.L. Narayanan, S. Whitney, R. Das, R. Bulut, M. LaBore, C., "Limited domain synthesis of expressive military speech for animated characters", *IEEE Workshop on Speech Synthesis*, September 2002.

Unit Selection Synthesis in the SmartWeb Project

Martin Barbisch, Grzegorz Dogil, Bernd Möbius, Bettina Säuberlich, Antje Schweitzer

Institute for Natural Language Processing
University of Stuttgart, Germany
{Firstname.Lastname}@ims.uni-stuttgart.de

Abstract

This paper describes three aspects of the unit selection synthesis used in the SmartWeb dialog system. The synthesis module has been implemented in the IMS German Festival speech synthesis system. First, we compare a unit selection strategy developed in the course of the project to a strategy developed earlier. Second, we discuss our experiences with F0 smoothing and amplitude modeling, which were both devised to reduce audible discontinuities. However, the results are inconclusive so far. Finally, we sketch a simple mechanism that addresses the problem of language disambiguation for proper names.

1. Introduction

SmartWeb is a research project funded by the German government [1]. The goal of the project is to implement a mobile intuitive user interface to the *Semantic Web* which allows requests involving natural speech and gestures. Answers are also rendered by speech, which is synthesized by the unit selection synthesis module described in this paper.

The synthesis module used in SmartWeb is based on the synthesis module developed in the predecessor project [2] and is implemented in the IMS German Festival framework [3].

In the course of the SmartWeb project, we have built two databases, one for a male speaker, and one for a female speaker. We have added a new unit selection strategy as an alternative to the existing strategy. Thus, there are two different unit selection algorithms available using the same database, text preprocessing and symbolic synthesis components. Both variants render very natural and intelligible speech. We compared the two variants in a first perception experiment to verify the validity of the new approach. The two variants and the perception experiment are described in some detail in section 2.

Although the synthesis results are very good altogether, there are some occasional glitches that seem to be caused by discontinuities in amplitude and pitch. We therefore experimented with amplitude modeling and different F0 discontinuity penalties. However, the results are inconclusive so far. The experiments and their results are discussed in section 3.

One key application of SmartWeb is the access to information on the soccer World Championships 2006. In this scenario, we faced the problem that proper names, particularly first names, are often ambiguous between several languages. We briefly sketch a simple mechanism to deal with this problem in section 4.

2. Comparing the two unit selection approaches

Both approaches combine aspects of two existing unit selection approaches, viz. phonological structure matching (PSM, [4]) and acoustic unit clustering (AC, [5]). We will call the first approach PSM/AC in the following because it combines PSM and AC in a straightforward way. The alternative approach will be called PSM/MC because in contrast to the original AC, the clustering is carried out manually.

2.1. PSM versus AC

The PSM algorithm [4] employs a top-down strategy for selecting the units from a speech database in which all sentences are represented as phonological tree structures. For each target sentence to be synthesized, the corresponding target tree structure is calculated. The PSM algorithm starts on the sentence level by comparing the available sentence tree structures to the target tree structure and possibly descends in the target tree structure until matching candidates are found. Generally, on any level a candidate matches if the trees below the target node match. If no adequate candidate is found on one level, the algorithm descends to the next lower level by assigning the daughters of the current node as new targets. This approach ensures that the longest available unit from the database is selected, minimizing concatenation points.

By contrast, the AC algorithm [5] only searches for candidates on the segment level. Longer continuous stretches of speech are only favored indirectly because they cause no concatenation costs later on. As the number of candidates is usually very high on the segment level, the candidates are clustered in an offline process. This is done automatically by creating a decision tree for each phoneme type with its leaves representing clusters of similar items. The features that are used for the questions at the nodes of the decision tree are linguistic-phonological features. The trees are built in a way that the acoustic similarity within the cluster is maximized, selecting only features that are significant in partitioning the tree. Thus, in building the tree, those features are determined that have the greatest impact on the acoustic realization.

The clusters can be pruned in order to obtain smaller clusters, by excluding segments that are farthest from the center of the cluster. This is intended to remove potentially poorly articulated or incorrectly labeled units. A second type of pruning is aimed at reducing units that are very common by removing units that are very similar to other existing units.

During the synthesis process, for each target segment the relevant cluster is determined by selecting the cluster which matches the desired linguistic-phonological context. The units belonging to that cluster are then taken as candidates.

The disadvantage of the AC algorithm is that in some cases the selection of continuous segments from the database is prohibited because they have either been assigned to a cluster which is not taken into consideration in the actual context, or because they have been removed during the pruning process.

Also during the construction of the decision trees no explicit linguistic and phonetic knowlegde is applied. For instance, it is impossible to give a higher priority to certain features, such as the manner and place of articulation of the context segments, which is expected to determine the strength and type of coarticulation effects.

The PSM algorithm on the other hand is problematic in open-domain scenarios because it does not restrict the number of candidates for each target unit. Particularly in open-domain scenarios, it will often be necessary to concatenate segment-level candidates because the database can not be tailored to cover all possible utterances by higher-level units. Since there will be very many segment candidates at least for the more frequent phonemes, the candidate network grows very large, reducing the efficiency of the algorithm.[1]

2.2. The PSM/AC approach

In the predecessor project to SmartWeb, we implemented a unit selection strategy combining PSM and AC by incorporating the strengths of both algorithms while avoiding their drawbacks in open-domain scenarios discussed above [2]. We call this approach the PSM/AC approach. The combination was motivated by the claim that PSM would prefer longer units in a more direct way than the AC approach, while clustering is appropriate to reduce candidate sets in cases where no long units are available.

Accordingly, PSM is used for phrase, word and syllable-sized units. If no appropriate candidates are available on the phrase, word, or syllable levels, AC is used on the segment level to reduce the segment candidate sets. This procedure ensures that at least longer units can be selected in their entirety; we do not run the risk that single segments within these units are not accessible because they have been assigned to another cluster or because they have been pruned during the clustering process.

2.3. The PSM/MC approach

The alternatively developed approach also employs the PSM strategy for candidate selection, but uses manual clustering (MC) to reduce the candidate sets on all levels, hence the name PSM/MC. The clustering is achieved by manually constructed decision trees. The use of decision trees on all unit levels allows for the consistent administration of all units and an efficient access via indexing.

The structure of the decision trees is given manually by ranking the features according to their linguistic-phonological relevance. The order of the ranking determines the questions at the nodes at each level of the decision trees. Each level of the tree represents a specific feature (e.g. place of articulation of the preceding or the next segment, or syllable stress, syllable position, etc.). The place of articulation of the segmental context is ranked very high in the decision tree as it is very important to model coarticulation effects.

The MC approach is highly flexible in that the decision tree can be easily reconstructed if a specific feature order turns out to be suboptimal. If no or only few candidates are found on a specific level, it is possible to collect all subordinate candidates on a higher level. Also the basic unit type can be selected freely (i.e. phone, diphone or demi-phone) with no further modification to the algorithm. This facilitates the comparison of the different basic unit types.

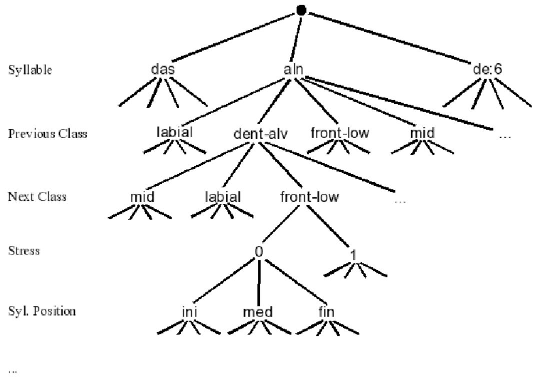


Figure 1: *Syllable level decision tree with exemplary feature ranking (left column). For each syllable type, this tree splits the candidates according to the place of articulation of the preceding and following segments (features "previous class" and "next class", respectively). Candidates are classified further according to the stress level of the syllable (feature "stress") and the position of the syllable in the phrase (feature "syl. position").*

The PSM/MC algorithm offers some advantages over PSM/AC. Firstly, its flexibility allows for comparing not only different basic unit types but also which phonetic features are most important for perception. The latter can be achieved by specifying different feature rankings for the phonetic features in question and rebuilding the decision trees with the respective order. This step requires no further manual interaction beyond the specification of the ranking and can easily be executed several times to test different rankings.

Secondly, PSM/MC usually does not run the risk of excluding or involuntarily ignoring potentially good candidates even before the selection process. Depending on the number of candidates, the selection process can be terminated on any level in the tree, selecting all candidates in the sub-trees beneath.

Thirdly, the clustering is adapted to the specific unit type and its requirements. This way phonetic knowledge can be directly applied in creating the decision trees. For instance, the place of articulation of the preceding segment is the primary selection criterion for all unit types. This is intended to model coarticulation effects, such as the influence of preceding labial consonants on the spectral properties of a vowel for instance, which would be expected to be different enough from the influence of, say, a preceding velar consonant to warrant the assignment of segments in these contexts to different clusters. On the syllable level, stress and the syllable's position in the corresponding phrase are important features. The high ranking of syllabic stress is motivated by the fact that it has been claimed to affect the spectral balance of the corresponding vowels [6, 7]. The position of the syllable in the phrase is expected to have an impact on the duration of the syllable and its segments as well as on their pitch level. Also, phrase-final segments and syllables are often laryngealized in German.

The disadvantage of PSM/MC lies in the statistically unbalanced distribution of the feature vectors in the corpus due to the

[1] For instance, on the segment level, our database contains 107,000 tokens representing 84 different German and foreign phonemes, which corresponds to an average of approx. 1,300 tokens per type, whereas on the syllable level, 41,000 tokens represent 3,350 syllable types, corresponding to an average of only 12 tokens per type.

LNRE characteristics of natural language [8], resulting in unbalanced decision trees. Since a few candidates are represented above average in the database, the trees exhibit large differences in the number of candidates at the leaf level, as units with identical feature vectors can not be differentiated and thus end up in identical clusters. Possible acoustic differences are not taken into account because MC only operates on the symbolic level, in contrast to AC, where the classification is driven by the signal.

2.4. Evaluation of PSM/AC versus PSM/MC

We compared both selection strategies in a first unsupervised perception experiment. We used the diphone-based version of PSM/MC because it was expected to model coarticulation effects better than the segment-based version. In this experiment, 26 subjects listened to 30 pairs of stimuli. The stimuli were 15 moderately long sentences (4 to 11 words) randomly selected from different text genres, synthesized using the two different algorithms and presented pairwise in different orders. Each pair could be played several times, but always in the same order. Listeners had to judge which stimulus sounded better, or if both stimuli sounded equally good, and they could take as long as they wanted to make their decision. In 22 cases, the stimuli in a pair were different (AB or BA order), and in 8 cases, they were identical. Participants were instructed that some stimuli would be identical. The identical pairs were included to assess the listeners' reliability.

Listeners favored PSM/AC over PSM/MC (49.8% vs. 40.7%, 9.4% undecided). The differences were statistically significant (χ^2(2,N=572)=154.03, p$\ll$0.05). The difference was not due to personal preferences, since only 3 participants consistently favored PSM/AC over PSM/MC (p<0.002)[2]. Instead, the differences were dependent on the stimulus pair: for 10 out of 22 pairs PSM/AC was rated significantly better, and for 6 of these pairs PSM/MC was rated significantly better (p<0.002)[3]. This means that for the majority of stimulus pairs, participants agreed in their judgment – they usually favored the same variant. The reason for this is that in some cases, the units selected from the database were not ideal realizations of the target unit, and that sometimes, the concatenation was suboptimal. These problems, which are typical for any unit selection algorithm, in some cases occurred in the PSM/AC stimulus, and in some cases in the PSM/MC stimulus, but the PSM/MC variant was affected slightly more often.

Altogether we consider the results of the evaluation encouraging enough to pursue the PSM/MC algorithm further, even more so because there are at least two aspects in which we are confident to improve the algorithm in the future.

First of all, an informal assessment of the specific problems in the PSM/MC stimuli suggests that the concatenation of diphones containing plosives was problematic in some cases, in that the corresponding stop releases could not be perceived properly. This is because our variant of the original optimal coupling algorithm [9] has been adapted to concatenate diphones by starting the search for a good concatenation point at the middle of the phoneme. The middle of the phoneme in case of stops is often close to the burst, and thus it happens occasionally that the burst is completely omitted when concatenating stops. One way to remedy this problem is to label the bursts in the database and to take the place of the burst into consideration when searching for the optimal concatenation point. Another way may be to modify the optimal coupling algorithm to detect the silence part of stops automatically. Compared to the first solution, this would eliminate the necessity to prepare the database beforehand.

Some additional improvement of the PSM/MC algorithm could be achieved by re-assessing the manually defined feature order in the selection trees. Although the current order was partly determined on the basis of phonetic knowledge, in some cases the ranking was not obvious and only preliminarily established by ad hoc decisions. These decisions might be reconsidered with the help of further perceptual evaluation procedures.

3. Prosodic modifications

In order to improve synthesis quality even further we investigated several possibilities for prosodic modifications. The motivation was that audible discontinuities seemed to be mostly caused by concatenation of prosodically too different candidates. Concerning pitch we experimented with different weights for the concatenation costs caused by pitch discontinuities. As for amplitude, we built a loudness model for each phoneme and adjusted actually selected segment candidates to fit those models.

3.1. Pitch Continuity

A smooth pitch contour is most important for intonation. Discontinuities of the pitch contour at unit boundaries cause audible glitches. In a first step, we investigated the influence of different weights for F0 differences in determining the concatenation costs.

The difference in F0 between consecutive units is already taken into account when calculating concatenation costs in Festival [9]. An additional weight factor has been added [10] to bring the costs caused by F0 differences into the same order of magnitude as the costs for spectral discontinuities. This weight factor has been predetermined for both the male and the female voice by synthesizing a large number of sentences and comparing the means for the spectral costs to the means for the F0 costs. The weight factor was chosen in a way that the same means are obtained for spectral costs and F0 costs. A second factor was defined in a configuration file which is intended to allow experimenting with different weight factors to give more or less priority to F0 continuity [10].

The effectiveness and usefulness of the newly introduced F0 weights were tested with three objective evaluation methods, varying the configurable weight factor to be 0, 1, 2, 3, or 5.

The first method was to compare the resulting F0 values with the idealistic F0 curve as predicted by the PaIntE model [11] by calculating the size of the area between the two curves:

$$curve_{RMSE} = \sqrt{\frac{\sum_{i=0}^{length(wave)} (f0(i) - f0_{PaIntE}(i))^2}{length(wave)}} \quad (1)$$

The smaller the area the better the F0 curve approximates the "optimum". However, the significance of this calculation depends on the quality of the reference curve and does not directly measure the smoothness of the F0 curve.

The second method was to determine an F0 curve "smoothness" correlate. The smoothness correlate was obtained by simply adding the absolute differences of consecutive F0 values in

[2] The significance level was adapted to p = 0.05/26 $\simeq$ 0.002 because of the 26 repeated χ^2 tests

[3] The significance level was adapted to p = 0.05/22 $\simeq$ 0.002 because of the 22 repeated χ^2 tests

the synthesized signal:

$$\sum_{i=0}^{n-1} |x_{i+1} - x_i| \qquad (2)$$

(where n is the number of frames and x the F0 value). If the smoothness increases with larger F0 weights, this is a good sign for fewer discontinuities in the F0 curve. However, this approach did not seem to be a good benchmark for the F0 cost function, since both increasing and decreasing smoothness were found for larger F0 weights, depending on the sentence synthesized.

The third method was to verify that different F0 weights did indeed have an effect in that they resulted in different candidates being selected, and to quantify the change by determining in how many cases different candidates were selected. The results confirmed that with increasing F0 weights, the number of different candidates increases as well. This was not generally the case; for some sentences, effects occurred only for F0 weight 3 or 5, while for others, there were changes even for an F0 weight of 1. This shows that the introduction of the additional weight factor successfully brings the F0 weights in an order of magnitude that is comparable to the weights applied to spectral differences. However, this does not answer the question whether the changes are positive or negative.

We conclude that a perceptual experiment similar to the one described in section 2.4 would be better suited to verify the usefulness of manipulating the F0 weight in the concatenation cost function.

3.2. Amplitude Modeling

Even for very carefully recorded speech databases, different realizations of one phoneme will have different sound levels, since they were produced in different contexts. Since loudness is no explicit selection criterion and only is taken into account when calculating the concatenation costs, it is possible that a unit is selected which fits perfectly except for the volume. To remedy this problem, we tried to apply, as the final step in synthesis, amplitude modification based on models we built before. The models were created by inspecting every occurrence of each phoneme in our database, measuring the RMSE values at 10, 25, 50, 75 and 90% of the phoneme duration and calculating the means [10]. In applying these models to the synthesized signal in the final step, each sample is multiplied by the factor determined by these models. Values between the calculation points are linearly interpolated. The procedure was based on the one used in the Bell Labs speech synthesis system [12, p. 222].

Pauses and plosives are not modified; the former since they have no energy, and the latter since they are hard to normalize due to their different phases (pause, burst and friction).

Figure 2 shows an example comparing the amplitude profile of the unmodified signal (blue dashed line) and the profile of the amplitude normalized signal (red solid line) for the phrase *einer der zentralen Plätze (“one of the central squares”).* The speech signal looks more natural after the modification. For instance, the [a:] is louder than the schwa [@] after the modification, which seems more natural than the other way round, which it was before the modification.

However, a perception experiment with 35 subjects using the same experimental procedure as described in section 2.4 showed that the unmodified signal was very clearly preferred over the amplitude normalized signal. The original signal was rated better for 52.2% of the stimulus pairs, while the normalized variant was preferred for only 12.0% of the pairs, and both variants were rated equally good for the remaining 35.8% of the pairs. The differences were statistically significant (χ^2(2,N=1040)=261.56,p$\ll$0.05). The accordance in listeners' judgements was overwhelming. Not a single listener consistently preferred the normalized variant. On the contrary, every listener rated the original variant better more often than the normalized variant, and this preference was significant for 15 out of 35 listeners (15 out of 35 listener-specific χ^2 tests yield values of p $<$ 0.05/35 $\simeq$ 0.001, and only 3 tests yield p $>$ 0.05). Also, for no pair of stimuli, the normalized variant was rated better by more listeners than the original variant. The preference again was significant for most stimulus pairs (15 out of 23 stimuli-specific χ^2 tests yield values of p $<$ 0.05/23 $\simeq$ 0.002).

Given the negative outcome of the perception experiment, we analyzed the stimuli once again. On first visual inspection, the amplitude normalized variants seemed to be superior to the original variants. The amplitude profiles looked smoother and more natural, and usually, the normalization did not “stick out” perceptually. However, in few cases, the normalization caused problems for some segments. This occurred mostly for segments which exhibited lower amplitudes than expected. In these cases, the normalization resulted in boosting the respective segment too much, revealing phenomena that would otherwise not have been heard so clearly. In one example, a very low [l] segment contained an almost inaudible burst caused by the articulatory movement from a preceding [S]. After normalizing the [l] segment to the average amplitude of /l/ phonemes, the burst is perceived as an irritating noise. In another example, a problematic concatenation in a very low [@] segment caused a discontinuity that became much more obvious after the normalization. Some phonemes were generally problematic. For instance, syllable-initial vowels are often glottalized to some degree in German, there may even be the release of a glottal stop at the beginning of the vowel. These glottalized realizations have lower amplitudes than the non-glottalized modal-voiced realizations, and in these cases, raising the amplitude results in unnaturally loud glottal stops or glottalized vowel phases. Also, initial /h/ was generally boosted too much, making it sound almost like /x/. Thus, although the normalization for most segments did not compromise the quality of the speech signal, there was often at least one of the few problematic segments in the test sentences, and listeners seldom failed to detect them. A possible solution to this problem may be to limit the degree to which very low segments are manipulated, e.g. by assuming an upper limit for the normalization factor, but this will have to be investigated in the future.

4. Language disambiguation for proper names

In the course of the project, we added a simple language disambiguation component for proper names. Apart from the fact that proper names pose problems because of their often irregular pronunciation, particularly first names are often ambiguous between several languages. For instance, the first name *David* is pronounced differently depending on whether it is a German, English, French or Spanish name. However, the context often helps to disambiguate, e.g. in the above example, if the name *Beckham* follows, the English variant is obviously correct.

We have added a mechanism that facilitates disambiguation in such cases. This mechanism presupposes that there is a lexicon that not only contains transcriptions of the proper names

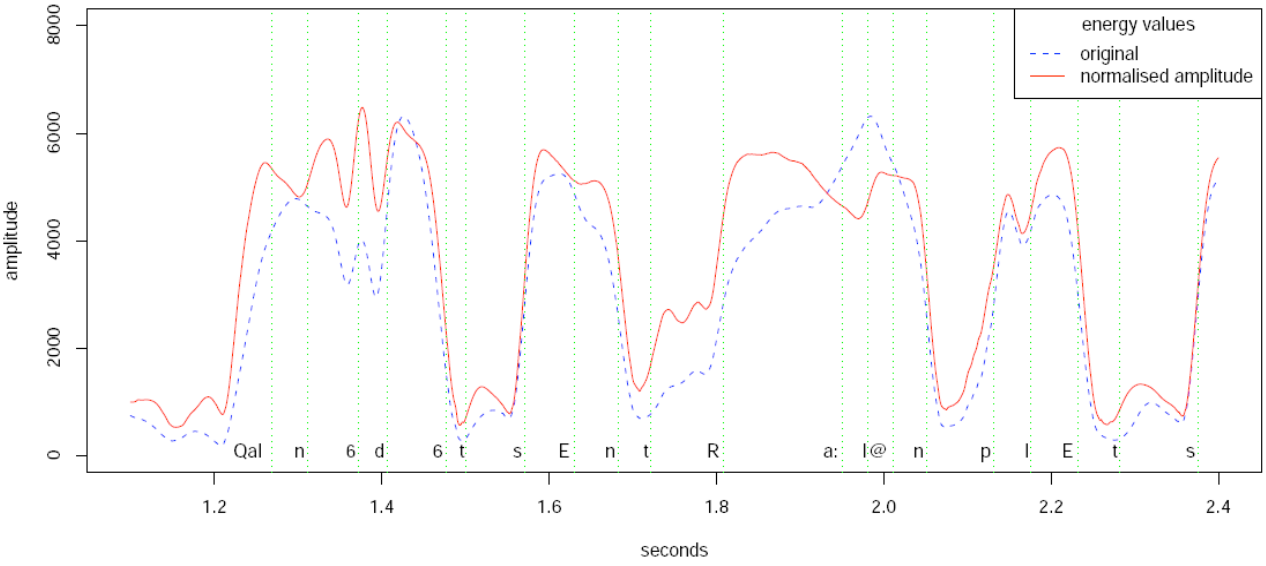


Figure 2: *Amplitude normalization. Note that the* [*a:*] *(1.8s) is louder than the* [*@*] *after the modification.*

but also that their origin is coded into the part of speech tag. In the example cited above, the lexicon would have to contain several entries for *David*, marked as a proper name of German, English, French, or Spanish origin, respectively, and one entry for *Beckham*, marked as English. This information could result from extracting proper names from foreign lexicons. In our experience, it is sufficient to differentiate between several languages or groups of languages. For instance, given the degree to which we adapted foreign pronunciations to our phoneme inventory, it was adequate to differentiate between English, German, and French, but the Spanish languages and Portuguese could be grouped together, and there was no further distinction necessary between different Slavic languages, or between different Asian languages.

The mechanism automatically collects all different transcriptions of orthographically identical proper names including their tags into a table that lists all possible origins for all ambiguous proper names. During synthesis, upon encountering an ambiguous name, the pronunciation is left underspecified by assigning the set of all possible origins to each name. Then, they are disambiguated by unifying the sets of possible origins of consecutive proper names.

Although we have only a moderate number of proper names that are marked for their origin (approximately 2,000 names), the mechanism has greatly improved the subjective synthesis quality because some of the most frequent cases of ambiguous names occurred very frequently in a SmartWeb key application, viz. the access of information on the soccer World Championships 2006.

5. Conclusions

We have described three aspects of the unit selection synthesis used in the SmartWeb system. First, we have described and compared two unit selection strategies. With respect to the PSM/MC strategy, the optimal feature rankings and the most adequate basic unit type should be investigated further. Particularly the optimal feature ranking will give interesting insights in the perceptual relevance of the respective features from a theoretical perspective. Another open issue is the treatment of bursts in concatenation, which should be addressed in the future. With these improvements, we expect the PSM/MC approach to surpass the PSM/AC approach in the future. For the time being, the PSM/AC approach is preferred over the PSM/MC approach in the SmartWeb project.

Second, we have discussed our experiences with different weights to enforce pitch continuity and with amplitude modeling. In the case of pitch continuity, we introduced an additional F0 weight factor that successfully brings the F0 weights in an order of magnitude comparable to the weights applied to spectral differences. However, a perceptual experiment to confirm the usefulness of manipulating the F0 weights has yet to be conducted. With regard to amplitude modeling, we found that it is clearly not useful, at least not in the way it has been applied here. Assuming an upper limit for the normalization factor may be expedient, but this has not been verified yet.

Finally, we have sketched a simple mechanism for language disambiguation of proper names that improved the subjective synthesis quality particularly for a SmartWeb key application.

6. Acknowledgments

This work was funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the SmartWeb project under Grant 01IMD01H. The responsibility for the content lies with the authors.

The perception experiment and the amplitude modeling were implemented by Andreas Madsack in the context of his diploma thesis. He also experimented with the pitch weights. The perception tests concerning the two selection strategies were carried out by Christian Scheible and Sylvia Riester.

7. References

[1] W. Wahlster, “Smartweb: Mobile applications of the semantic web,” in *KI 2004: Advances in Artificial Intel-*

ligence, S. Biundo, T. Frühwirth, and G. Palm, Eds. Berlin/Heidelberg: Springer, 2004, pp. 50 – 51.

[2] A. Schweitzer, N. Braunschweiler, G. Dogil, T. Klankert, B. Möbius, G. Möhler, E. Morais, B. Säuberlich, and M. Thomae, "Multimodal speech synthesis," in *SmartKom: Foundations of Multimodal Dialogue Systems*, W. Wahlster, Ed. Springer, 2004, pp. 411–435.

[3] "IMS German Festival home page," [http://www.ims.uni-stuttgart.de/phonetik/synthesis/], 2007.

[4] P. Taylor and A. W. Black, "Speech synthesis by phonological structure matching," in *Proceedings of the 6th European Conference on Speech Communication and Technology (Budapest, Hungary)*, vol. 2, 1999, pp. 623–626.

[5] A. W. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in *Proceedings of the 5th European Conference on Speech Communication and Technology (Rhodos, Greece)*, vol. 2, 1997, pp. 601–604.

[6] K. Claßen, G. Dogil, M. Jessen, K. Marasek, and W. Wokurek, "Stimmqualität und Wortbetonung im Deutschen," *Linguistische Berichte*, vol. 174, pp. 202–245, 1998.

[7] M. Jessen, K. Marasek, K. Schneider, and K. Claßen, "Acoustic correlates of word stress and the tense/lax opposition in the vowel system of German," in *Proceedings of the 13th International Congress of Phonetic Sciences (Stockholm, Sweden)*, vol. 4, 1995, pp. 428–431.

[8] B. Möbius, "Rare events and closed domains: Two delicate concepts in speech synthesis," *International Journal of Speech Technology*, vol. 6, no. 1, pp. 57–71, 2003.

[9] "The Festival Speech Synthesis System," [http://www.cstr.ed.ac.uk/projects/festival/], 2007.

[10] A. Madsack, "Amplitude normalisation and intonation continuity modelling for unit selection," Diplomarbeit, IMS, Universität Stuttgart, Stuttgart, 2006.

[11] G. Möhler and A. Conkie, "Parametric modeling of intonation using vector quantization," in *Proceedings of the Third International Workshop on Speech Synthesis (Jenolan Caves, Australia)*, 1998, pp. 311–316.

[12] R. Sproat, Ed., *Multilingual Text-to-Speech Synthesis: The Bell Labs Approach*. Dordrecht: Kluwer, 1998.

Building a Finnish Unit Selection TTS system

Hanna Silen, Elina Helander,
Konsta Koppinen, Moncef Gabbouj

Institute of Signal Processing
Tampere University of Technology, Finland
{hanna.silen, elina.helander, konsta.koppinen, moncef.gabbouj}@tut.fi

Abstract

Speech synthesis based on unit selection can produce far more natural speech than conventional diphone-based methods. Unit selection based text-to-speech synthesizers have been built for many different languages. In this paper, we describe the development of TUT_VOICE, the first Finnish unit selection synthesis engine for academic research. The system includes database construction, synthesis engine implementation and optimization for Finnish.

1. Introduction

Unit selection [1] is a method of corpus-based concatenative speech synthesis. It uses a large pre-recorded speech inventory to provide a sufficient phonetic and prosodic coverage for a language. Speech is produced by cutting and concatenating units from the database.

One of the major challenges is how to select units from the database. The selection process is guided by two costs, *target* and *join cost* [1]. The target cost estimates similarity between a candidate unit and a desired unit and join cost measures the concatenation quality of two consecutive units in terms of the continuity of the spectrum, F_0, and energy.

When appropriate units are chosen in synthesis, contexts are taken into account. However, the quality of synthesized speech depends highly on the coverage of the database. One basic idea of unit selection is to avoid signal processing modifications i.e. prosodic modifications. This poses a challenge to the inventory; it must provide not only a complete coverage of synthesis units but also many instances of a same unit in different contexts. The design of the database is thus important and should be tailored to language-specific requirements. In addition, the style of the synthesized speech follows the style of the database.

There has been a vast amount of research on unit selection TTS (text-to-speech) and voices have been developed for many languages. Although there exists a fair amount of freely available research and speech analysis tools (e.g. Festival [2]), for a new language a proper database is still needed as well as rules for grapheme-to-phoneme conversion, linguistic parsing, etc. Previously, a Finnish diphone voice (*hy_fi_mv_diphone* [3]) has been built in Festival [2]. However, no prior academic research has been devoted to building a unit selection voice for Finnish. In addition, no suitable database has been available.

This paper describes the process of building a Finnish prototype open-domain unit selection system TUT_VOICE. The building process of TUT_VOICE consisted of two phases. The first part, inventory construction, involved prompt selection, recording of the inventory, utterance labeling and feature extraction. In the second part, a unit selection synthesis engine was implemented consisting of target construction, unit sequence selection and waveform concatenation. For TUT_VOICE, some ideas from Festival were adopted but the system was built to work independently from it.

This paper is organized as follows. Finnish phonetics and phonology are explained briefly in Section 2. Section 3 describes the prompt design, recordings and labeling of the database. In Section 4, the synthesis engine implementation and cost adaptation for Finnish are explained. Section 5 reports the results of the listening experiments. Some important findings and future work are given in Section 6. Section 7 concludes the paper.

2. Finnish phonetics and phonology

Although a unit selection synthesizer can be built on having little or no knowledge of the language (e.g. [4]), understanding the characteristics of the language is important for good quality TTS. Some issues presented in this chapter helped us to understand some errors occurring in the synthesized speech. This chapter outlines the basic principles of Finnish phonetics and phonology from the viewpoint of what is needed for building a speech synthesizer.

2.1. Phoneme system and orthography

Phonemes are typically divided into consonants and vowels. There are eight vowels in Finnish: /ɑ/, /e/, /i/, /o/, /u/, /y/, /a/, and /ø/. Vowels can occur both short and long and form sequences and diphthongs. Compared to other languages, the number of vowels in Finnish is high [5]. On the other hand, a relatively low number of consonants exists. The low number of consonants enables the appearance of a high number of allophones [5]. The consonants and allophones are summarized in Table 1. Consonants are marked using the notation of the Finno-Ugric transcription instead of the International Phonetic Alphabet. Most of the consonants in Table 1 can form geminates. In addition, the consonants /b/, /g/, /f/, and /ʃ/ occur only in relatively new loanwords.

Finnish orthography is phonemic: each phoneme corresponds to a certain grapheme and allophones are not pointed out. Short phoneme quantities are written with a single grapheme (e.g. *i*) whereas long phoneme quantities (e.g. *ii*) and diphthongs (e.g. *au*) with two graphemes. There is only one exception: the orthographic correspondent for the phoneme /ŋ/ is *ng*. The main differences between the Finnish orthography and pronunciation are due to assimilation and boundary gemination [5].

Table 1: *Finnish consonants and their allophones.*

	phoneme	allophones	examples
plosive	p	[p]	*pallo*
	t	[t] [t̹]	*tutti*, *tutit*
	k	[k] [k͕]	*katu*, *kirje*
	d	[d]	*lyhde*
nasal	m	[m] [m̹]	*maila*, *kamferi*
	n	[n] [m]	*onni*, *päähänpisto*
		[m̹] [n͕]	*fanfaari*, *tunti*
		[ŋ]	*kenkä*
	ŋ	[ŋ] [ŋ͔]	*kengät*, *kangas*
fricative	s	[s]	*sana*
	h	[h] [ɦ]	*tahto*, *raha*
lateral	l	[l] [ɫ]	*lika*, *laki*
trill	r	[r] [ð]	*penger*, *taru*
		[ř]	*tutkimusretki*
approximant	ʋ	[ʋ]	*vanha*
	j	[j]	*juhla*

2.2. Syllables and syllabification

Every word can be divided into one or more syllables. Each syllable in Finnish has a vowel as a sonant, i.e. every syllable must contain at least one vowel. The syllable structure list is given in Table 2. Letters C and V denote a consonant and a vowel, respectively. The notation VV denotes a long vowel or a diphthong. The structure of the most common Finnish syllables is simple and no complex consonant clusters exist as Table 2 shows. The majority of the words are polysyllabic. Only 17% of the words in the sentence set used in database construction (Section 3) were monosyllabic. For comparison, the respective number for English calculated from CMU ARCTIC database [6] was 72%.

Table 2: *Finnish syllable types.*

common	CV	CVC	CVV	CVVC	VC
	V	VV	CVCC	VVC	VCC
rare	CCV	CCVC	CCVV	CCVVC	CCVCC

The syllabification rules for Finnish are simple and no dictionaries are needed. Only foreign words and compound words can cause some exceptions. According to [5], Finnish syllabification can be carried out using the following rules:

- A syllable boundary appears before every sequence CV (e.g. *ka-tu*)
- A syllable boundary appears inside every sequence VV unless the sequence is a diphthong or long vowel (e.g. *a-lu-e*)
- A vowel sequence VV ending with /i/ is a diphthong if it is not in the first syllable (e.g. *u-te-li-ai-suus*)
- A vowel sequence VV ending with /u/ or /y/ and not located in the first syllable can be realized as a diphthong or a vowel sequence (e.g. *päi-vä-ys*, *päi-väys*).

2.3. Prosody

Prosody is not expressed through simple phonetic segments but larger units like syllables, words, sentences or even paragraphs. Prosodic features, such as quantity, stress and intonation play an important role in conveying information. It is generally believed that the naturalness of synthesized speech is improved through better prosody modeling. Although our unit selection synthesizer does not explicitly model prosody, there is a need to extract linguistic features that are assumed to affect the synthesized prosody. Thus understanding of Finnish prosody can help optimize the database and come up with meaningful target costs for synthesis.

One important manifestation of prosody is quantity. It can be determined physically or linguistically [7]. Physical quantity corresponds to a duration of a phoneme while linguistic quantity describes how a native speaker perceives the length. In Finnish there are two distinctive quantities: short and long for both vowels and consonants (geminates). The ratio of short and long vowel durations often differs from 1 : 2 [7].

Finnish word stress is fixed. The primary stress is always on the first syllable while the second and the last syllable are unstressed. In longer words, secondary stress can occur as well.

Voice quality can also be considered as a dimension of prosody. In Finnish, the use of a creaky voice at least at the end of a sentence is a frequent phenomenon although it can also appear elsewhere in the sentence [8]. Diphone-based synthesizers avoid the problem of creaky endings, since diphones are extracted from a stable speech section and they are modified to be suitable for every part of a sentence. However, a unit selection synthesizer faces the problem since all the speech material is used for synthesis and for example TUT_VOICE does not carry out prosodic modifications.

3. Database and voice construction

The lack of appropriate speech databases is a major problem for smaller languages like Finnish. An important output of this project is a speech database consisting of 1 003 utterances optimized for TTS synthesis and spoken by a female speaker. The sentences are narrative and read in a vivid style, since we aimed at expressive prosody. Hence, the database should also be useful for prosody research purposes.

3.1. Prompt design

The speech inventory was designed for Finnish unit selection synthesis using diphone-sized units. The design followed the idea of the English CMU ARCTIC databases [6]. In total 33 Finnish out-of-copyright books with 203 339 sentences were extracted from Project Gutenberg [9]. Altogether 46 067 sentences with 6-15 words were used as source data for the greedy prompt sentence selection.

Short and long phoneme quantities were treated as different phonemes in the selection. Due to the high allophonic variation of the Finnish consonants, a diphone variant-based approach was taken. By the concept of a diphone variant we distinguish diphones with similar phone content but variation in allophone content or syllabic position. For example, the intra-syllabic diphone *a_n* in the word *vanha* ([ʋɑn-hɑ], *old* in English) is considered different from the intra-syllabic variant in the word *vanki* ([ʋɑŋ-ki], *prisoner*) as well as the inter-syllabic variant in the word *vana* ([ʋɑ-nɑ], *trail*). If the variants are ignored in the prompt selection, there is no guarantee, that all the variants are included in the inventory.

Two separate sets of prompt sentences were selected. The first set (Set A) was optimized to provide full coverage of diphone variants occurring in the source data. New sentences were included as long as there were diphone variants missing

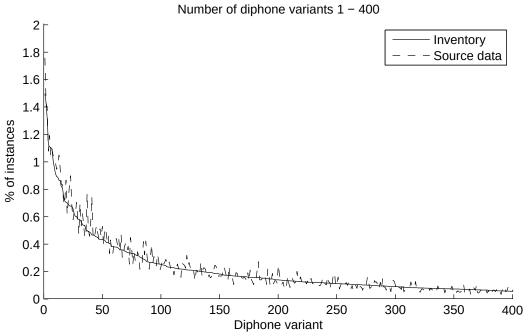


Figure 1: *Distribution of the most frequent diphone variants in the inventory.*

from the inventory. If several sentences with an equal number of new diphone variants were encountered, selection was based on the frequency of the sentences' diphone variants in the source data. Since boundary gemination makes it difficult to predict the actual pronunciation on the word boundaries, inter-word diphone variants were ignored in optimization. Due to vowel harmony in Finnish, front and back vowels do not appear in the same word and inter-word diphone variants consisting of vowels were taken into account.

The second set (Set B) was designed to be rich in syllables. Allophonic variation and stress were taken into account in the selection. Stress in Finnish is fixed and word-initial syllables were considered stressed and the word-final unstressed. Sentences were greedily selected by choosing always the one providing the largest amount of new syllables. The last word of each sentence was ignored due to the possible occurrence of creaky endings. The first word of each sentence as well as the monosyllabic words were ignored. Sentences of the Set A were taken into account in the selection. After manual removal of archaic and foreign sentences, a set of 1 003 prompt sentences was left. Sentences were recorded with a female voice using a sampling frequency of 32 kHz.

The distribution of the most frequent diphone variants in the inventory is illustrated in Figure 1. The solid line denotes the percentage of diphone variant instances of all the instances in the inventory while the dashed line denotes the corresponding value for the source data. As can be seen, the inventory distribution follows well the source data. The 440 most frequent diphone variants cover 90% of the inventory.

3.2. Automatic labeling

Automatic labeling of the inventory utterances used scripts of the Multisyn build tool [10] with slight modifications. HMM-based (hidden Markov model) phoneme models were trained with HTK (Hidden Markov Model toolkit) [11] and forced alignment was used for the phone boundary determination. Boundary alignment was done by using 5-state monophone HMMs. Plosives were divided into closure and explosion phases and separate 4-state HMMs were trained for them. Diphthongs turned out to be very difficult for the alignment and were therefore trained as separate models instead of separating the phones. Diphone boundaries were computed as the midpoint between the phone boundaries except for the plosives which had an aligned boundary between the closure and explosion.

Inventory utterances were spoken relatively fast which complicated automatic labeling. Some of the phones were very short, such as /l/, /j/, and vowels belonging to diphthongs. They get fused together and even manual labeling of these phones turned out to be difficult. In synthesis, /l/ and /j/ should be extracted as triphones rather than diphones. However, this would require including more data in the inventory in order to guarantee full coverage.

4. Synthesis engine implementation

The TUT_VOICE synthesis engine was implemented as a prototype unit selection TTS system for academic use. The implementation was inspired by the Festival TTS framework [2]. Adding new voices is easy and requires no system compilation. Adjusting the synthesis parameters such as the target and join subcost weights as well as changing the grapheme-to-phoneme rule sets can be done without compilation. The core system is implemented for Linux in C++ and the voice construction scripts in Perl. Examples of synthesized speech are available at [12].

4.1. Target construction

Grapheme-to-phoneme mapping for Finnish is quite straightforward and syllabification is done based on some simple rules described in Section 2. The structure of the input sentence is determined by parsing the sentence into a tree-like form similarly to Festival.

4.2. Unit sequence search

Selection of the candidate unit sequence is carried out by computing the total cost $C(\mathbf{t},\mathbf{u})$ between the target unit sequence $\mathbf{t}$ and a candidate unit sequence $\mathbf{u}$ [1] as

$$C(\mathbf{t},\mathbf{u}) = \sum_{i=1}^{N} C^t(t_i,u_i) + \sum_{i=2}^{N} C^j(u_{i-1},u_i). \qquad (1)$$

Here $C^t(t_i,u_i)$ denotes the target cost between a target unit t_i and a candidate unit u_i and $C^j(u_{i-1},u_i)$ the join cost between candidate units u_{i-1} and u_i. The best candidate unit sequence $\mathbf{u}^*$ is the one that minimizes the total cost, i.e.

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} C(\mathbf{t},\mathbf{u}). \qquad (2)$$

Optimization is done by using the Viterbi search algorithm [1].

4.3. Target cost

The target cost is used to estimate the dissimilarity between a target unit and a candidate unit from the inventory. It is formed as a weighted sum of subcosts. Subcosts are selected in a way that they can characterize the phonetic and prosodic properties of the units.

The formula for the target cost $C^t(t_i,u_i)$ of a target unit t_i and a candidate unit u_i is [1]

$$C^t(t_i,u_i) = \sum_{n=1}^{q} w_n^t C_n^t(t_i,u_i), \qquad (3)$$

where q denotes the number of subcosts C_n^t and w_n^t a weight given to each subcost.

The used subcosts were the position in syllable, word, and sentence; stress; and left and right phoneme context. All the subcost weights were manually adjusted.

Table 3: *Synthesis parameters for the target subcosts.*

target subcost	feature values	weight
position in		
syllable	{initial, medial, final, inter}	0.1
word	{initial, medial, final, inter}	0.1
sentence	{initial, medial, final}	0.3
stress	{primary, secondary, unstressed}	0.35
phone context		
left	{*a*, *a:*, *b*, *b:* ..., *oe:*}	0.1
right	{*a*, *a:*, *b*, *b:* ..., *oe:*}	0.05

As in Festival, the highest weight was given to stress. Three different cases were distinguished: syllables with primary and secondary stress and syllables with no stress. Since the primary stress is always on the first syllable in Finnish, word stress is related to word boundary detection [7]. Stress was therefore considered important in terms of intelligibility as well.

The second highest weight was given to the unit's position in a sentence. High weight was used in order to avoid the selection of the candidate units with a creaky voice for the target units not from sentence-final words. A unit's position in a syllable and word were not considered as important and were less highly weighted. Units were considered either sentence/word/syllable initial, medial, final, or interword/intersyllable.

Allophonic variation of the phonemes was not taken into account in the transcription. Instead, coarticulation was estimated by the left and right context subcosts.

4.4. Join cost

The join cost is used to estimate the audible mismatches occurring in unit concatenation. Similarly to the target cost, the join cost is formed as a weighted sum of subcosts. Differences in spectral features, F$_0$, and power are typically considered in join cost computation [1]. Formula for the join cost $C^j(u_{i-1}, u_i)$ for candidate units u_{i-1} and u_i is [1]

$$C^j(u_{i-1}, u_i) = \sum_{n=1}^{p} w_n^j C_n^j(u_{i-1}, u_i), \tag{4}$$

p denoting the number of join subcosts C_n^j and w_n^j the weight given to each subcost.

A continuous pitch contour on the unit boundaries was achieved by using the distance of the units' F$_0$ as a join subcost. However, since no F$_0$ is extracted for the unvoiced segments, the F$_0$ join subcost between two arbitrarily selected unvoiced segments equals zero. Therefore the use of F$_0$ subcost can not guarantee good overall pitch contour. To overcome the problem, we linearly interpolated the values for the unvoiced parts based on the F$_0$ values of the surrounding voiced parts. Values were normalized to have mean value of 0 and variation of 1.

Spectral mismatches were estimated by the weighted mean-square error (WMSE) of LSFs (Line spectral frequency coefficients). In comparison to MFCCs (Mel-frequency cepstral coefficients), LSFs have turned out to estimate better the occurring audible mismatches [13]. The WMSE of two LSF frames $\mathbf{f}_1$ and $\mathbf{f}_2$ is computed as [14]

$$d(\mathbf{f}_1, \mathbf{f}_2) = \sum_{n=1}^{p} w_n (f_1(n) - f_2(n))^2, \tag{5}$$

where w_n denotes the weight and $f_1(n)$ and $f_2(n)$ the nth coefficients of the frames $\mathbf{f}_1$ and $\mathbf{f}_2$, respectively. The weight w_n is given as

$$w_n = \max_{i=1,2} \frac{1}{f_i(n) - f_i(n-1)} + \frac{1}{f_i(n) - f_i(n+1)}. \tag{6}$$

The waveform amplitude was controlled by the power join cost. The subcost value was computed as the absolute difference of the power of one pitch period at the concatenation point. Extracted power values were normalized into range of $[0, 1]$.

Difficulties in labeling of some short and poorly detectable phonemes were compensated by introducing a triphone join subcost. Especially the phonemes /l/ and /j/ were found to be very difficult to label correctly, even manually. The aim of the triphone cost was to guide the selection in these cases towards triphone-sized candidate units rather than splitting the short phonemes in order to form diphones. The triphone cost was defined to get a value of 1 if the diphone should be selected as a triphone and 0 otherwise. By using the weighted triphone subcost rather than forced triphone selection, a wider variety of candidate units was achieved.

The weights for each join subcost are listed in Table 4. Differences in weighting indicate the different range of subcost values rather than importance of a certain feature.

Table 4: *Weights for the TUT_VOICE join subcosts.*

target subcost	weight
F$_0$	0.05
LSF	1
power	3
triphone	1

4.5. Pre-selection

In order to speed up the unit sequence search, a pre-selection was used. Units with numerous instances in the inventory were divided into groups of inter- and intra-syllabic instances. Units from the other group were not considered as candidate units and were therefore left out from the search. For the diphones with less than 5 instances, no pre-selection was carried out. The effect of pre-selection was tested by synthesizing a set of utterances with and without the pre-selection. Among 300 synthesized sentences, 222 were the same regardless of whether the pre-selection was used or not.

4.6. Waveform concatenation

Unit waveforms were extracted from the inventory utterances pitch-synchronously. Diphone boundaries were aligned at the midpoints between the phone boundaries determined by HTK. As an exception, the plosives were divided at the end of the closure determined by HTK. In order to achieve pitch-synchronous waveform extraction, the boundary was moved on the nearest pitch mark.

Glitches on the unit boundaries were avoided by allowing some overlapping. The best concatenation point was determined by finding the width of overlap that provided the highest value of cross correlation. A smooth transition was obtained by averaging the signals in the overlapping region. Roughly one pitch period of overlap was allowed.

Figure 2 illustrates the utterance *"Sehän on jo valmis rautatie, penger tehty, ojat kaivettu, kiskot pantu paikoilleen."*

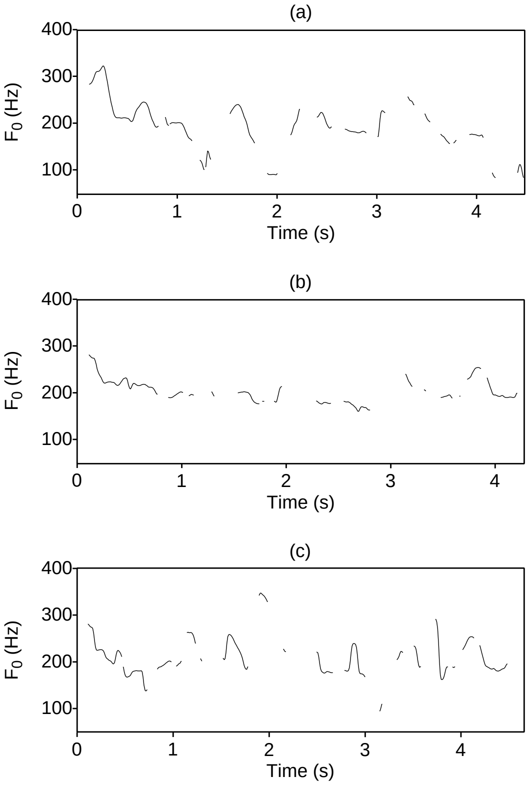


Figure 2: *F_0 contours for (a) recorded utterance, synthesized utterance (b) with F_0 interpolation, and (c) without F_0 interpolation.*

as (a) recorded, (b) synthesized with F_0 interpolation, and (c) without F_0 interpolation. Note the undesirable jumps in the F_0 in the utterance with no interpolation used in F_0 extraction.

5. Listening experiments

Since the output of a TTS system is speech, the evaluation of a TTS system is usually carried out by conducting listening tests. With speech coders, a MOS test is commonly used and it has also been applied to TTS system assessment. Our questionnaire (in Table 6) included three question concerning intelligibility, general quality and naturalness.

A total of 1 000 sentences were generated using sentences similar to those in [15] as well as narrative sentences. From these, 14 sentences were randomly chosen. One major problem in unit selection speech synthesizers is to make them produce robust quality. An all-inclusive or even a comprehensive evaluation through listening experiments is extremely difficult or impossible. The random selection was inevitable due to the varying quality of the sentences. Thus, we could have chosen a set of 14 sentences that would get the highest scores as well as a set of 14 sentences that would obtain low scores at least for quality and naturalness.

These 14 sentences were rated by 8 native Finnish listeners and the averaged results and the corresponding values of standard deviation are shown in Table 5. The ratings from different MOS tests can not be compared with each other, but an interested reader is referred to [15] where some commercial Finnish TTS systems were evaluated.

Table 5: *Listening test ratings.*

	average	worst sentence	best sentence
intelligibility	3.61 (1.00)	2.25 (0.89)	4.63 (0.52)
naturalness	2.99 (0.89)	2.38 (0.92)	4.25 (0.46)
quality	3.20 (0.70)	2.63 (0.52)	3.88 (0.64)

Table 6: *Evaluation questionnaire.*

INTELLIGIBILITY: Did you understand everything without an effort, how would you describe the pronunciation?	
5	Excellent (no efforts, very clear pronunciation)
4	Good (small mistakes on pronunciation but did not bother)
3	Fair (a little annoying mistakes appeared)
2	Poor (annoying errors)
1	Bad (I did not understand the content because of too strong errors)
QUALITY: How would you describe the speech quality?	
5	Excellent (nothing bothered)
4	Good
3	Fair
2	Poor
1	Bad (I could not listen to speech of this quality a moment longer)
PROSODY AND NATURALNESS: Did the sample sound natural?	
5	Very natural
4	Natural
3	Somewhat natural
2	Unnatural
1	Highly unnatural

6. Findings and future work

The database consisted only of approximately 1.5 hours of speech. CMU ARCTIC databases are of similar size but the results of Blizzard Challenge implied that the original databases were too small [10]. It is a small database compared to the many commercial systems that use around tens of hours of speech. Due to the small database and its expressive style, naturalness and concatenation smoothness turned out to be somewhat contradictory requirements. The synthesis was found to sound rich in prosody but sometimes at the cost of concatenation smoothness.

The current TUT_VOICE system was implemented as a prototype and no extensive tuning of the system was done. The weights for the costs were tuned by hand but automatic phone-specific subcost training will be carried out in the future. Some very bad labeling mistakes were corrected manually and extra logic was included in the synthesizer to reduce label mismatches but finally the whole database should be manually corrected. In its current form, TUT_VOICE is not yet suitable for real-time speech synthesis but in the future, it will be modified to work real-time.

Creaky endings that are common in Finnish require some extra handling. The database design process took sentence-final syllables into account and did not accept them for coverage optimization. In the recordings, although special attention was paid on using a creaky voice at the end of a sentence, they still appeared. In synthesis, creaky endings are currently handled through target costs, i.e. by penalizing the use of sentence final diphones elsewhere. On the other hand, the use of creaky voice quality at the end of a synthesized sentence can improve naturalness.

Sentences for the listening test were picked randomly and the results illustrated the general problem of unit selection: quality variability. The prosody of the synthesized speech was not rated as high as we expected. We found that it was mainly because of strange phoneme durations. In Finnish, phoneme duration plays a relatively important role. On the contrary, intonation is generally rather monotonic compared to many other languages. F_0 in the synthesized speech was found to be quite successful.

7. Conclusions

This paper described the design and implementation of a Finnish unit selection TTS system called TUT_VOICE. The quality of current commercial English unit selection speech synthesizers is high and the focus has moved into flexibility, for example generating new styles and emotions. The quality of TUT_VOICE is not yet at the same level, and one reason for that is a rather small database (1.5 hours). However, TUT_VOICE is a step towards natural, and style-variable flexible high-quality speech synthesis in Finnish.

8. Acknowledgments

This work was supported by the Academy of Finland, project No. 5213462 (Finnish Centre of Excellence program 2006 - 2011).

9. References

[1] Hunt, A. and Black, A., "Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database", Proc. of ICASSP, pp. 373–376, 1996.

[2] Black, A., Taylor, P., and Caley, R., "The Festival Speech Synthesis System: system documentation". The Centre for Speech Technology Research, The University of Edinburgh, Scotland, UK, 2002. Edition 1.4, for Festival version 1.4.3. Available at www.festvox.org/docs/manual-1.4.3/festival_toc.html. Referred 29.06.2007.

[3] Vainio, M., Werner, S., Volk, N., Välikangas, J., and Järvikivi, J., "Finnish Speech Technology: A Multidisciplinary Project", 2006, Unofficial web page. Available at www.ling.helsinki.fi/suopuhe. Referred 29.06.2007.

[4] Black, A. and Llitjos, A., "Unit selection without a phoneme set", Proc. of IEEE Workshop on Speech Synthesis, pp. 207–210, 2002.

[5] Lieko, A. "Suomen kielen fonetiikkaa ja fonologiaa ulkomaalaisille", Oy Finn Lectura Ab, Loimaa, 1992, 197 p.

[6] CMU_ARCTIC speech synthesis databases. Available at http://festvox.org/cmu_arctic. Referred 29.06.2007.

[7] Wiik, K., "Fonetiikan perusteet", WSOY, Helsinki, 2nd edition 1998. 133 p.

[8] Iivonen, A., "Creaky voice as a prosodic feature in Finnish", Nordic Prosody, Proc. of the IX Conference, Peter Lang ed., pp. 137–146, 2004.

[9] Project Gutenberg. Available at www.gutenberg.org. Referred 29.06.2007.

[10] Clark, R., Richmond, K. and King, S., "Multisyn voices from ARCTIC data for the Blizzard challenge", Proc. of Interspeech, 2005.

[11] Young, S., Everman, G., Gales, M., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P., "The HTK book for HTK version 3.3", Cambridge University Engineering Department, 2005, 344p.

[12] TUT_VOICE synthesis examples. Available at www.cs.tut.fi/sgn/arg/ssw6/tut_voice.html. Referred 29.06.2007.

[13] Vepa, J. and King, S., "Subjective Evaluation of Join Cost and Smoothing Methods for Unit Selection Speech Synthesis", IEEE Transactions on audio, speech, and language processing, 14(5):1763–1771, 2006.

[14] Kondoz, A., "Digital Speech, Coding for Low Bit Rate Communication Systems", John Wiley & Sons, Ltd, West Sussex, England, 2nd edition, 2004.

[15] Ojala, T., "Auditory quality evaluation of present Finnish text-to-speech systems", Helsinki University of Technology, 2006, 65 p.

Evaluating Automatic Syllabification Algorithms for English

Yannick Marchand[1,2], *Connie R. Adsett*[1,2] *and Robert I. Damper*[1,3]

[1]Institute for Biodiagnostics (Atlantic), National Research Council Canada,
1796 Summer Street, Suite 3900,
Halifax, Nova Scotia, Canada B3H 3A7

[2]Faculty of Computer Science, Dalhousie University,
Halifax, Nova Scotia, Canada B3H 1W5

[3]School of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, UK

{yannick.marchand, connie.adsett}@nrc-cnrc.gc.ca, rid@ecs.soton.ac.uk

Abstract

Automatic syllabification of words is challenging, not least because the syllable is difficult to define precisely. This task is important for word modelling in the composition process of concatenative synthesis as well as in automatic speech recognition. There are two broad approaches to perform automatic syllabification: rule-based and data-driven. The rule-based method effectively embodies some theoretical position regarding the syllable, whereas the data-driven paradigm infers 'new' syllabifications from examples assumed to be correctly-syllabified already. This paper compares the performance of the two basic approaches. However, it is difficult to determine a correct syllabification in all cases and so to establish the quality of the 'gold standard' corpus used either to quantitatively evaluate the output of an automatic algorithm or as the example-set on which data-driven methods crucially depend. Thus, three lexical databases of pre-syllabified words were used. Two of these lexicons hold the same 18,016 words with their corresponding syllabifications coming from independent sources, whereas the third corresponds to the 13,594 words that share the same syllabifications according to these two sources. As well as one rule-based approach (Fisher's implementation of Kahn's syllabification theory), three data-driven techniques are evaluated: a look-up procedure, an exemplar-based generalization technique, and syllabification by analogy (SbA). The results on the three databases show consistent and robust patterns: the data-driven techniques outperform the rule-based system in word and juncture accuracies by a very significant margin and best results are obtained with SbA.

1. Introduction

The syllable has been much discussed as a linguistic unit. Whereas some linguists make it central to their theories (e.g., [1, 2]), others have ignored it or even argued against it as a useful theoretical construct (e.g., [3]). Much of the controversy centers around the difficulty of defining the syllable. Crystal [4], for instance, states that the syllable is "[a] unit of pronunciation typically larger than a single sound and smaller than a word" but goes on to write: "Providing a precise definition of the syllable is not an easy task" [p. 342]. There is general agreement that a syllable consists of a *nucleus* that is almost always a vowel, together with zero or more preceding consonants (the *onset*) and zero or more following consonants (the *coda*). However, determining exactly which consonants of a multisyllabic word belong to which syllable is problematic. Good general accounts of the controversy are provided by [5] and [6], with the former more specifically considering English—the language of interest in this paper—and the latter focusing on French.

However it is defined, and whatever the rights or wrongs of theorising about its linguistic status, syllable knowledge aids word modeling in automatic speech recognition and/or the unit selection and composition process of concatenative synthesis. For instance, Müller, Möbius and Prescher [7] write "syllable structure represents valuable information for pronunciation systems" [p. 225]. That is, the pronunciation of a phoneme can depend upon where it is in a syllable and therefore there are good practical reasons for seeking powerful algorithms to syllabify words.

Traditional approaches to automatic syllabification have been *rule-based* (or knowledge-based), implementing notions such as the maximal onset principle [1, 8] and sonority hierarchy [9], including ideas about what constitute phonotactically legal sequences in the coda, for instance. An alternative to the rule-based methodology is the *data-driven* (or corpus-based) approach, which attempts to infer 'new' syllabifications from an evidence base of already-syllabified words (i.e., a dictionary or lexicon[1]).

This paper compares the performance of these two basic approaches to automatic syllabification in the pronunciation domain. Our work attempts to be *predictive*, aimed at finding good syllabifications for practical application in speech technology and computational linguistics, rather than *descriptive*, aimed at explaining experimental data and/or giving insight into any linguistic theory of the syllable.

2. Electronic lexical databases

A key issue in assessing algorithms for automatic syllabification is the quality of the 'gold standard' corpus used to define the correct result. Further, in the data-driven paradigm, this corpus forms the evidence base for inferring new syllabifications;

[1] In this paper, we will use the terms *evidence base*, *lexical database*, *dictionary*, *corpus*, and *lexicon* interchangeably, except where we refer to a 'dictionary' by name (e.g., *Webster's Pocket Dictionary*).

hence, it is vital that its content is accurate. This, however, is extremely difficult due to the absence of any means of determining canonically correct syllabifications. Our approach is to use multiple dictionaries and to seek consensus among them, so as to reduce the possibility that our results are affected by the choice of a particular, idiosyncratic corpus.

In this work, we used two public-domain dictionaries—*Webster's Pocket Dictionary* and the *Wordsmyth English Dictionary-Thesaurus*—as the sources from which we derive three lexical databases, as described below.

2.1. Webster's Pocket Dictionary

The primary lexical database in this work is *Webster's Pocket Dictionary* (20,009 words), as used by [10] to train their NETtalk neural network. The database is publicly available for non-commercial use from `ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/dictionaries/` (last accessed 11 May 2007). For consistency with our previous work on pronunciation using this dictionary, homonyms (413 entries) were removed from the original NETtalk dataset leaving 19,596 entries. Sejnowski and Rosenberg have manually aligned the data, to impose a strict one-to-one correspondence between letters and phonemes[2]. The phoneme inventory is of size 51, including the null phoneme and 'new' phonemes (e.g., /K/ and /X/) invented to avoid the use of null letters when one letter corresponds to two phonemes, as in <x> → /ks/. The null phoneme (represented by the '–' symbol) was introduced to give a strict one-to-one alignment between letters and phonemes to satisfy the training requirements of NETtalk. In this paper, we retain the use of the original phonetic symbols (see [10], Appendix A, pp. 161–162) rather than transliterating to the symbols recommended by the International Phonetic Association. We do so to maintain consistency with this publicly-available lexicon.

In addition to the pronunciation, Sejnowski and Rosenberg have also indicated stress and syllabification patterns for each word. The form of the data is:

accumulate	`xk-YmYlet-`	0<>1>0>2<<
adaptation	`@d@pteS-xn`	2<2<>1>0<<

The second column is the pronunciation and the third column encodes the syllable boundaries for the words and their corresponding stress patterns:

<	denotes	syllable boundary (right)
>	"	syllable boundary (left)
1	"	primary stress
2	"	secondary stress
0	"	tertiary stress

Stress is associated with vowel letters and arrows with consonants. The arrows point towards the stress nuclei and change direction at syllable boundaries. To this extent, "syllable boundary (right/left)" is a misnomer because this information is not adequate by itself to place syllable boundaries directly. We can, however, infer four rules (or regular expressions) to identify syllable boundaries. Denoting boundaries by ' | ':

R1: [<>] ⇒ [< | >]
R2: [< digit] ⇒ [< | digit]
R3: [digit >] ⇒ [digit | >]
R4: [digit digit] ⇒ [digit | digit]

Word	*accumulate*	*adaptation*
Stress pattern	0<>1>0>2<<	2<2<>1>0<<
Syllabification	ac | cu | mu | late	ad | ap | ta | tion
Digit stress	00 | 11 | 00 | 2222	22 | 22 | 11 | 0000

Table 1: *Examples of stress and syllabification patterns.*

These have been confirmed as correct by Sejnowski (personal communication). Table 1 gives the syllable patterns of the three above examples.

2.2. Wordsmyth English Dictionary-Thesaurus

Disagreements may exist about the way a word should be segmented into syllables. A second (independent) lexical source was therefore used, namely the *Wordsmyth English Dictionary-Thesaurus*, so that our results would not be overly specialized to one particular dictionary. This source is also available via the World Wide Web from `www.wordsmyth.net` (last accessed 11 May 2007). This on-line lexical database originated in the early 1980's when Robert Parks, a Fulbright Fellowship researcher in Japan, began to develop an English dictionary for students to use on their computers. In 1991 and 1992, the dictionary was licensed to IBM to integrate into their products, and IBM in turn supported the development of the associated thesaurus. In 1996, the University of Chicago's ARTFL (American and French Research on the Treasury of the French Language) Project assisted in presenting the first World Wide Web edition. The dictionary is composed of about 50,000 headwords covering all areas of knowledge without technical vocabulary. It provides the syllables, pronunciation, part of speech, inflected forms, and definition for each word.

2.3. The three lexical databases

Homonyms were removed from the original *Webster's Pocket Dictionary* leaving 19,596 entries. Of these words, 18,016 were also found in the *Wordsmyth English Dictionary-Thesaurus*. These two independant dictionaries, each consisting of 18,016 syllabified entries, are referred to as *S&R* and *Wordsmyth*, respectively. A third database of syllabified words (hereafter *Intersection*) was derived consisting of the 13,594 words present in both public-domain dictionaries with identical syllabification patterns in these two independent lexical sources.

3. Syllabification algorithms

In this section, we briefly describe the four automatic syllabification techniques for which performance was compared.

3.1. Fisher's implementation of Kahn's procedure

In his PhD dissertation, Kahn proposed a theory of syllabification based on a different type of constraint [8]. Kahn postulated that syllabification in English is derived from three categories of consonant clusters: possible syllable-initial, possible syllable-final and 'universally-bad' syllable-initial (in his terminology). These consonant clusters are derived from the beginnings and endings of existing English words. For example, the two-phoneme sound /br/ is a possible syllable-initial consonant cluster because it forms the beginning of the word pronunciation /bred/ (<bread>) and it is therefore possible to syllabify the pronunciation /ənbreɪd/ (<unbraid>) as

[2] See [11] for extensive discussion of this alignment process and an algorithm for doing it automatically.

/ən | breɪd/. By contrast, /rk/ is considered a universally-bad syllable-initial consonant cluster because no English word begins with this sound combination. Therefore the pronunciation /mɑrkət/ (<market>) would be syllabified as /mɑr | kət/ and not /mɑ | rkət/.

A C implementation of Kahn's theory was developed in 1996 by William Fisher and can be downloaded from the file: `ftp://jaguar.ncsl.nist.gov/pub/tsylb2-1.1.tar.Z` (last accessed 11 May 2007). Because we were interested in the standard syllabification, we selected the two most appropriate of the five speech rates available in the program, the "slow, over-precise" (hereafter Basic) and the "ordinary conversational speech" (hereafter OCS) rates. The program also allowed the unsyllabified input to be provided with stress information (primary, secondary and no stress) on some specific phonemes[3] and without stress information. We processed the word list both ways, using the stress information provided in *S&R* (i.e., the digit stress—see Table 2.1). The phoneme set used in his program was translated to the phoneme set of *S&R* and all instances of the null phoneme were also removed because this special 'phoneme' was not part of Fisher's set.

3.2. Syllabification by analogy

Syllabification by analogy closely follows the principles of pronunciation by analogy (PbA) set out in detail in our earlier publications [12, 13, 14, 15]. In PbA, when an unknown word is presented as input to the system, so-called full pattern matching between the input letter string and dictionary entries is performed, starting with the initial letter of the input string aligned with the end letter of the dictionary entry. If common letters are found in matching positions in the two strings, their corresponding phonemes (according to the prior alignment) and information about their positions in the input string are used to build a pronunciation lattice, as detailed below. One of the two strings is then shifted relative to the other by one letter and the matching process continues, until the end letter of the input string aligns with the initial letter of the dictionary entry.

The pronunciation lattice is a directed graph that defines possible pronunciations for the input string, built from the matching substring information. A lattice node represents a matched letter, L_i, at some position, i, in the input. The node is labelled with its position i and the phoneme corresponding to L_i in the matched substring, P_{im} say, for the mth matched substring. An arc is labelled with the phonemes intermediate between P_{im} and P_{jm} $(j > i)$ in the phoneme part of the matched substring and the frequency count, increasing by one each time the substring with these phonemes is matched during the search through the lexicon. Arcs are directed from i to j. If the arcs correspond to bigrams, the arcs are labelled only with the frequency. (The string of phonemes intermediate between P_{im} and P_{jm} is empty.) Phonemes P_{im} and P_{jm} label the nodes at each end of the arc, i.e., i and j respectively. Additionally, there is a *Start* node at position 0 and an *End* node at position equal to the length of the input string plus one.

Finally, the decision function identifies the 'best' candidate pronunciation of the input according to some criterion. Possible pronunciations correspond to the string assembled by concatenating the phoneme labels on the nodes or arcs in the order that they are traversed in moving through the lattice from *Start* to *End*. If there is just one candidate corresponding to a unique shortest path, this is selected as the output. If there are tied shortest paths, five different scoring strategies are applied and the winning candidate selected on the basis of their rank [13, 14].

The major modification in converting PbA to SbA is to represent all junctures between phonemes explicitly. This representation must be different in the case of:

1. input words, where the syllabification is unknown;
2. lexical entries, where it is known;
3. the SbA output, where it is inferred.

For example, the input pronunciation /@bi/[4] (<abbey>) is expanded to /@ ∗ b ∗ i/. Here the '∗' symbol merely indicates the *possibility* of a syllable boundary. On the other hand, a dictionary entry such as /@bncrmL/ <abnormal> is expanded to /@ ∗ b | n ∗ c ∗ r | m ∗ L/. In this case, the '∗' symbols indicate the known absence of a syllable boundary. During pattern matching, '∗' in the input is allowed to match either with '∗' or with '|' in the dictionary entries. A '∗ − ∗' match is entered into the syllabification lattice as a '∗' whereas a '∗ − |' match is entered into the syllabification lattice as a '|'. The syllabification lattice has exactly the same form as the pronunciation lattice, except that '∗' is explicitly represented as an input symbol (labelling nodes), '∗' and '|' are explicitly represented as possible output symbols (labelling arcs). From here, the process proceeds exactly as for PbA, eventually producing as output a syllabified version of /@bi/ <abbey> such as /@ ∗ b | i/, from which the '∗' symbols are removed to yield the final output /@b | i/.

Figure 1 shows the syllabification lattice for the word <phonograph>, we have three candidate syllabifications. Of course, candidate syllabifications are not necessarily distinct: different shortest paths can obviously correspond to the same syllabified string.

In our previous syllabification work using analogy [15], we obtained best results by combining only 3 of the 5 scoring strategies when choosing between tied shortest paths. These were the product of arc frequencies, the frequency of the same pronunciation, and the 'weak link' (see [13] and [14] for full specification). Accordingly, in this work, these same three scoring strategies are used exclusively, and combined by rank fusion, for SbA.

3.3. Look-up procedure

This method was originally proposed by [16] as a means of letter-to-phoneme conversion (i.e., automatic pronunciation), where it was shown to be superior to NETtalk, the well-known neural network [10]. It was then adapted for the syllabification process and presented in the comparison of syllabification algorithms for Dutch spellings by [17]. The first step is to construct a table encoding the knowledge implicit in the training set by converting each syllabified entry into a series of N-grams. Each N-gram has a left and right context and a central, 'focus' character. The length of the N-gram (i.e., N) is equal to the sum of the sizes of the left and right contexts plus one (the focus character).

For example, if the syllabified word /KId | ni/ (<kid | ney>) is part of the training corpus, then with a left context of 1 character and a right context of 2 characters, the N-grams (or 4-grams in this case) for this word would be: <– KId>, <KIdn>,

[3] ...designated as syllabic by Fisher. These are: 'ux', 'ih', 'ix', 'ey', 'eh', 'ae', 'aa', 'aax', 's', 'ao', 'ow', 'uh', 'uw', 'ay', 'oy', 'aw', 'er', 'axr', 'ax', 'ah', 'el', 'em', and 'en' using his phoneme notation.

[4] Examples from this point on use the phoneme set from *Webster's Pocket Dictionary*.

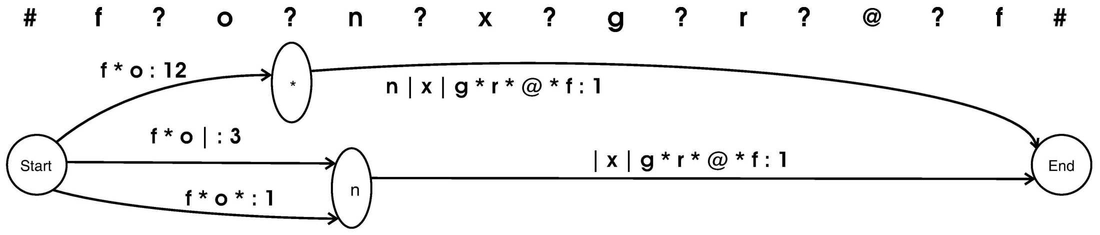


Figure 1: *Example of the syllabification lattice for the word* phonograph. *For simplicity, only arcs contributing to the shortest (length-2) paths are shown. The '?' symbol represents the junctures and the phoneme symbols are those employed by Sejnowski and Rosenberg.*

<Idni>, <dni–>, and <ni– –>. That is, to allow every character to be a focus character, there is an N-gram for each character in a word. When the focus character has no left context (as in <–KId>) or right context (as in <ni– –>), the character positions in the context are filled with null characters. Each N-gram is stored in the table along with the corresponding juncture class, i.e., the syllabification information.

Once the construction of the look-up table is complete, words for which the syllabification is unknown can be syllabified based on the information in the table. Input words are broken down into a set of N-grams in the same manner described above for table construction. The table is then searched for the closest matches to each N-gram. When found, closest matches are examined to determine whether the majority has, or does not have, a syllable boundary following the focus character. If the majority has a syllable boundary, a syllable boundary is placed at the appropriate position in the word; otherwise, a non-syllable boundary is placed at that position.

The process of determining which N-grams in the pre-compiled look-up table fit best a given N-gram is described in Algorithm 1. Here, `NgramT` is a given N-gram stored in the table and `NgramS` is an N-gram to be syllabified. It follows that `NgramT[i]` is the ith position in the N-gram (for example, `NgramT[1] = m` when `NgramT` is <midn>). The closest-fit N-grams are those with the highest `MatchValue`.

Algorithm 1 : Computation of best-fit N-gram in the look-up procedure.

```
FindMatchValue(weights, NgramT, NgramS)
MatchValue := 0
for i := 1 to length(weights) do
  if (NgramT[i] = NgramS[i]) then
    MatchValue := MatchValue +
    weights[i]
  end if
end for
```

We ran the look-up procedure using all 15 different sets of weights presented in the original description of the method [16].

3.4. Exemplar-Based Generalization

The version tested here (also known as IB1-IG) is due to [18]. It operates in a manner similar to the look-up procedure with the only difference being the weights used to determine the closest-fit N-grams. In this method, the weights are calculated with a function that determines the relative importance of each position in the N-gram (i.e., phoneme positions). The process of determining the weights is based on the concept of information entropy by using information from the table of stored N-grams. Each position in an N-gram is considered to contribute a real-valued amount of information to the process of determining the placement of a syllable boundary. This value can be determined via the series of steps presented below.

First, the entropy of the entire table of N-grams extracted from the training corpus is calculated. Essentially, Daelemans, van den Bosch and Weijters define database (or look-up table) information entropy as "the number of bits of information needed to know the decision [whether a syllable boundary should be placed after the focus character or not] of a database given a pattern [or N-gram]." This is calculated as:

$$E(D) = -\sum_{i=1}^{2} P_i \log_2 P_i \tag{1}$$

where $E(D)$ is the information entropy of database D, P_1 is the probability of an N-gram being associated with a syllable-boundary decision, and P_2 is the probability of an N-gram being associated with a non-syllable-boundary decision. As there are only two possibilities—to place or not to place a syllable boundary after the focus character—equation (1) can also be written as:

$$\begin{aligned} E(D) &= -\alpha \log_2 \alpha + \beta \log_2 \beta \\ &\text{where } \alpha = \frac{N_S}{N_T} \text{and } \beta = \frac{N_{\neg S}}{N_T} \end{aligned} \tag{2}$$

where N_S is the number of stored N-grams that have a syllable boundary following the focus character, $N_{\neg S}$ is the number of stored N-grams that do not have a syllable boundary following the focus character, and N_T is the number of stored N-grams (i.e., $N_S + N_{\neg S}$).

From equation (2), the information gain of each position in an N-gram can now be determined. This requires two additional equations. The first computes the average information entropy at position f in an N-gram, $E(D_f)$, by taking the "information entropy of the database [or table] restricted to each possible value [or character] for the [position in the N-gram]." This is given by:

$$E(D_f) = \sum_{c \in V} E(D_{f=c}) \frac{\text{card}(D_{f=c})}{\text{card}(D)}$$

where $D_{f=c}$ is the set of those N-grams in the table that have character c at position f, V is the set of characters that occur at position f in a N-gram, and card( ) is the cardinality of a set (i.e., card(D) is the total number of N-grams in database D).

The second equation necessary for calculating the information gain $G(f)$ at a given position f in an N-gram is:

$$G(f) = E(D) - E(D_f)$$

To run this method, we first followed Daelemans, van den Bosch and Weijters and used the same values of N as in their work, namely 3, 5 and 7 with the focus letter in the middle of the N-gram. In addition to these values, we extended the study to use N-grams of size 9 and 11 (with left and right contexts of 4 and 5 respectively).

4. Results

For the rule-based method, there is no difficulty in evaluating syllabification performance on each of the three datasets in their entirety. For data-driven methods, we use the well-established leave-one-out procedure, whereby each word is removed from the corpus in turn, and its syllabification inferred from the remaining words.

	Accuracy			
Algorithm	Word	Juncture	|	*
Fisher/Kahn				
Basic	54.23	78.93	62.63	85.34
OCS	54.14	77.47	59.84	84.41
OCS with stress	68.97	86.41	75.65	90.64
SbA	88.53	96.02	92.29	97.50
Look-up Table				
1st, version 10	80.20	94.95	90.51	96.70
2nd, version 8	79.75	94.90	90.49	96.63
3rd, version 13	79.40	94.78	89.90	96.70
Exemplar-based				
$N = 5$	76.47	94.17	87.54	96.79
$N = 7$	79.37	94.80	88.92	97.11
$N = 9$	79.36	94.78	89.04	97.04
$N = 11$	79.10	94.71	88.91	96.99

Table 2: *Syllabification results (percentage correct) on the* S&R *database for word and juncture accuracy.*

	Accuracy			
Algorithm	Word	Juncture	|	*
Fisher/Kahn				
Basic	58.02	81.34	67.04	86.91
OCS	52.58	75.93	57.16	83.24
OCS with stress	63.37	83.40	70.49	88.43
SbA	85.88	94.87	90.32	96.64
Look-up Table				
1st, version 10	75.71	93.41	87.93	95.54
2nd, version 8	75.37	93.36	87.96	95.47
3rd, version 11	74.86	93.26	87.44	95.53
Exemplar-based				
$N = 5$	72.92	92.81	85.04	95.84
$N = 7$	74.92	93.17	85.76	96.05
$N = 9$	82.90	95.54	89.23	97.71
$N = 11$	74.87	93.12	85.86	95.96

Table 3: *Syllabification results (percentage correct) on the* Wordsmyth *database for word and juncture accuracy.*

Tables 2, 3 and 4 show the results for the various automatic syllabification methods on the *S&R*, *Wordsmyth* and *Intersection* databases. For table look-up, the three sets of weights which provided the best results (for each dictionary) are presented in the tables[5]. Results were obtained for N-grams from $N = 5$ up to $N = 11$ for the exemplar-based approach. As expected, results were poor for $N = 3$ as insufficient context is captured around the focus phoneme, and by $N = 11$ the algorithm indicates that performance was falling off. For the Fisher/Kahn system, there was no difference between the results when stress was provided and when it was not for the Basic (slow) rate of speech. However, this was not the case for the ordinary conversational speech condition, where the inclusion of stress improves the results.

	Accuracy			
Algorithm	Word	Juncture	|	*
Fisher/Kahn				
Basic	63.40	83.54	67.80	88.97
OCS	60.90	79.68	60.07	86.44
OCS with stress	74.42	88.14	76.56	92.13
SbA	91.08	96.82	92.90	98.17
Look-up Table				
1st, version 10	83.66	95.74	90.58	97.52
2nd, version 8	83.60	95.76	90.66	97.52
3rd, version 11	82.71	95.55	89.99	97.47
Exemplar-based				
$N = 5$	81.26	95.20	88.51	97.50
$N = 7$	83.12	95.56	89.28	97.73
$N = 9$	82.90	95.54	89.23	97.71
$N = 11$	82.87	95.52	89.23	97.69

Table 4: *Syllabification results (percentage correct) on the* Intersection *database for word and juncture accuracy.*

Results are remarkably consistent across dictionaries. The rule-based method (Fisher/Kahn) is much worse than the data-driven methods. In regards to the data-driven methods, it is difficult to choose between the best table look-up and exemplar-based results although the former does better on two of the three dictionaries. The most striking result, however, is the obvious superiority of SbA.

These tables also show junctures-correct performance overall, as well as the percentages of correct syllable (|) and non-syllable (*) identifications. For all methods, non-syllable boundary identification is less error prone than syllable boundary detection. It seems that all methods are conservative in their placement of syllable boundaries, which are rarer than non-syllable boundaries, resulting in a preponderance of false negative errors over false positives.

5. Conclusions

Automatic syllabification is an important but difficult problem that has implications on pronunciation generation for text-to-speech synthesis and pronunciation modeling in speech recognition. There are essentially two possible approaches to automatic syllabification: rule-based and data-driven.

In this work, we have compared one rule set based on expert knowledge and three data-driven methods based on automatic inference from a corpus of already-syllabified words. In the latter case, the issue of a gold standard arises. We attempt to address this by using two independent dictionaries of syllabified

[5] Version 8:[1,4,16,4,2]; Version 10:[1,4,16,64,16,5,1]; Version 11:[1,4,16,64,256,64,17,4] and Version 13:[4,16,64,256,64,17,4,1].

words. We also use the 'intersection' of entries in the different dictionaries as a separate corpus which ought to be closer to a gold standard than either of the individual contributors, since it does not include words on which they disagree. In this work, we have used two independent dictionaries (*S&R* and *Wordsmyth*) and their intersection. The four methods studied are the rule set from Fisher/Kahn, a table look-up method developed by Weijters, the exemplar-based method of Daelemans, van den Bosch and Weijters and syllabification by analogy (SbA) from Marchand and Damper. In each case, performance is evaluated across the whole of each available corpus.

Syllabification performance is found to be very consistent across dictionaries in terms of the relative merits of the four techniques. The knowledge-based rule set performs poorly compared to the data-driven methods. Among the data-driven methods, SbA is easily the best. With regards to the dictionaries, best performance is obtained on the *Intersection* dictionary—probably because the intersection process removes idiosyncratic entries from *S&R* and *Wordsmyth*.

We believe there are sound reasons to expect the pattern of results seen here and the same trends showed on the problem of automatic pronunciation [19]. In our opinion, expert rule-based approaches suffer many drawbacks, including lack of conformance with real data, the limited ability of human experts to distinguish real from apparent regularities in very large datasets (like the effectively unbounded whole of natural language), and a tendency to over-rate dramatically the strength of weak, tentative linguistic theories.

6. Acknowledgements

This work was supported in part by funding from the Natural Sciences and Engineering Research Council of Canada (NSERC). In addition, the second author was funded by the National Research Council (NRC) Graduate Student Scholarship Supplement Program (GSSSP), and an Izaak Walton Killam Predoctoral Scholarship.

7. References

[1] E. Pulgram, *Syllable, Word, Nexus, Cursus*. The Hague, The Netherlands: Mouton, 1970.

[2] E. Selkirk, "The syllable," in *The Structure of Phonological Representations*, H. van der Hulst and N. Smith, Eds. Dordrecht, The Netherlands: Foris, 1982, vol. 2, pp. 337–383.

[3] K. J. Kohler, "Is the syllable a phonological universal?" *Journal of Linguistics*, vol. 2, pp. 207–208, 1966.

[4] D. Crystal, *A First Dictionary of Linguistics and Phonetics*. London: André Deutsch, 1980.

[5] R. Treiman and A. Zukowski, "Toward an understanding of English syllabification," *Journal of Memory and Language*, vol. 29, no. 1, pp. 66–85, 1990.

[6] J. Goslin and U. H. Frauenfelder, "A comparison of theoretical and human syllabification," *Language and Speech*, vol. 44, no. 4, pp. 409–436, 2000.

[7] K. Müller, B. Möbius, and D. Prescher, "Inducing probabilistic syllable classes using multivariate clustering," in *Proceedings of 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong, China, 2000, pp. 225–232.

[8] D. Kahn, *Syllable-Based Generalizations in English Phonology*. Bloomington, IN: Indiana University Linguistics Club, 1976.

[9] G. N. Clements, "The role of the sonority cycle in core syllabification," 1988, working Papers of the Cornell Phonetics Laboratory, WPCPL No. 2, Research in Laboratory Phonology, Cornell University, Ithaca, NY.

[10] T. J. Sejnowski and C. R. Rosenberg, "Parallel networks that learn to pronounce English text," *Complex Systems*, vol. 1, no. 1, pp. 145–168, 1987.

[11] R. I. Damper, Y. Marchand, J.-D. S. Marsters, and A. I. Bazin, "Aligning text and phonemes for speech technology applications using an EM-like algorithm," *International Journal of Speech Technology*, vol. 8, no. 2, pp. 149–162, 2005.

[12] R. I. Damper and J. F. G. Eastmond, "Pronunciation by analogy: Impact of implementational choices on performance," *Language and Speech*, vol. 40, no. 1, pp. 1–23, 1997.

[13] Y. Marchand and R. I. Damper, "A multistrategy approach to improving pronunciation by analogy," *Computational Linguistics*, vol. 26, no. 2, pp. 195–219, 2000.

[14] R. I. Damper and Y. Marchand, "Information fusion approaches to the automatic pronunciation of print by analogy," *Information Fusion*, vol. 71, no. 2, pp. 207–220, 2006.

[15] Y. Marchand and R. I. Damper, "Can syllabification improve pronunciation by analogy?" *Natural Language Engineering*, vol. 13, no. 1, pp. 1–24, 2007.

[16] A. Weijters, "A simple look-up procedure superior to NETtalk?" in *Proceedings of International Conference on Artificial Neural Networks (ICANN-91)*, vol. 2, Espoo, Finland, 1991, pp. 1645–1648.

[17] W. Daelemans and A. van den Bosch, "Generalisation performance of backpropagation learning on a syllabification task," in *TWLT3: Connectionism and Natural Language Processing*, M. F. J. Drossaers and A. Nijholt, Eds. Enschede, The Netherlands: Twente University, 1992, pp. 27–37.

[18] W. Daelemans, A. van den Bosch, and T. Weijters, "IGTree: Using trees for compression and classification in lazy learning algorithms," *Artificial Intelligence Review*, vol. 11, no. 1–5, pp. 407–423, 1997.

[19] R. I. Damper, Y. Marchand, M. J. Adamson, and K. Gustafson, "Evaluating the pronunciation component of text-to-speech systems for English: A performance comparison of different approaches," *Computer Speech and Language*, vol. 13, no. 2, pp. 155–176, 1999.

Voice Building from Insufficient Data – Classroom Experiences with Web-based Language Development Tools

John Kominek, Tanja Schultz, Alan W Black

Language Technologies Institute
Carnegie Mellon University, Pittsburgh, USA
{jkominek,tanja,awb}@cs.cmu.edu

Abstract

To make the goal of building voices in new languages easier and more accessible to non-experts, the combined tasks of phoneme set definition, text selection, prompt recording, lexicon building, and voice creation in Festival are now integrated behind a web-based development environment. This environment has been exercised in a semester-long laboratory course taught at Carnegie Mellon University. Here we report on the students' efforts in building voices for the languages of Bulgarian, English, German, Hindi, Konkani, Mandarin, and Vietnamese. In some cases intelligible synthesizers were built from as little as ten minutes of recorded speech.

1. Introduction

In the past decade, the performance and capability of automatic speech processing systems, including speech recognition and speech synthesis, has matured significantly. With the addition of machine translation linking input to output, the prospect of two people of different languages communicating together becomes a tantalizing possibility.

In light of the increasing trend towards Globalization, it has become important to support multiple input and output languages beyond the dominant Western languages (English, German, Spanish, etc.). Due to the high costs and long development times typical of ASR, TTS, and MT, the need for new techniques to support the rapid adaptation of speech processing systems to previously uncovered languages becomes paramount [1].

The 3-year project and software toolkit known as SPICE[1] is an initiative intended to dramatically reduce the difficulty of building and deploying speech technology systems. It is designed to support any pair of languages in the world for which a writing system exists, and for which sufficient text and speech resources can be made available. This is accomplished by integrating and presenting in a web-based development interface several core technologies that have been developed at Carnegie Mellon University. These include the Janus ASR trainer and decoder [2], GlobalPhone multilingual inventory and speech database [3], CMU/ Cambridge language modeling toolkit [4], Festival speech synthesis software [5] and FestVox voice building toolkit [6], Lexlearner pronunciation dictionary builder [7], Lemur information retrieval system [8], and CMU statistical machine translation system [9].

A new addition to this software suite is an embeddable Javascript applet that provides within-browser recording and playback facilities. By this means any two people in the word who would previously be separated by a language barrier can potentially speak with each other through our recognition/ translation/synthesis server (presuming access to a compliant Internet browser.) Also, our in-browser recorder provides a solution to an enduring problem of system development: namely, that of speech collection. It is no longer necessary that the system developer be able to locate native speakers of a particular language living nearby.

The SPICE software toolkit is in an early stage of development. To stress and evaluate the current state of the system, a hands-on laboratory course "Multilingual speech-to-speech translation" was offered for credit at Carnegie Mellon University. It ran for a single semester from January to May 2007, taught by three instructors: Tanja Schultz (ASR), Alan W Black (TTS), and Stephan Vogel (MT) [10]. All participants are graduate students studying language technologies. Students were paired into teams of two and asked to create working speech-to-speech systems, for a limited domain of their choosing, by the end of the course. The languages tackled were English, German, Bulgarian, Mandarin, Vietnamese, Hindi, and Konkani, a secondary language of India that does not have its own writing system but is transcribed through various competing foreign scripts. Interim results from this course are described in [11].

Here we report on the student's attempts in building synthetic voices in their language, including the role that pronunciation-lexicon creation played in their efforts. Creating a speech-to-speech translation system is a very ambitious task, meaning that only a portion of their time was allocated to TTS. Consequently, the students attempted to make good out of less material than is typical, i.e. much less than the one hour of speech of an Arctic-sized database [12]. Some hopeful students relied on less than 10 minutes of speech to build a voice from scratch – insufficient data, without doubt, hence the title of this paper.

2. TTS as a part of Speech-to-Speech

The speech synthesis system at the heart of SPICE is the CLUSTERGEN statistical parametric synthesizer [13], now released as part of the standard Festival distribution. We chose this technology because experience has shown that it (and the similar HTS [14]) degrades gracefully as the amount of training data decreases.

[1] Speech Processing – Interactive Creation and Evaluation Toolkit for new Languages.

Unlike the normal development procedure, however, in which the user executes a series of Unix scripts and hand-verifies the intermediate results, in SPICE all operations are orchestrated behind a web interface. The ASR and TTS components share resources. This includes text collection, prompt extraction, speech collection, and phoneme selection, and dictionary building. The interdependencies are depicted below.

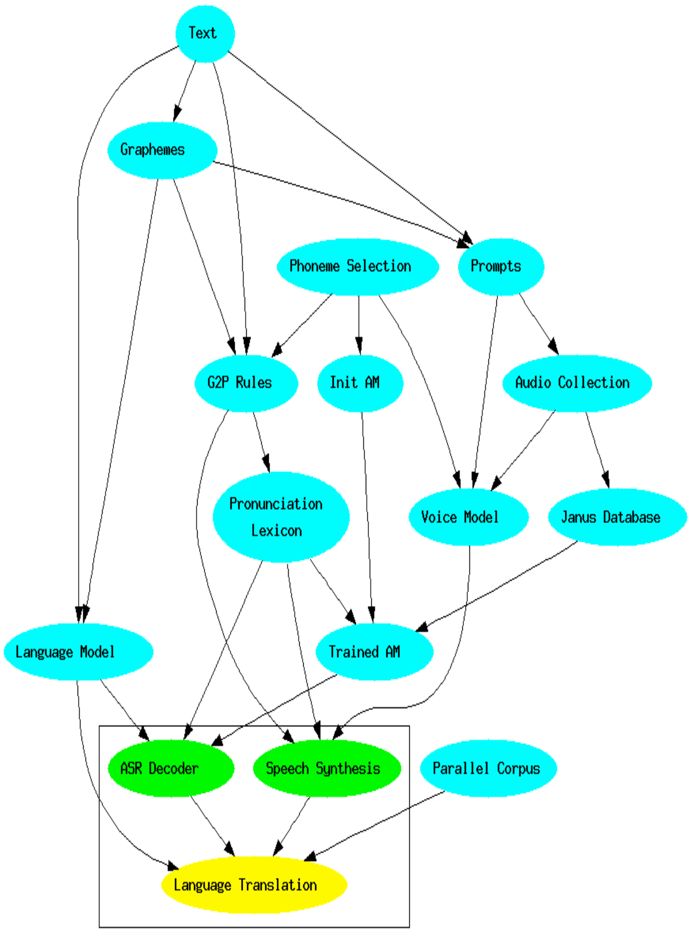
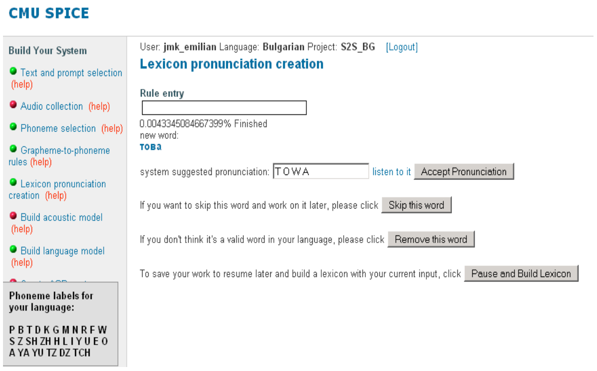


Figure 1. *High level component dependencies in the SPICE system.*

2.1. Development Work Flow – TTS

- **Text collection**. The development process begins with text collection. Text may be collected from the world-wide web by pointing the SPICE webcrawler at a particular homepage, e.g. of a online news site. For additional control, the user can upload a prepared body of text. All of the students used this option, so that they could verify that the text is all within their chosen domain. Given the multitude of character encodings used worldwide, we imposed the constraint that the text had to be encoded in utf-8 (which includes ASCII, but not the vast population of 8-bit code pages.). Some began with large collections; others small, *cf.* section 3.
- **Prompt selection.** Typically one would convert the collected text to phonemes, then select sentence-length utterances that provide a balanced coverage of predicted acoustics, i.e. of diphones or triphones. At this stage though we have no means of predicting acoustics, and so prompts are selected on the basis of grapheme coverage.
- **Audio collection.** Depending on the source text, the prompt list is of varying length. We instructed the students to record at least 200 sentence-length prompts, adding material as needed if they had a shortfall.
- **Grapheme definition.** Once text is provided, the SPICE software culls all of the characters and asks the user to define basic attributes of each. This includes class membership (letter, digit, punctuation, other), and casing.
- **Phoneme selection.** In this, perhaps the most critical stage, students select and name a phoneme set for their language. The interface assists this by providing a list of available phonemes laid out similar to the official IPA charts. An example wavefile is available for most phonemes so help in the selection. While this interface was intended to allow a phoneme set to be built up from scratch, not one student did that. Instead, they started from one or two reference lists and used the interface to make refinements.
- **G2P rules.** The development of grapheme-to-phoneme (or letter-to-sound) rules proceeds in a two-stage process. First, the user is asked to assign a default phoneme to each grapheme, including those that are unspoken (e.g. punctuation). Explaining his request required multiple clarifications, as students tended initially to provide word sound-outs – declaring, for example, that 'w' is not associated with /W/ but is pronounced /D UH B AH L Y UW/ “double u”.
- **Pronunciation lexicon.** The second phase of G2P rule building goes on behind the scenes as the user accumulates their pronunciation lexicon. Words are selected from the supplied text in an order that favors the most frequent words first. Each word is presented with a suggested pronunciation, which the user may accept or manually correct. As an additional aid, each suggestion is accompanied by a wavefile synthesized using a universal discrete-phoneme synthesizer. Figure 2 shows a screen shot. After each word is supplied to the system, the G2P rules are rebuilt, thereby incrementally providing better predictions, similar to that of [15]. When the user is satisfied with the size of their lexicon the final G2P rules are compiled. These are then used to predict pronunciations for all the remaining lower frequency words.
- **Speech synthesis.** With the necessary resources provided, the standard FestVox scripts have been modified to a) automatically import the phoneme set and look up IPA feature values, b) import the pronunciation dictionary, and c) compile the G2P rules into a transducer. The recorded prompts are labeled and utterance structures created. With this the language-specific F0, duration, and sub-phonetic spectral models are trained. The user can then test their synthesizer by entering text into a type-in box.



User: jmk_emilian Language: Bulgarian Project: S2S_BG

Lexicon pronunciation creation

0.0043345084667399% Finished

new word:

тоBа

system suggested pronunciation: TOWA

If you want to skip this word and work on it later, please click

If you don't think it's a valid word in your language, please click

To save your work to resume later and build a lexicon with your current input, click

Phoneme labels for your language:

P B T D K G M N R F W
S Z SH ZH H L I Y U E O
A YA YU TZ DZ TCH

Figure 2. *Web interface to lexicon builder.*

3. Descriptive Statistics

Prior to tacking their own language, students were encouraged to complete an English walk-though. The “English walk-through” is a prepared body of text and audio that is accompanied by step-by-step instructions. The walk-through material is based on the first 200 utterances of the rms Arctic database [12]. Having been designed for phonetic coverage and balance, the subset offered sufficient support for both ASR acoustic model adaptation and TTS construction; 200 utterances approximates the minimum required for success. In the tables below the figures for English are from this database.

3.1. Corpus Size

There are several measures when referring to the size of database using in voice construction. First is the original text corpus, from which the SPICE tools select a prompt list suitable for recording. Students are permitted to modify their prompt list, adding sentences they want covered in their domain and deleting those deemed inappropriate. During recording it is not unusual for additional prompts to be skipped (due to containing unpronounceable character sequences). As a rule of thumb, students were advised to record a minimum of 200 sentence-length prompts, even though the English voice was built from 96.

Language	***text corpus***			***selected prompts***		
		word counts			*word counts*	
	Utts	*types*	*tokens*	*Utts*	*types*	*tokens*
Bulgarian	23049	69607	508349	563	928	3517
English	200	798	1792	96	446	864
German	46328	49304	446765	435	1003	2913
Hindi	1543	558	12185	192	557	1524
Konkani	761	2008	3422	200	503	890
Mandarin	9925	22252	196120	199	1608	3669
Vietnamese	203	408	1520	203	400	1524

Table 1. *Size of language corpora in utterances and words (left), and of the selected prompt list (right).*

Language	***Prompts***	***Time***
Bulgarian	358	14:42
English	96	4:00
German	424	22:33
Hindi	191	9:51
Konkani	195	7:49
Mandarin	199	37:47
Vietnamese (rec)	203	10:41
Vietnamese (built)	77	3:38

Table 2. *Size of speech recordings (time in mm:ss). Whitespace was not trimmed from the prompts. Due to gaps in the lexicon, the Vietnamese voice was built from only a third of the available recordings.*

3.2. Word and Character Coverage

When building a voice for a new (i.e. previously uncovered) language, the development of a pronunciation dictionary is a major element of this task. In the name of expediency, pronunciations for the 757 words needed for the English voice were extracted from CMU-DICT [16]. The students did not have this luxury and instead used the *lexlearner* component of SPICE to create a dictionary based on their supplied text. The one exception is Mandarin, for which the student uploaded a larger prepared dictionary.

Students were allowed to modify, supplant, and even replace the automatically selected prompts with their own list. This was true for Bulgarian, Hindi, and Vietnamese. Such allowance is a consequence of working in multi-purpose system: the language modeling component of ASR generally requires a large body of text, whereas TTS can often be improved if the text is targeted to the intended domain. The Hindi text for example was drawn from the Emille corpus [17], but the prompt list targeted the domain of cooking, restaurants and food recipes. In such cases there is a mismatch between the intended usage and assembled lexicon. Consequently the voice is forced to rely on grapheme-to-phoneme rules to “carry the day.”

Language	***Dict***	***Text corpus***		***Selected prompts***	
	words	*types*	*tokens*	*types*	*tokens*
Bulgarian	396	0.57	49.36	0.0	0.0
English	757	95.55	99.77	100.0	100.0
German	1037	1.96	60.39	31.80	66.36
Hindi	356	64.03	86.87	0.0	0.0
Konkani	318	14.54	15.93	16.70	14.94
Vietnamese	288	70.34	59.54	1.25	0.46

Table 3. *Dictionary coverage of the original text corpus and selected prompts, in percent. For Bulgarian, Hindi,and Vietnamese the recorded prompts were not derived from the text.*

During construction of a pronunciation lexicon, it is the system that selects words and asks the user for the correct sequence of phonemes. Words are ordered from the most to least frequently occurring. The benefit of this can be seen for Hindi, Bulgarian, and German. In German, the 1000 most frequent words is enough to cover 60% of the 450k text. However, coverage of the prompts is poor when the student opted not to go with the automatically selected list (breaking our original assumptions). Transcripts for the prompts then depend on the fidelity of grapheme-to-phoneme rules learned from a few hundred words. Since this is not optimal, a better alternative is to solicit coverage of the recorded prompts first, before proceeding onto the larger body of text, is necessary.

Grapheme coverage may be even more important than word coverage, due to the fact that Festival will reject an entire utterance if it contains graphemes with an undefined pronunciation. This information is solicited during the development process (see section 2.1), but the system does not have checks in place to strictly enforce complete coverage. Oversights thus slip through, particularly when the student appends data to their text collection without revisiting the character definition protocol. The languages where this became problematic were Konkani (uppercase letters) and Vietnamese (various omissions).

Language	***Graphemes***	***Text corpus***	
	count	*types*	*tokens*
Bulgarian	74	85.14	99.81
English	51	100.0	100.0
German	53	100.0	100.0
Hindi	67	85.71	99.82
Konkani	52	57.69	93.14
Vietnamese	57	80.70	86.05

Table 4. *Grapheme coverage. Values are in percent.*

4. G2P Rule Learning

The complexity of the relation between graphemes and phonemes of a language of course varies dramatically from language to language. Of the languages described here, Bulgarian has the most straightforward, while English is highly irregular and Mandarin, being ideographic, exhibits no relation at all. Clearly, languages with a simple relation extrapolate more readily to unseen items, thus increasing the chances of a successful voice. And as previously pointed out, those projects with poor word coverage from the lexicon depend heavily on the G2P rules. From Table 3 these are Bulgarian, Hindi, Konkani, and Vietnamese.

The relative difficulty of languages (excepting Mandarin) can be seen by comparing the number of rules and rate of G2P rule growth with respect to the vocabulary size. These values are summarized in Table 5. Average letter perplexity – another indicative figure – is also included. As expected, English has the most complex G2P relationship. Bulgarian has a script that is nearly perfectly phonetic.

Language	***G2P Rules***			
	300 words	all words	*rules / letter*	*ave letter perplex.*
Bulgarian	51	54	1.019	1.002
English	360	727	25.07	3.350
German	236	523	4.023	1.932
Hindi	190	212	3.655	1.693
Konkani	205	223	7.964	2.356
Vietnamese	139	139	2.837	2.524

Table 5. *Comparison of G2P complexity. Note that Vietnamese is limited to 288 words.*

4.1. Case Study: Hindi

To demonstrate the effort required to build a challenging lexicon, we report on the case of Hindi. Text was extracted from the Emille Lancaster Corpus [17], comprising 210 thousand words and 10.2 million tokens from the domain of current news. The Hindi speaker in the course used the SPICE toolkit to perform the following tasks: a) provide default letter-to-sound rules for each grapheme, b) provide pronunciations for the most frequent 200 words, c) correct automatically generated pronunciations for the next 200 words, and d) correct automatically generated pronunciations for the 200 words randomly selected from the remainder of the corpus. Error rates for these 200 words were then compared for three cases: a) G2P rules based solely on the default assignment, b) rules trained on the first 200 words, and c) words trained on the first 400 words. As summarized in Table 6, word accuracy increased from 23 to 41 to 51%. Additional detail is plotted in Figure 3.

G2P Rules		***Test Set***		
Training Size	Num Rules	1-200 words	201-400 words	400+ words
Default	49	52.7	32.3	22.6
LTS-200	127		51.0	40.9
LTS-400	216			50.5

Table 6*: Word accuracy on 187-word test set, for letter-to-sound rules based on 0, 200, and 400 training words (Emille corpus).*

The impact of these numbers can be seen in Figure 4, where projected token coverage of the Emille Hindi corpus is compare to the optimal coverage offered by a complete dictionary.

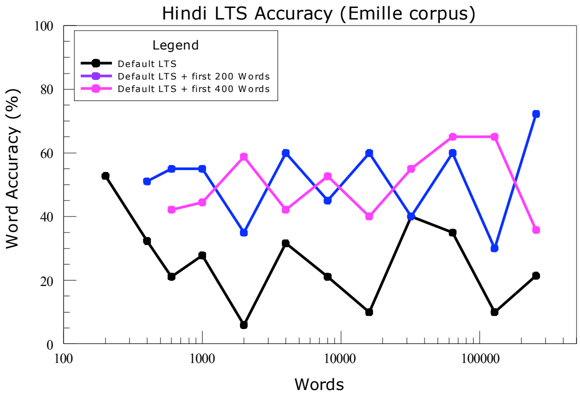


Figure 3: *Phone Error Rate of randomly sampled Hindi words taken from blocks on the log frequency scale. Each dot represents20 words.*

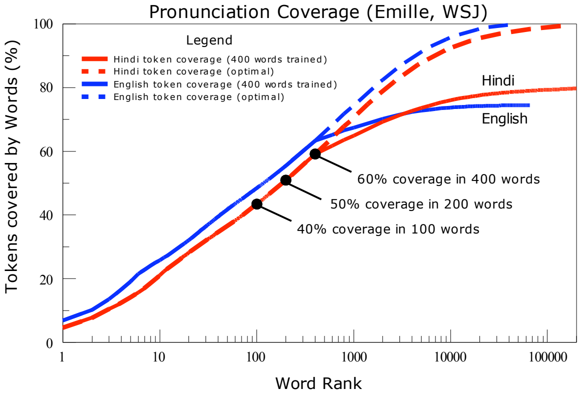


Figure 4: *Coverage of word tokens for Hindi when trained on the 400 most frequent words. A compatible curve for English is provided as a reference.*

5. Voice Quality Assessment

Each student was asked to provide an subjective impression of their voice, based on synthesis of in-domain prompts and of “random” things they think to type in. Three of the voices were a success, with the German voice receiving the most positive feedback (three people in the course speak German) – though see section 5.1 for more quantitative measurement. The Hindi voice was also deemed good, with the caveat that sentence-initial words tended to be confusing. The relative success for this pair can be attributed to reliable G2P rules, in order to go beyond the explicit lexicon (see Table 3). The English voice, built from just four minutes of speech was surprisingly understandable, though words outside the lexicon words were not uncommonly mispronounced.

The Vietnamese voice was poor – our two native speakers had trouble understanding what was said – though the tone contour was often correct. Since the Vietnamese voice was built from a mere three minutes of speech this result is understandable. Less understandable is the case of Mandarin, which for the amount data available *should* have been a good voice. We don't know yet whether this is attributable to errors in processing (i.e. bugs in the software), or some deep limitation confronted by tonal languages. The Konkani voice has been jokingly heralded as the best of its kind in the world (being the only one!) but is, the speaker admitted, incomprehensible. The details of why need to be determined. At this point we can safely conclude that 15% word coverage of the prompt list is insufficient for this language.

Language	***Time***	***Impression of Quality***
Bulgarian	14:42	(no feedback at time of writing)
English	4:00	understandable, mispronounces words
German	22:33	good, including prosody
Hindi	9:51	good, most words understood
Konkani	7:49	incomprehensible
Mandarin	37:47	fair
Vietnamese	3:38	poor

Table 7: *Impressionistic quality of voices as assessed by native speakers. For convenience the total length of each database is repeated from Table 2.*

5.1. Word Comprehension

To establish a more quantitative assessment of intelligibility, we chose two of the better synthesizers for listening tests: German and Hindi. One of the German students provided transcriptions of the German voice. Twenty sentences were randomly selected from within the application domain and synthesized, with an additional four extracted from out of domain. The tester was allowed to listen to the synthesized sentences more than once, and to note which words became more clear after multiple listening. Transcripts were double-check for typographic errors. The Hindi listener was not a part of the class and thus was not familiar with the domain.

For the German in-domain sentences, 76% of the words were transcribed correctly, versus 55% for the out-of-domain. For Hindi the corresponding rates are very similar: 76% and 54%. Details are tabulated in Table 8.

Words correct (German)					
1-4	5-8	9-12	13-16	17-20	21-24
3/5	7/8	8/8	4/5	8/8	9/11
1/6	6/8	3/6	3/6	6/7	5/9
4/6	2/8	5/6	3/5	8/9	5/11
8/8	4/6	7/7	9/9	6/7	3/9
Words correct (Hindi)					
7/7	6/6	4/8	6/6	4/4	4/6
4/6	5/12	3/5	9/11	5/6	0/6
10/10	3/7	5/8	4/6	6/7	4/5
10/11	7/7	5/8	5/7	2/3	5/5

Table 8: *Words correct on randomly selected sentence for German (top) and Hindi (bottom). Sentences 1-20 are in-domain; 21-24 out of domain.*

6. Conclusions

By integrating ASR, TTS, and lexicon building into a single, simplified, web-based development framework, the aim of SPICE is to make speech technology available to developers that are not expert in language technology. Admittedly, the students participating in this lab do not fit the bill of *naive* users – our ultimate target audience. All are graduate students in the Languages Technology Institute and, due to the course credits on offer, were not just technically proficient but *motivated*. Their experience and observations has helped us identify deficiencies that need to be addressed before the software can reliably be employed by less sophisticated users – those that "just know their language."

For the task of voice building, more data-validity checks need to be incorporated. So that, for example, the user does not reach the end of a failed voice-building attempt only to discover that the phoneme set, or character definition, or lexicon is in some ways deficient. In a similar vein: faced with the sizable task of providing pronunciations for thousands of words, our users have requested that they only be presented with the essential fraction, i.e. only words that the system is unsure about.

This raises a deep and challenging question: can the system be sufficiently "self-aware" that it knows when it needs more information, and when it can stop? At a practical level, we'd like the system to know when it has an amount of speech sufficient for building a good quality voice. Possibly we can resort to proxy measures of quality, such as mean cepstral distortion and average prediction error of prosodic models. Additionally, determining a suitable stopping point may involve iterative cycles of feedback from the user to perform transcription tests and point out misspoken words. These remain open issues.

7. Acknowledgments

We thank Stephan Vogel for providing software and expertise in statistical machine translation. We also thank the following people for active participation: Nguyen Bach, Sameer Badaskar, Matthias Bracht, Joerg Brun, Roger Hsiao, Thuy Linh Nguyen, Kemal Oflazar, Sharath Rao, Emilian Stoimenov, and Andreas Zollmann. Extra thanks go to Joerg Brun and Sameer Badaskar for transcribing test sentences. Sameer Badaskar also wrote the embedded recorder used for collecting speech. Matthew Hornyak kept all the various software in running order and deserves our undying gratitude.

This work is in part supported by the US National Science Foundation under grant number 0415021 "SPICE: Speech Processing Interactive Creation and Evaluation Toolkit for new Languages." Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. References

[1] Schultz T. and Kirchhoff, K. (Eds.), *Multilingual Speech Processing,* Academic Press, 2006.

[2] Schultz T, Westphal M, Waibel A, *The Global Phone Project: Multilingual LVCSR with Janus-3*, 2nd SQEL Workshop, Plzen, Czech Republic, 1997.

[3] Schultz, T., *GlobalPhone: A Multilingual Speech and Text Database developed at Karlsruhe University.* ICSLP, Denver, USA, 2002.

[4] Clarkson P, and Rosenfeld R. *Statistical Language Modeling Using the CMU-Cambridge Toolkit,* Proceedings of ESCA, Eurospeech 1997.

[5] Paul Taylor, Alan Black, and Richard Caley, *The architecture of the Festival speech synthesis system,* Procedings of the Third ESCA Workshop in Speech Synthesis, Jenolan Caves, Australia, 1998, pp. 147–151.

[6] Black, A., and Lenzo, K., *The FestVox Project: Building Synthetic Voices,* http://festvox.org/bsv, 2000.

[7] John Kominek, Alan W Black, *Learning Pronunciation Dictionaries: Language Complexity and Word Selection Strategies,* Proceedings of the Human Language Technology Conference of the NAACL, pp. 232-239, New York City, USA.

[8] The Lemur Toolkit. www.lemurproject.org.

[9] Stephan Vogel, Ying Zhang, Alician Tribble, Fei Huang, Ashish Venugopal, Bing Zhao, Alex Waibel, *The CMU Statistical Machine Translation System*, Proceedings of the MT Summit IX, New Orleans, USA, Sept. 2003.

[10] 11-733/735 Multilingual Speech-to-Speech Translation Seminar/Lab, http://penance.is.cs.cmu.edu/11-733.

[11] Tanja Schultz, Alan W Black, Sameer Badaskar, Matthew Harnyak, John Kominek, SPICE: Web-based Tools for Rapid Language Adaptation in Speech Processing Systesm, InterSpeech 2007, Antwerp, Belguim.

[12] John Kominek, Alan W Black, *CMU Arctic Speech Database, Speech Synthesis Workshop 5, Pittsburgh, USA, 2004.*

[13] Alan W Black, *CLUSTERGEN: a statistical parametric synthesizer using trajectory modeling,* InterSpeech 2006, Pittsburgh, USA, September 2006.

[14] Heiga Zen, Keiichi Tokuda, Tadashi Kitamura, *An introduction of trajectory model into HMM-based speech synthesis*, Speech Synthesis Workshop 5, Pittsburgh, USA, 2004.

[15] Davel, M. and Barnard, E. *Efficient generation of pronunciation dictionaries: machine learning factors during bootstrapping*, ICSLP2004, Jeju, Korea.

[16] CMUDICT, www.speech.cs.cmu.edu/cgi-bin/cmudict.

[17] Hindi EMILLE Lancaster Corpus, www.elda.org/catague /en/text/

SVM Based Feature Extraction in Speech Synthesis

Peter Cahill, Jan Macek, Julie Carson-Berndsen

School of Computer Science and Informatics, University College Dublin, Dublin, Ireland

peter.cahill@ucd.ie, jan.macek@ucd.ie, julie.berndsen@ucd.ie

Abstract

Annotations of speech recordings are a fundamental part of any unit selection speech synthesiser. However, obtaining flawless annotations is an almost impossible task. Manual techniques can achieve the most accurate annotations, provided that enough time is available to analyse every phone individually. Automatic annotation techniques are a lot faster than manual, doing the task in a much more reasonable time frame, but such annotations contain a considerable amount of error. In this paper a technique is introduced that can quite accurately ensure a degree of articulatory-acoustic similarity between annotated units. The synthesiser will encourage the use of units that have been identified to have appropriate articulatory-acoustic parameters, but will not limit the domain of the speech database. This helps to identify where joins can be performed best and also identifies which annotations should be avoided at the phone level.

1. Introduction

Unit selection speech synthesis can produce natural sounding intelligible speech. This is particularly true if the domain of the speech being synthesised is the same as the domain of the recorded speech database being used. It is when out-of-domain words or phrases are used that the quality and naturalness reduces significantly. The reduction of quality is often due to relying on small speech units to construct the target speech. Modern synthesis algorithms are capable of constructing human sounding words from small speech units, however the quality of constructing natural sounding units from smaller units is dependent on the available speech annotations.

As the primary dependency of any unit selection speech synthesiser is the speech database in use, having accurate annotations of the speech data is crucial. Annotations of speech data for synthesis are commonly available in a phone or diphone format. The accuracy of such annotations is variable, and very dependent on the how the annotations were created. Experiments on different annotation techniques show that there will always be quite a significant degree of error [1]. The minimal degree of error is obtained when a detailed manual correction is performed on automatically segmented data. In [1] manual correction to obtain minimal error is described to take approximately 2 minutes for a first pass and 30 minutes for the second pass of correction per utterance. Annotating a full speech database this way (which may be from 1000-10000 utterances) is very resource intensive. It is also quite likely that when doing such a tedious annotation that after a few utterances the segmentors will loose focus and start to annotate less accurately. It is clear from [1] that no matter which annotation technique is used, there will always be a significant amount of error introduced into the synthesis system.

In this paper we introduce an automated language independent technique that identifies inaccurately annotated phones so that the synthesiser can avoid them at a later stage. Parts of this concept have been previously described in earlier work [2]. In this paper, the concept is developed further and a comparison is performed with the technique described in [2].

The remainder of this paper is structured as follows: Section 2 describes the concepts of how the use of an articulatory-acoustic analysis may assist in the selection of more suitable units for synthesis. Section 3 describes some potential applications of this work as well as the primary application, unit selection speech synthesis. Section 4 contains an analysis of how voices built using the techniques described in this paper compare with our previous work on this topic. Section 5 concludes and describes future work.

2. Hypothesis

Annotations of the speech data in unit selection speech synthesis are commonly in a phonological format, consisting of a unit label (perhaps phoneme or diphone) as well as the temporal endpoints for that particular unit. The unit labels are often estimated automatically by using a grapheme to phoneme or word to phoneme technique on the orthographic transcriptions of the speech database. Such annotations work quite well and are the foundation of many modern synthesis systems. The problem with using this type of annotation is that the unit label is estimated from the orthographic transcriptions. For this reason, if two speech databases were recorded using the same orthographic transcriptions, all that would differ in the resulting annotations would be the temporal endpoints, but it is unlikely that the two speakers would have articulated every basic unit in the database identically.

The use of an independent, second level of annotations is presented in this paper. The second level of annotations is a phonetic analysis of the audio data in the speech database. This form of annotation is intended to be used in a cross validation technique with the more common phonological form of annotation. The aim of this is to improve the consistency of the annotations as well as to automatically identify misaligned, mislabelled or mispronounced units in the speech database.

The phonetic analysis is an analysis of the actual speech data in the database. This is done to give an alternative perspective on the speech data, as the phonological annotations are derived from the orthographic transcriptions only. Some phonetic data has been used previously in speech synthesis, although it was typically used at the joining of units rather than the annotation stage. In cases where phonetic analysis has been used for annotation previously, it was mostly using spectral parameters such as Mel-frequency Cepstral Coefficients (MFCCs), F0 and power in techniques such as [3]. This work focuses on using articulatory-acoustic features rather than the spectral parameters.

As each articulatory-acoustic feature extractor is essentially checking for a group of spectral properties, the set of feature extractors used will ensure the presence or absence of a number of acoustic properties. When the feature extractors are used, the decision as to whether or not a phone should be marked as preferred is essentially a check on how acoustically similar the phone is to other units with the same phonological label. Units that contain the appropriate acoustic properties are marked as preferred units. During diphone synthesis, the point at which the diphone boundary lies is the same as the point where the articulatory-acoustic feature comparison is performed. This technique results in the end of a diphone having the same articulatory-acoustic features present as the start of the following diphone. Therefore, many of the potentially bad joins are removed before the distance measure or join cost between units needs to be evaluated.

Situations where the feature extractors used are not 100% accurate still perform well, as long as the feature extractor is performing any acoustic analysis. This concept can be extended further than just feature extraction as well, any form of acoustic or spectral analysis can be quite useful in this technique.

This technique has also proved useful for optimisation of unit selection speech synthesis as it can be used to reduce the number of potential units used in the Viterbi search, and can result in a much faster synthesis depending on how many of the phones have been identified as preferred. Some informal tests showed a speed increase of approximately 300%, at the same time as an improvement in output quality. The speed increase is due solely to the use of preferred phones reducing the range of units in the Viterbi search.

2.1. Articulatory-Acoustic Feature Extraction

2.1.1. Articulatory-Acoustic Features

The articulatory-acoustic features were introduced in the speech recognition community as an alternative extension to the acoustic analysis of a speech signal. They help to improve robustness of speech recognition systems used in various uncontrolled environments where performance of traditional speech recognition systems degrades rapidly [4].

Articulatory-acoustic features are thought to be a good compromise in the description of a speech signal, offering a more detailed description of the acoustic signal than phonemes, yet still providing a linguistically interpretable symbolic annotation. Acoustic correlates of features are described in [5].

Machine learning techniques were used to detect the presence or absence of articulatory-acoustic features.

2.1.2. Support Vector Machines

Support Vector Machines (SVMs) learn separating hyperplanes to classify instances in the feature space that are mapped from the input space of the classified data. The mapping from input space to feature space is performed with the application of a kernel on the feature space. The dimension of the feature space is typically much higher than that of the original input space. [6] provides a thorough mathematical background.

The motivation for SVMs comes from the pattern recognition community with mathematical properties of linear classifiers and from the statistical learning theory community with the structural risk minimisation properties of SVMs [7, 8].

For the training of the SVM feature extraction models the TIMIT corpus of read speech was used. 52 values were extracted for every frame of the speech signal, these values were used as inputs for the SVM classifiers. From each frame 12 MFCCs were extracted together with first and second order differences, frequencies of formants (F1-F5) with first order differences, bandwidths of detected formants, and fundamental frequency. The length of the speech signal frames was set to 25 ms and step between two adjacent frames to 10 ms. The original speech signal was sampled at 16 kHz. The distributions of classes vary significantly for different types of features. While the distribution of classes is almost equal (the case of the *vocalic* feature) for half of the articulatory features, in the rest of the cases the positive classes are rare in the data. This has a strong influence on the recall of the positive classes while the overall accuracy remains high.

The training of the SVMs was performed only on one dialect region of the TIMIT corpus (namely dialect region no. 3) mainly for the reasons of time complexity of the training. It has been shown in [9] that the SVMs with second-order polynomial kernels give best performance for the task of articulatory feature recognition and this setting was used throughout the experiments reported in this article.

2.1.3. Performance of SVMs in context

In previous work it has been shown that SVMs outperform other classifiers at the task of articulatory feature recognition [9]. Namely, it gives superior results over hidden Markov models (HMMs), a fact that is analysed in more detail below.

The main distinction between the two approaches, that of SVMs and that of HMMs, is that the former treats the stream of speech signal frames as independent frames and the latter treats them as adjacently dependent. The dependency of adjacent frames is employed in current state-of-the-art speech recognition systems and it is exploited on the phone level. At this level the HMMs model a much larger set of events in the speech signal as opposed to the binary set of feature presence/absence in the case of recognition of articulatory features. This limits the possibility of constructing reliable articulatory feature models with HMMs as the dependency between adjacent frames can not be utilised in the case of binary classification.

In the approach taken with SVMs, only information from the processed speech frame is used. As this might be limiting in some tasks, in the case of articulatory feature recognition it is more detrimental to the performance when the model tries to capture non-existent dependencies in the data as in the HMM based approach. The performances of both of the methods are compared in Section 4.

2.2. Phonetic annotations

Previous work [2] investigated the use of articulatory-acoustic feature extraction in speech synthesis. In this article the idea has been developed further. We investigate the use of SVM based feature extraction as well as the HMM based feature extraction previously used.

Features are extracted on all audio data in the speech database. The features present at the diphone joining point of each phone are examined. For each phone label, the features present are compared with all other phones with the same label. The aim is to identify most common set of features present and absent at the diphone joining point for each phone. The comparison of the most common set of features present is explicit.

When the phones that have the most common features present at their diphone joining point are identified, they are then considered to be preferred phones. This is done so that at a later stage, when the synthesiser is performing the synthesis, it can try to use as many preferred phones as possible.

Furthermore, the concept of using both SVM and HMM for a speech synthesis database is introduced. Features are extracted using both SVM and HMM models for a speech database. The types of features being identified are the same for both the SVM extractors and the HMM extractors. The SVM and HMM models are used independently to improve the robustness of the technique.

The preferred phones identified by the HMMs are organised into sets, resulting in the set of all preferred phones of label p being identified as A$_p$. Similarly the set of preferred phones identified by the SVM based feature extractors is identified as B$_p$. The aim is to find the set C$_p$ of phones that have been identified as preferred by both A$_p$ and B$_p$, such that

$$C_p = A_p \cap B_p \tag{1}$$

where C$_p$ is the intersection of A$_p$ and B$_p$. The elements of C$_p$ are then given the highest phone priority possible.

The use of the set C$_p$ is essentially a cross validation on the preferred phones. This is performed in an attempt to identify the most acoustically similar units. The use of both SVMs and HMMs is an attempt to maximise consistency by verifying the preferred phones.

We expect that the set of units C$_p$ is best used in larger speech databases, as in a relatively small speech database too many units may be pruned from the speech database. Although it will also be significantly dependent on how many articulatory-acoustic feature types are being extracted. Identifying the thresholds of optimal database size for this technique has yet to be identified. Current work suggests it varies significantly between speech databases; furthermore, the phone set in use will also have considerable influence on this.

3. Applications

The technique described was designed for use in a unit selection diphone synthesiser. The synthesiser uses a Viterbi search on all possible sequences of appropriate diphones to identify the ideal path of units. The synthesiser uses the Mahalanobis distance measure to estimate join cost between the two vectors ($\vec{x}$,$\vec{y}$), such that:

$$D(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})} \tag{2}$$

where Σ is the covariance matrix, and $\vec{x}$ and $\vec{y}$ are the vectors at the diphone joining point of different phones. The vectors contain Mel-frequency cepstral coefficients (MFCCs), F$_0$ and power. For further information on using this distance measure see [10, 11].

The comparison of the phonetic annotations, which are the articulatory-acoustic features in the case of this study, is always done at the diphone boundaries of phones. This ensures that when two units are being considered for a join by the Mahalanobis distance measure, both units have previously been classified as acoustically similar units at that exact join point.

Voice data for the synthesiser was compiled into a single file. Building of the voice was a fully automatic process, where the orthographic transcriptions and the speech recordings were input, all necessary processing was performed on the speech data and the result was output as a single voice file. Annotations in the voice data are stored in Unicode, using IPA characters when labelling phones.

The speech annotations were used in a phonological hierarchy, quite similar to the phonological structure matching technique described in [12].

It is quite likely that this technique could also be useful for other applications than intended. Any system that uses a database of speech recordings may find the phonetic annotations useful. We also expect that this technique may be useful for measuring how good or consistent the phonological annotations for a set of speech recordings are.

4. Testing and Results

The speech database used for testing was the full ATR Blizzard Challenge 2007 speech data. The speech recordings had phonological hierarchies automatically generated using C4.5 decision trees ([13]) trained from the CMUDICT dictionary. The CMUDICT phone set was used with the phoneme labels translated into their IPA equivalents.

Three voices were built from the data. For all of the voices built, all of the data except the extracted features was identical. The first voice, V$_{hmm}$ was the voice using the features extracted by the HMMs to identify the preferred phones. The second voice, V$_{svm}$ was the voice data using the features extracted by the SVM models to identify the preferred phones. The remaining voice, V$_{hybrid}$ was the same voice data as V$_{hmm}$ and V$_{svm}$, but the set of preferred phones was set to be the intersection of the preferred phones of V$_{hmm}$ and V$_{svm}$.

The set of feature extractors used for both SVMs and HMMs was: *anterior, consonantal, nasal, vocalic, and voiced.* This set of feature extractors was chosen as tests currently indicate that these are our best performing models. A set of five feature extractors seemed reasonable as in the case of this work they are being used to identify articulatory-acoustic coherent sounds, rather than to identify the phone labels exactly. The respective accuracies of the classifiers on the frame-level for each of the used features are presented in Table 1.

Feature	HMM Accuracy (in %)	SVM Accuracy (in %)
anterior	74.82	91.01
consonantal	82.65	89.10
nasal	89.65	97.92
vocalic	82.16	93.12
voiced	84.93	93.59

Table 1: *Performance of classifiers on articulatory-acoustic features.*

Of the speech data used, every word aligned in 6559 of the utterances. There was an average of 8756 phones for each phoneme label. This was dispersed over a wide range, where vowels would typically have the most phones - the most for any phone label was 33863 for the [ɪ] phone. The phone with the lowest occurrence was the [ʒ] phone with 159 occurrences.

In the test results, all sets of phones with an A$_p$ label refers to a set of preferred phones identified by the V$_{hmm}$ voice, the B$_p$ label refers to a set of preferred phones identified by the V$_{svm}$ voice, and C$_p$ is the intersecting set as described in Section 2.2. Tables on the data compare the sets that the phones are in. The first column shows the percentage of A$_p$ that is in the C$_p$ set, the second columns shows the same data for B$_p$, and the third column shows a percentage of the quantity of phones in the C$_p$ set to the amount of phones in the full set for that phone label. The data can also be interpreted that the percentages in the first column show the percentage of preferred phone agreement from the HMMs with the SVMs, and the second column shows the percentage of preferred phone agreement for the SVMs with the HMMs. In most cases most of the SVM preferred phones are a subset of the HMM preferred phones, but not visa versa.

Phone	$A_p \cap C_p$ %	$B_p \cap C_p$ %	$C_p \cap \text{Full}_p$ %
b	46.01	33.43	16.08
d	55.45	49.43	22.73
g	43.59	42.81	18.09
k	39.02	91.08	31.83
p	31.48	83.50	26.72
t	41.91	70.50	29.75
j	46.01	81.64	31.89

Table 2: *Phones with stops.*

Table 2 illustrates the performance of the feature extraction technique for phones with stops. Phones with stops have the lowest amount of agreement between the two techniques used, resulting in the smallest C_p sets, where the [b] phone has only 16.08% of phones of all phones labelled [b] in the C_b set. This is the lowest percentage of any phone in the C_p sets. The fact that phones with stops have the smallest C_p sets is expected. This is due to the dynamic nature of stops, where the transition leading to occlusion is quite variable. As the focus of this application of features is only on the diphone joining points of phones, perhaps the stops would get larger C_p sets if the diphone points of speech with the presence of stops allowed for temporal variance. Since the current technique only looks at a single point in each phone the acoustics of stop sounds are only measured at a single point, not allowing for the acoustic variance that occurs at stops, resulting in the low amount of articulatory acoustic consistency indicated by the size of the C_p sets. It is also possible that the forced alignment technique (or perhaps the acoustic model used) does not work very well with stops. If this was the case it would result in an increased amount of variance between the diphone boundaries. In cases such as this the synthesiser will not prioritise preferred phones when the percentage of phones in a C_p set is this low, resulting in it encouraging a minimal amount of joins to occur at a stop.

Phone	$A_p \cap C_p$ %	$B_p \cap C_p$ %	$C_p \cap \text{Full}_p$ %
ʧ	79.67	99.09	78.08
f	36.39	84.51	31.53
h	36.81	48.45	17.35
ʤ	59.66	95.43	56.07
s	79.16	99.39	78.35
ʃ	92.56	99.46	91.68
θ	47.22	95.80	42.75
v	68.90	60.76	39.24
z	65.56	94.94	62.53
ʒ	76.43	98.36	75.47

Table 3: *Phones with frication.*

Table 3 shows the results for fricative sounds. In almost every case the results in Table 3 are higher than the results in Table 2. It is clear that the intersection set, C_p has much more phone coverage than in the case of the stops. The C set for the [ʃ] phone had the highest percentage of any C_p set. Of the data in Table 3, the lowest set C scores were for C_f and C_h. The score for both of these is reasonable as the pronunciation of a [f] or [h] sound in real speech is very dependent on the following phoneme. The [h] phone scores far lower than any other fricative phone, it obviously has some significantly different property than the other fricative sounds, hence its relatively low C_p score. We expect this to be due to it being glottal, which is the unique factor that distinguishes the [h] phone from the other voiceless fricatives described in the table. Additionally, glottal sounds are underrepresented in the training data and are typically of low energy which makes the task of distinguishing them more difficult.

Phone	$A_p \cap C_p$ %	$B_p \cap C_p$ %	$C_p \cap \text{Full}_p$ %
b	46.01	33.43	16.08
h	36.81	48.45	17.35
ɡ	43.59	42.81	18.09
ɪ	29.43	55.40	19.90

Table 4: *Lowest four scoring phones in terms of C_p.*

Table 4 shows the four lowest scoring phones in respect of C_p. Two of these phones also occurred in Table 2, and one of them was in Table 3. In cases where the synthesiser comes across phones that have such a low score for the C_p set, it will use the full set of that phone rather than C_p. The [ɪ] phone got the lowest score for a vowel. This is as expected in this case, and is due to the dictionary using the [ɪ] phone too frequently when other vowel sounds would be more appropriate. This is also the reason for the [ɪ] phone to be the most common in the speech data.

Although the use of the C_p sets of phones instead of the full set of phones prunes the database considerably, in the case of a speech database similar in size to the one used in this article it still leaves a significant portion of data for synthesis. Of the 8 hour corpus used, the average size of a C_p set was 38.33% of the full set of phones, resulting in 3 hours and 4 minutes of the speech data remaining in the C_p sets. This is still a very large amount of data for the synthesiser to use in comparison with the commonly used 1 hour ARCTIC corpus as used in [14]. If a threshold is used for the synthesiser to use the full set of phones in the case of the percentage of C_p in the full set being below the defined threshold, the duration of the preferred data would increase significantly.

The patterns in the performances of the classifiers can be summarised in three types of behaviour. In the first type, the outputs from the SVMs were a clear subset of the outputs from the HMMs, regardless of the performance of the HMMs. This could be attributed to generally high recall with lower precision of the HMMs and nearly even precision and recall of the SVMs in the task of recognition of articulatory-acoustic features. This type of observation counts for 16 cases of the total of 36 phones used, where the presence of preferred phones calculated from the SVMs is higher than 90% of the set of phones preferred by the HMMs.

The second type of behaviour observes a low percentage of preferred phones being the result of agreement between the two phone classification approaches. A percentage is considered to be ‘low’ when it comes below 20%. In this case either the agreement of assignments to the class preferred is low between the two classifiers or the agreement is high for one of the classifiers but the percentage of phones preferred by the other of all phones is low.

The third type of behaviour shows a higher agreement of the HMMs with the assignments for the preferred phones given by the SVMs. The level of agreement is much lower in this case for HMMs than it is in the first type of behaviour for SVMs (<65% vs. >90%).

From a general perspective, the SVM based phone selection is more conservative than the HMM based one as it results in more non-preferred phones in 30 of the 36 cases.

When using smaller speech databases is it reasonable to reduce the set of features being used to increase the sizes of the C_p sets. Even using one or two feature extractors will still help to prune out many acoustic mismatches that would have otherwise resulted in a bad join, even when using the Mahalanobis distance measure.

5. Conclusion

A technique was introduced that identifies the most acoustically similar units in the speech database. The acoustically similar units are intended for use in diphone synthesis, as the synthesiser is aware of the acoustic properties of the start and end points of units. This allows for the synthesiser to ensure a high degree of acoustic consistency during joins, as well as the Mahalanobis distance measure is still used to measure join cost. In experiments to date this technique has only been used at diphone joining points and if the synthesiser is synthesising a word that does exist in the voice data as a complete word, it will select the word regardless of the acoustic properties at the diphone points in the unit, resulting in the domain of the speech data not being affected.

This paper develops further previous work on this topic [2], and now the technique is more robust by using both SVM and HMM models to analyse the acoustic properties of the speech data. A cross validation technique does result in identifying the set of the most acoustically similar units to be used during synthesis.

An analysis of how the HMM and SVM models performed is described. Experiments were done on the 8 hour ATR Blizzard Challenge 2007 speech data. The use of the technique described identified the most acoustically similar units, approximately 3 hours of the speech data. In many cases the set of preferred phones identified by the SVM models used would be almost a complete subset of the set of preferred phones identified by the HMM models used. Results show that phones containing stops had the least amount of acoustic consistency at their diphone boundaries, and phones containing frication had the most acoustic consistency. Vowel sounds had a quite variable amount of consistency, but was always between that of the stops and the fricatives. It is expected that the variation of the vowels' consistency to be due to the dictionary used to train the grapheme to phoneme technique used, and the pronunciation of vowels being more irregular than consonants in real speech.

In future work we intend to further develop this concept. Current results are very encouraging, and the net result is a fully automatic, language independent technique to obtain an additional, reliable perspective on the content of the voice data. The most acoustically similar 3 hours of speech data in the test data was identified, and the system is aware of which phones have the most acoustic irregularities—allowing the synthesiser to weigh joins at such phones to encourage more joins at phones that are more acoustically consistent.

6. Acknowledgements

This work was funded by the Irish Research Council for Science, Engineering and Technology (IRCSET), IBM, and SFI. The experiments were done using the ATR Blizzard Challenge 2007 speech database. This material is based upon works supported by the Science Foundation Ireland for the support under Grant No. 02/IN1/I100. The opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Science Foundation Ireland. The authors express their thanks to ATR for access to their speech database. The authors also express thanks to Supphanat Kanokphara for training some of the feature extractors.

7. References

[1] J. Kominek, C. Bennett, and A. Black, "Evaluating and Correcting Phoneme Segmentation for Unit Selection Synthesis," *Proceedings of Eurospeech*, pp. 313–316, 2003.

[2] P. Cahill, D. Aioanei, and J. Carson-Berndsen, "Articulatory Acoustic Feature Applications in Speech Synthesis," *submitted to Interspeech 2007*, 2007.

[3] A. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," *Proc. Eurospeech*, vol. 2, pp. 601–604, 1997.

[4] J. Carson-Berndsen, *Time Map Phonology: Finite State Models and Event Logics in Speech Recognition*. Kluwer, 1998.

[5] K. Stevens, *Acoustic Phonetics*. Cambridge, MA, USA: MIT Press, 1999.

[6] B. Schölkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA, USA: MIT Press, 2002.

[7] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer Verlag, 1995.

[8] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[9] J. Macek and J. Carson-Berndsen, "Articulatory manner features recognition with linear and polynomial kernels," in *Fifth Slovenian and First International Language Technologies Conference*, (Ljubljana, Slovenia), Oct. 2006.

[10] A. Gray Jr and J. Markel, "Distance measures for speech processing," *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, vol. 24, no. 5, pp. 380–391, 1976.

[11] R. Donovan, "A new distance measure for costing spectral discontinuities in concatenative speech synthesisers," *The 4th ISCA Tutorial and Research Workshop on Speech Synthesis*, 2001.

[12] P. Taylor, "Concept-to-speech synthesis by phonological structure matching," *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 358, no. 1769, pp. 1403–1417, 2000.

[13] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[14] A. Black and K. Tokuda, "The Blizzard Challenge–2005: Evaluating corpus-based speech synthesis on common datasets," *Proc. of Interspeech 2005*, pp. 77–80, 2005.

Spectral Conversion Based on Statistical Models Including Time-Sequence Matching

Yoshihiko Nankaku†*, Kenichi Nakamura*†*, Tomoki Toda*‡ *and Keiichi Tokuda*†

† Nagoya Institute of Technology,
Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555 Japan
‡Nara Institute of Science and Technology,
8916-5 Takayama, Ikoma, Nara, 630-0101 Japan
{nankaku, k-n, tokuda}@sp.nitech.ac.jp, tomoki@is.naist.jp

Abstract

This paper proposes a spectral conversion technique based on a new statistical model which includes time-sequence matching. In conventional GMM-based approaches, the Dynamic Programming (DP) matching between source and target feature sequences is performed prior to the training of GMMs. Although a similarity measure of two frames, e.g., the Euclid distance is typically adopted, this might be inappropriate for converting the spectral features. The likelihood function of the proposed model can directly deal with two different length sequences, in which a frame alignment of source and target feature sequences is represented by discrete hidden variables. In the proposed algorithm, the maximum likelihood criterion is consistently applied to the training of model parameters, sequence matching and spectral conversion. In the subjective preference test, the proposed method is superior than the conventional GMM-based method.

1. Introduction

In recent years, voice conversion especially the statistical model based approaches are widely investigated. This technique can modify speech characteristics using conversion rules statistically extracted from a small amount of training data. As a typical spectral conversion method, a mapping algorithm based on the Gaussian Mixture Model (GMM) has been proposed [1]. In this method, the mapping between spectral features of the source and target is determined based on GMMs. In each mixture component, the conditional mean vector of target features given source features is calculated as a simple linear transformation using the covariance matrix of the concatenated feature vector. The converted vector is defined as the weighted sum of the conditional mean vectors, and the conditional occupancy probabilities of mixture components are used as weights. A more accurate formulation of spectral conversion based on ML (Maximum Likelihood) criterion has been presented [2]. The ML-based conversion is a sophisticated technique because all processes in the algorithm is derived based on the single objective function.

In these GMM-based method, GMMs are trained using joint feature vectors which are references of mapping rules, and the DP matching between feature sequences of source and target are conducted prior to the training of GMMs. Typically the similarity measure of two frames is adopted independently of the training of GMMs, e.g. Euclid distance. However, this might be inappropriate for converting the spectral features. To avoid this problem, we propose a voice conversion technique based on a new statistical including temporal matching between source and target feature sequences. The likelihood function can directly deal with two different length sequences, in which a frame alignment between two sequences is represented by discrete hidden variables. In the proposed voice conversion technique, the ML criterion is consistently applied to the training of model parameters, sequence matching and spectral conversion.

The parameters of the proposed model can be estimated via the expectation maximization (EM) algorithm for approximating the Maximum Likelihood (ML) estimate. However, a complex model structure leads to an exponential increase in the amount of computation for its training algorithm and the exact expectation step is computationally intractable. To derive a feasible algorithm, we applied the variational EM algorithm [4], [5]. Variational methods approximate the posterior distribution over the hidden variables by a tractable distribution. A structure approximation is presented in which the hidden variables of GMMs and the temporal matching are decoupled. However, the convergence point of the EM algorithm depends on the initial model parameters. Moreover, in the variational EM algorithm for the proposed model, the decoupled posterior distributions are updated individually based on the other distributions which are unreliable at an early stage of training. To overcome these problems, we applied the deterministic annealing EM (DAEM) algorithm [6] to the variational algorithm for the proposed model.

The paper is organized as follows. Section 2 explains the conventional voice conversion technique based on GMMs. Section 3 describes a new statistical model including temporal matching, and section 4 explains its training algorithm. Voice conversion based on the proposed model is presented in section 5 and experimental results are reported in Section 6. Finally, conclusions and future works are given in Section 7.

2. GMM-based Spectral Conversion

To convert spectral feature sequences of a source speaker to that of a target speaker, the joint probability density of two features are modeled by GMM [2]. Let a vector $\boldsymbol{O}_t = \left[\boldsymbol{O}_t^{(1)\top}, \boldsymbol{O}_t^{(2)\top}\right]^\top$ be a joint feature vector of the source one $\boldsymbol{O}_t^{(1)}$ and the target one $\boldsymbol{O}_t^{(2)}$ at time t. An alignment between two feature sequences is obtained by the Dynamic Programming (DP) matching. In the GMM-based voice conversion, the vector sequence $\boldsymbol{O} = \left[\boldsymbol{O}_1{}^\top, \ldots, \boldsymbol{O}_t{}^\top, \ldots, \boldsymbol{O}_T{}^\top\right]^\top$ is modeled by GMM to learn a relation between source and target features. The output probability of $\boldsymbol{O}$ given GMM Λ can be written as follows:

$$P(\boldsymbol{O} \mid \Lambda) = \prod_{t=1}^{T} \sum_{i=1}^{M} w_i \mathcal{N}(\boldsymbol{O}_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{1}$$

where

$$\boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_i^{(1)} \\ \boldsymbol{\mu}_i^{(2)} \end{bmatrix}, \boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_i^{(1,1)} & \boldsymbol{\Sigma}_i^{(1,2)} \\ \boldsymbol{\Sigma}_i^{(2,1)} & \boldsymbol{\Sigma}_i^{(2,2)} \end{bmatrix}. \tag{2}$$

and M means the number of mixtures, $w_i = P(i \mid \Lambda)$ is the mixture weight of the i-th component, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the mean vector and covariance matrix, respectively. These model parameters are estimated via the Expectation Maximization (EM) algorithm.

2.1. Maximum likelihood spectral conversion

In the maximum likelihood spectral conversion, the optimal converted feature sequence $\boldsymbol{O}^{(2)} = \left[\boldsymbol{O}_1^{(2)\top}, \ldots, \boldsymbol{O}_t^{(2)\top}, \ldots, \boldsymbol{O}_T^{(2)\top}\right]^\top$ given a source feature sequence $\boldsymbol{O}^{(1)} = \left[\boldsymbol{O}_1^{(1)\top}, \ldots, \boldsymbol{O}_t^{(1)\top}, \ldots, \boldsymbol{O}_T^{(1)\top}\right]^\top$ is obtained by maximizing the following conditional distribution:

$$\begin{aligned} &P(\boldsymbol{O}^{(2)} \mid \boldsymbol{O}^{(1)}, \Lambda) \\ &= \sum_{\boldsymbol{m}} \left[P(\boldsymbol{m} \mid \boldsymbol{O}^{(1)}, \Lambda) \prod_{t=1}^{T} P(\boldsymbol{O}_t^{(2)} \mid \boldsymbol{O}_t^{(1)}, m_t, \Lambda) \right] \end{aligned} \tag{3}$$

where $\boldsymbol{m} = (m_1, m_2, \ldots, m_T)$ is a mixture number sequence. The conditional distribution can also be written as GMM, and its output probability distribution is presented as follows:

$$P(\boldsymbol{O}_t^{(2)} \mid \boldsymbol{O}_t^{(1)}, m_t = i, \Lambda) = \mathcal{N}\left(\boldsymbol{O}_t^{(2)}; \boldsymbol{E}_i(t), \boldsymbol{D}_i\right) \tag{4}$$

where

$$\begin{aligned} \boldsymbol{E}_i(t) &= \boldsymbol{\mu}_i^{(2)} + \boldsymbol{\Sigma}_i^{(2,1)} {\boldsymbol{\Sigma}_i^{(1,1)}}^{-1} \left(\boldsymbol{O}_t^{(1)} - \boldsymbol{\mu}_i^{(1)}\right) && (5) \\ \boldsymbol{D}_i &= \boldsymbol{\Sigma}_i^{(2,2)} - \boldsymbol{\Sigma}_i^{(2,1)} {\boldsymbol{\Sigma}_i^{(1,1)}}^{-1} \boldsymbol{\Sigma}_i^{(1,2)} && (6) \end{aligned}$$

Since the equation (3) includes latent variables, the optimal sequence of $\boldsymbol{O}^{(2)}$ is estimated via the EM algorithm. The EM algorithm is an iterative method for approximating the maximum likelihood estimation. It maximizes the expectation of the complete data log-likelihood so called $\mathcal{Q}$-function (auxiliary function):

$$\begin{aligned} \mathcal{Q}(\boldsymbol{O}^{(2)}, \hat{\boldsymbol{O}}^{(2)}) &= \sum_{all\ \boldsymbol{m}} \Big[P(\boldsymbol{O}^{(2)}, \boldsymbol{m} \mid \boldsymbol{O}^{(1)}, \Lambda) \\ &\qquad \times \ln P(\hat{\boldsymbol{O}}^{(2)}, \boldsymbol{m} \mid \boldsymbol{O}^{(1)}, \Lambda) \Big] \end{aligned} \tag{7}$$

Taking the derivative of the $\mathcal{Q}$-function, the spectral sequence $\hat{\boldsymbol{O}}^{(2)}$ which maximizes the $\mathcal{Q}$-function is given by

$$\hat{\boldsymbol{O}}^{(2)} = \left(\overline{\boldsymbol{D}^{-1}}\right)^{-1} \overline{\boldsymbol{D}^{-1}\boldsymbol{E}} \tag{8}$$

where

$$\begin{aligned} \overline{\boldsymbol{D}^{-1}} &= \text{diag}\left[\overline{\boldsymbol{D}_1^{-1}}, \overline{\boldsymbol{D}_2^{-1}}, \cdots, \overline{\boldsymbol{D}_T^{-1}}\right] && (9) \\ \overline{\boldsymbol{D}_t^{-1}} &= \sum_{i=1}^{M} \gamma_i(t) \boldsymbol{D}_i^{-1} && (10) \\ \overline{\boldsymbol{D}^{-1}\boldsymbol{E}} &= \left[\overline{\boldsymbol{D}^{-1}\boldsymbol{E}_1}^\top, \overline{\boldsymbol{D}^{-1}\boldsymbol{E}_2}^\top, \cdots, \overline{\boldsymbol{D}^{-1}\boldsymbol{E}_T}^\top\right]^\top && (11) \\ \overline{\boldsymbol{D}^{-1}\boldsymbol{E}_t} &= \sum_{i=1}^{M} \gamma_i(t) \boldsymbol{D}_i^{-1} \boldsymbol{E}_i(t) && (12) \\ \gamma_i(t) &= p(m_t = i \mid \boldsymbol{O}_t^{(1)}, \boldsymbol{O}_t^{(2)}, \Lambda) && (13) \end{aligned}$$

3. Statistical Model Including Time-Sequence Matching

3.1. Definition of Model Structure

In the conventional method, the DP matching is conducted based on a similarity measure between two frames. However, this matching might not be optimal for spectral conversion. To overcome this problem, we define the likelihood function $P(\boldsymbol{O}^{(1)}, \boldsymbol{O}^{(2)} \mid \Lambda)$ including the structure of sequence matching. The simultaneous optimization is performed for DP matching and training of model parameters based on the ML criterion. The advantage of the the proposed model can directly deal with two different length sequences $\boldsymbol{O}^{(1)} = \left[\boldsymbol{O}_1^{(1)\top}, \ldots, \boldsymbol{O}_{t^{(1)}}^{(1)\top}, \ldots, \boldsymbol{O}_{T^{(1)}}^{(1)\top}\right]^\top$ and $\boldsymbol{O}^{(2)} = \left[\boldsymbol{O}_1^{(2)\top}, \ldots, \boldsymbol{O}_{t^{(2)}}^{(2)\top}, \ldots, \boldsymbol{O}_{T^{(2)}}^{(2)\top}\right]^\top$. The likelihood function of observation sequences $\boldsymbol{O} = \{\boldsymbol{O}^{(1)}, \boldsymbol{O}^{(2)}\}$ is written as follows:

$$\begin{aligned} P(\boldsymbol{O} \mid \Lambda) = \sum_{\boldsymbol{m}, \boldsymbol{a}} \Big[& P(\boldsymbol{m} \mid \Lambda) P(\boldsymbol{O}^{(1)} \mid \boldsymbol{m}, \Lambda) \\ & \times P(\boldsymbol{a} \mid \Lambda) P(\boldsymbol{O}^{(2)} \mid \boldsymbol{O}^{(1)}, \boldsymbol{m}, \boldsymbol{a}, \Lambda) \Big] \end{aligned} \tag{14}$$

where $\boldsymbol{m} = [m_1, \ldots, m_{t^{(1)}}, \ldots, m_{T^{(1)}}]$ is a mixture number sequence and its element $m_{t^{(1)}}$ means the mixture number of the observation $\boldsymbol{O}^{(1)}$ at time $t^{(1)}$. The variable $\boldsymbol{a} = [a_1, \ldots, a_{t^{(2)}}, \ldots, a_{T^{(2)}}]$ is a transition cost function of the temporal matching and $a_{t^{(2)}} \in \{1, \ldots, T^{(1)}\}$ indicates the frame number of source sequence $\boldsymbol{O}^{(1)}$ which corresponds to the $t^{(2)}$-th frame of target sequence $\boldsymbol{O}^{(2)}$. Each element of the complete data likelihood is defined as follows:

$$P(\boldsymbol{m} \mid \Lambda) = \prod_{t^{(1)}} P(m_{t^{(1)}} \mid \Lambda) \tag{15}$$

$$\begin{aligned} &P(\boldsymbol{O}^{(1)} \mid \boldsymbol{m}, \Lambda) \\ &\quad = \prod_{t^{(1)}} \mathcal{N}\left(\boldsymbol{O}_{t^{(1)}}^{(1)}; \boldsymbol{\mu}_{m_{t^{(1)}}}^{(1)}, \boldsymbol{\Sigma}_{m_{t^{(1)}}}^{(1)}\right) \end{aligned} \tag{16}$$

$$P(\boldsymbol{a} \mid \Lambda) = \prod_{t^{(2)}} P(a_{t^{(2)}} \mid a_{t^{(2)}-1}, \Lambda) \tag{17}$$

$$\begin{aligned} &P(\boldsymbol{O}^{(2)} \mid \boldsymbol{O}^{(1)}, \boldsymbol{m}, \boldsymbol{a}, \Lambda) \\ &\quad = \prod_{t^{(2)}} \mathcal{N}\left(\boldsymbol{O}_{t^{(2)}}^{(2)}; \bar{\boldsymbol{W}}_{m_{a_{t^{(2)}}}} \bar{\boldsymbol{O}}_{a_{t^{(2)}}}^{(1)}, \bar{\boldsymbol{\Sigma}}_{m_{a_{t^{(2)}}}}\right) \end{aligned} \tag{18}$$

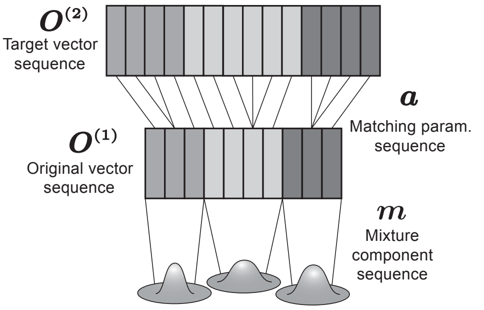


Figure 1: Model structure including DP matching.

where

$$\begin{aligned} \bar{\boldsymbol{W}}_i &= \begin{bmatrix} \boldsymbol{\mu}_i & \boldsymbol{W}_i \end{bmatrix} && (19) \\ \bar{\boldsymbol{O}}^{(1)}_{t^{(1)}} &= \begin{bmatrix} 1 & \boldsymbol{O}^{(1)\top}_{t^{(1)}} \end{bmatrix}^\top && (20) \end{aligned}$$

The model parameters of the proposed model are summarized as follows:

1. $\boldsymbol{w} = \{w_i \,|\, 1 \le i \le M\}$: the mixture weights of the GMM which generates the source feature sequence $\boldsymbol{O}^{(1)}$, where $w_i = P(m_{t^{(1)}} = i \,|\, \Lambda)$ is the probability of i-th mixture.
2. $\boldsymbol{B}^{(1)} = \{b^{(1)}_i \,|\, 1 \le i \le M\}$: the output probability distributions of source feature $\boldsymbol{O}^{(1)}$, where $b^{(1)}_i = P(\boldsymbol{O}^{(1)}_{t^{(1)}} \,|\, m_{t^{(1)}} = i)$ is the probability of source feature vector $\boldsymbol{O}^{(1)}_{t^{(t)}}$ at i-th mixture and which is assumed to be a Gaussian distribution: $\mathcal{N}(\boldsymbol{O}^{(1)}_{t^{(1)}}; \boldsymbol{\mu}^{(1)}_i, \boldsymbol{\Sigma}^{(1)}_i)$ where $\boldsymbol{\mu}^{(1)}_i$ and $\boldsymbol{\Sigma}^{(1)}_i$ are the mean vector and covariance matrix, respectively.
3. $\boldsymbol{c} = \{c_n \,|\, 1 \le n \le N\}$: the transition probabilities of the sequence matching where c_n indicates the probability $P(a_{t^{(2)}} = a_{t^{(2)}-1} + n \,|\, a_{t^{(2)}-1})$. This parameter corresponds to the cost function in the DP matching.
4. $\boldsymbol{B}^{(2)} = \{b^{(2)}_i \,|\, 1 \le i \le M\}$: the output distributions of the target features $\boldsymbol{O}^{(2)}$, where $b^{(2)}_i = P(\boldsymbol{O}^{(2)}_{t^{(2)}} \,|\, \boldsymbol{O}^{(1)}_{t^{(1)}}, m_{t^{(1)}} = i, a_{t^{(2)}} = t^{(1)})$ is the probability of target feature vector $\boldsymbol{O}^{(2)}_{t^{(2)}}$ given the corresponding source feature vector $\boldsymbol{O}^{(1)}_{t^{(1)}}$ at i-th mixture. This conditional distribution is assumed to be a Gaussian distribution: $\mathcal{N}(\boldsymbol{O}^{(2)}_{t^{(2)}}; \boldsymbol{W}_i \boldsymbol{O}^{(1)}_{t^{(1)}} + \boldsymbol{\mu}^{(2)}_i, \boldsymbol{\Sigma}^{(2)}_i)$ where $\boldsymbol{\mu}^{(2)}_i$ and $\boldsymbol{\Sigma}^{(2)}_i$ are the mean vector and the covariance matrix, respectively.

Using shorthand notation, the proposed model is defined as $\Lambda = \{\boldsymbol{w}, \boldsymbol{c}, \boldsymbol{B}^{(1)}, \boldsymbol{B}^{(2)}\}$. Figure 1 shows the generative process of observations $\boldsymbol{O}^{(1)}, \boldsymbol{O}^{(2)}$ by the proposed model. First, a mixture number sequence $\boldsymbol{m}$ is determined according to the weight $P(\boldsymbol{m} \,|\, \Lambda)$ and a source feature sequence $\boldsymbol{O}^{(1)}$ is generated from Gaussian distribution $P(\boldsymbol{O}^{(1)} \,|\, \boldsymbol{m}, \Lambda)$. Second, the frame matching between $\boldsymbol{O}^{(1)}$ and $\boldsymbol{O}^{(2)}$ is determined according to $P(\boldsymbol{a} \,|\, \Lambda)$. Finally, the target feature sequence $\boldsymbol{O}^{(2)}$ is generated according to the conditional Gaussian distribution $P(\boldsymbol{O}^{(2)} \,|\, \boldsymbol{O}^{(1)}, \boldsymbol{m}, \boldsymbol{a}, \Lambda)$ given the source feature sequence.

4. Training Algorithm

The parameters of the proposed model can be estimated via the expectation maximization (EM) algorithm which is an iterative procedure for approximating the Maximum Likelihood (ML) estimate. This procedure maximizes the expectation of the complete data log-likelihood so called $\mathcal{Q}$-function:

$$\mathcal{Q}(\Lambda, \Lambda') = \sum_{\boldsymbol{m},\boldsymbol{a}} P(\boldsymbol{m}, \boldsymbol{a} \,|\, \boldsymbol{O}, \Lambda) \ln P(\boldsymbol{O}, \boldsymbol{m}, \boldsymbol{a} \,|\, \Lambda') \quad (21)$$

The likelihood of the training data is guaranteed to increase by increasing the value of the $\mathcal{Q}$-function:

$$\mathcal{Q}(\Lambda, \Lambda') \ge \mathcal{Q}(\Lambda, \Lambda) \;\Rightarrow\; P(\boldsymbol{O}|\Lambda') \ge P(\boldsymbol{O}|\Lambda) \quad (22)$$

The EM algorithm starts with some initial model parameters and iterates between the following two steps:

$$\begin{aligned} &(\text{E step}): && \text{compute } \mathcal{Q}(\Lambda^{(k)}, \Lambda) \\ &(\text{M step}): && \Lambda^{(k+1)} = \underset{\Lambda}{\text{argmax}}\, \mathcal{Q}(\Lambda^{(k)}, \Lambda) \end{aligned}$$

where k denotes the iteration number. The E-step computes the posterior probabilities over the hidden variables while keeping model parameters Λ fixed to current values. The M-step uses these probabilities to calculate the expected log-likelihood of the training data as a function of the parameters and maximize the $\mathcal{Q}$-function with respect to model parameters Λ. In this procedure, each step increases the value of the $\mathcal{Q}$-function; hence the likelihood of the training data is also guaranteed to increase or remain unchanged on each iteration.

By maximizing the $\mathcal{Q}$-function, the re-estimation formulae in the M-step are derived as follows:

$$\begin{aligned} w_i &= \frac{1}{N^{(1)}} \sum_{t^{(1)}} \gamma^{(1)}_{t^{(1)}}(i) && (23) \\ \boldsymbol{\mu}^{(1)}_i &= \frac{1}{N^{(1)}_i} \sum_{t^{(1)}} \gamma^{(1)}_{t^{(1)}}(i) \boldsymbol{O}^{(1)}_{t^{(1)}} && (24) \\ \boldsymbol{\Sigma}^{(1)}_i &= \frac{1}{N^{(1)}_i} \sum_{t^{(1)}} \gamma^{(1)}_{t^{(1)}}(i) \left(\boldsymbol{O}^{(1)}_{t^{(1)}} - \boldsymbol{\mu}_i\right) \left(\boldsymbol{O}^{(1)}_{t^{(1)}} - \boldsymbol{\mu}_i\right)^\top && \\ & && (25) \\ c_n &= \frac{1}{N^{(2)}} \sum_{t^{(2)}} \sum_{t^{(1)}} \xi^{(2)}_{t^{(2)}}(t^{(1)}, n) && (26) \\ \bar{\boldsymbol{W}}_i &= \left(\sum_{t^{(2)}} \sum_{t^{(1)}} \gamma^{(2)}_{t^{(2)}}(t^{(1)}, i) \boldsymbol{O}^{(2)}_{t^{(2)}} \bar{\boldsymbol{O}}^{(1)\top}_{t^{(1)}} \right) && \\ & \quad \times \left(\sum_{t^{(2)}} \sum_{t^{(1)}} \gamma^{(2)}_{t^{(2)}}(t^{(1)}, i) \bar{\boldsymbol{O}}^{(1)}_{t^{(1)}} \bar{\boldsymbol{O}}^{(1)\top}_{t^{(1)}} \right)^{-1} && (27) \\ \boldsymbol{\Sigma}^{(2)}_i &= \frac{1}{N^{(2)}_i} \sum_{t^{(2)}} \sum_{t^{(1)}} \gamma^{(2)}_{t^{(2)}}(t^{(1)}, i) && \\ & \quad \times \left(\boldsymbol{O}^{(2)}_{t^{(2)}} - \bar{\boldsymbol{W}}_i \bar{\boldsymbol{O}}^{(1)}_{t^{(1)}}\right) \left(\boldsymbol{O}^{(2)}_{t^{(2)}} - \bar{\boldsymbol{W}}_i \bar{\boldsymbol{O}}^{(1)}_{t^{(1)}}\right)^\top && \\ & && (28) \end{aligned}$$

where γ and ξ denote the expectations with respect to the posterior distribution over the hidden variables. These expectations are computed in the E-step by the following equations.

$$\begin{aligned}
\gamma^{(1)}_{t^{(1)}}(i) &= P\left(s_{t^{(1)}}=i\,|\,\boldsymbol{O},\Lambda\right) \\
&= \sum_{\boldsymbol{m},\boldsymbol{a}} P(\boldsymbol{m},\boldsymbol{a}\,|\,\boldsymbol{O},\Lambda)\delta(m_{t^{(1)}},i) && (29) \\
\gamma^{(2)}_{t^{(2)}}(t^{(1)},i) &= P\left(a_{t^{(2)}}=t^{(1)}\,|\,\boldsymbol{O},\Lambda\right) \\
&= \sum_{\boldsymbol{m},\boldsymbol{a}} P(\boldsymbol{m},\boldsymbol{a}\,|\,\boldsymbol{O},\Lambda) \\
&\quad \times\delta(m_{t^{(1)}},i)\delta(a_{t^{(2)}},t^{(1)}) && (30) \\
\xi^{(2)}_{t^{(2)}}(t^{(1)},n) &= P\left(a_{t^{(2)}-1}=t^{(1)},a_{t^{(2)}}=t^{(1)}+n\,|\,\boldsymbol{O},\Lambda\right) \\
&= \sum_{\boldsymbol{m},\boldsymbol{a}} P(\boldsymbol{m},\boldsymbol{a}\,|\,\boldsymbol{O},\Lambda) \\
&\quad \times\delta(a_{t^{(2)}-1},t^{(1)})\delta(a_{t^{(2)}},t^{(1)}+n) && (31)
\end{aligned}$$

and $N^{(1)}$ and $N^{(2)}$ mean the total number of frames of source and target feature sequences, respectively, and $N^{(1)}_i$ and $N^{(2)}_i$ are the occupancy counts of i-th mixture which can be written as follows:

$$N^{(1)}_i=\sum_{t^{(1)}}\gamma^{(1)}_{t^{(1)}}(i),\ \ N^{(2)}_i=\sum_{t^{(2)}}\sum_{t^{(1)}}\gamma^{(2)}_{t^{(2)}}(t^{(1)},i) \qquad (32)$$

where $\delta(\cdot)$ is the Kronecker delta function: $\delta(u,v)=1$ if $u=v$, $\delta(u,v)=0$ otherwise. If we compute expectations in the exact E-step directly according to (29)–(31), we need to consider summations over all the combinations of $\boldsymbol{m}$ and $\boldsymbol{a}$. Therefore the complexity of the E-step becomes $O(M^{T^{(1)}}T^{(1)T^{(2)}})$ and it is infeasible due to the number of hidden variables.

4.1. Variational approximation

Variational methods have been used for approximate maximum likelihood estimation in probabilistic graphical models with hidden variables. We present a structure approximation in which the hidden variables representing mixture number sequences and time sequence matching are decoupled. The variational methods approximate the posterior distribution over the hidden variables by a tractable distribution. Any distribution $Q(\boldsymbol{m},\boldsymbol{a})$ over the hidden variables defines a lower bound on the log-likelihood:

$$\begin{aligned}
\ln P(\boldsymbol{O}\,|\,\Lambda) &= \ln\sum_{\boldsymbol{m},\boldsymbol{a}} Q(\boldsymbol{m},\boldsymbol{a})\frac{P(\boldsymbol{O},\boldsymbol{m},\boldsymbol{a}\,|\,\Lambda)}{Q(\boldsymbol{m},\boldsymbol{a})} \\
&\geq \sum_{\boldsymbol{m},\boldsymbol{a}} Q(\boldsymbol{m},\boldsymbol{a})\ln\frac{P(\boldsymbol{O},\boldsymbol{m},\boldsymbol{a}\,|\,\Lambda)}{Q(\boldsymbol{m},\boldsymbol{a})} \\
&= \sum_{\boldsymbol{m},\boldsymbol{a}} Q(\boldsymbol{m},\boldsymbol{a})\ln P(\boldsymbol{O},\boldsymbol{m},\boldsymbol{a}\,|\,\Lambda) \\
&\quad -\sum_{\boldsymbol{m},\boldsymbol{a}} Q(\boldsymbol{m},\boldsymbol{a})\ln Q(\boldsymbol{m},\boldsymbol{a}) && (33) \\
&= \mathcal{F}(Q,\Lambda) && (34)
\end{aligned}$$

where we have applied Jensen's inequality. Note that the notation of distribution $Q(\boldsymbol{m},\boldsymbol{a})$ is distinct from the notation of $\mathcal{Q}$-function $\mathcal{Q}(\Lambda,\Lambda')$. The difference between $\ln P(\boldsymbol{O}|\Lambda)$ and $\mathcal{F}$ is given by the Kullback-Leibler divergence between $Q(\boldsymbol{m},\boldsymbol{a})$ and the posterior distribution of the hidden variables $P(\boldsymbol{m},\boldsymbol{a}|\boldsymbol{O},\Lambda)$:

$$\begin{aligned}
\mathcal{F} &= \sum_{\boldsymbol{S}} Q(\boldsymbol{m},\boldsymbol{a})\ln\frac{P(\boldsymbol{m},\boldsymbol{a}|\boldsymbol{O},\Lambda)P(\boldsymbol{O}|\Lambda)}{Q(\boldsymbol{m},\boldsymbol{a})} \\
&= \ln P(\boldsymbol{O}|\Lambda)+\sum_{\boldsymbol{S}} Q(\boldsymbol{m},\boldsymbol{a})\ln\frac{P(\boldsymbol{m},\boldsymbol{a}|\boldsymbol{O},\Lambda)}{Q(\boldsymbol{m},\boldsymbol{a})} \\
&= \ln P(\boldsymbol{O}|\Lambda)-\mathrm{KL}(Q||P) && (35)
\end{aligned}$$

Since the true log-likelihood $\ln P(\boldsymbol{O}|\Lambda)$ is independent of $Q(\boldsymbol{m},\boldsymbol{a})$, maximizing the lower bound $\mathcal{F}$ is equivalent to minimizing the Kullback-Leibler divergence. If we allow $Q(\boldsymbol{m},\boldsymbol{a})$ to have complete flexibility then we see that the optimal $Q(\boldsymbol{m},\boldsymbol{a})$ distribution is given by the true posterior $P(\boldsymbol{m},\boldsymbol{a}|\boldsymbol{O},\Lambda)$, in the case where the KL divergence is zero and the bound becomes exact. In order to yield a tractable algorithm, it is necessary to consider a more restricted structure of $Q(\boldsymbol{m},\boldsymbol{a})$ distributions. Given the structure, the parameters of $Q(\boldsymbol{m},\boldsymbol{a})$ are varied so as to obtain the tightest possible bound, which maximizes $\mathcal{F}$.

The variational EM algorithm iteratively maximizes $\mathcal{F}$ with respect to the Q and Λ holding the other parameters fixed:

$$\begin{aligned}
&\text{(E step)} &:&\quad Q^{(k+1)}=\operatorname*{argmax}_{Q\in C}\mathcal{F}(Q,\Lambda^{(k)}) \\
&\text{(M step)} &:&\quad \Lambda^{(k+1)}=\operatorname*{argmax}_{\Lambda}\mathcal{F}(Q^{(k+1)},\Lambda)
\end{aligned}$$

where C is the set of constrained distributions. The maximum in the M-step is obtained by maximizing the term $\sum_{\boldsymbol{S}} Q(\boldsymbol{m},\boldsymbol{a})\ln P(\boldsymbol{O},\boldsymbol{m},\boldsymbol{a}|\Lambda)$ in (33), since the entropy of $Q(\boldsymbol{m},\boldsymbol{a})$ does not depend on model parameters Λ. Therefore, the re-estimation formula (23)–(28) can also be used for the variational EM algorithm by calculating the expectations (29)–(31) with respect to $Q(\boldsymbol{m},\boldsymbol{a})$ instead of the true posterior distribution $P(\boldsymbol{m},\boldsymbol{a}|\boldsymbol{O},\Lambda)$. In this procedure, the lower bound $\mathcal{F}$ is guaranteed to increase instead of the value of the $\mathcal{Q}$-function.

The complexity and the approximation property of the variational EM algorithm are dependent on a constraint to the posterior distribution $Q(\boldsymbol{m},\boldsymbol{a})$ and it should be determined for each structure of graphical models. Here we consider a constrained family of variational distributions by assuming that $Q(\boldsymbol{m},\boldsymbol{a})$ factorizes over $\boldsymbol{m}$ and $\boldsymbol{a}$, so that

$$Q(\boldsymbol{m},\boldsymbol{a})=Q(\boldsymbol{m})Q(\boldsymbol{a}) \qquad (36)$$

where $\sum_{\boldsymbol{m}} Q(\boldsymbol{m})=1, \sum_{\boldsymbol{a}} Q(\boldsymbol{a})=1$. To make the bound as tight as possible, we use elementary calculus of variations to take functional derivatives of the lower bound with respect to $Q(\boldsymbol{m})$ and $Q(\boldsymbol{a})$. In this case, the Euler-Lagrange equation can be solved simply by taking partial derivatives with respect to one of the distributions

$$\begin{aligned}
&\frac{\partial\mathcal{F}}{\partial Q(\boldsymbol{m}=\boldsymbol{m}')} \\
&\quad= \sum_{\boldsymbol{a}} Q(\boldsymbol{a})\ln P(\boldsymbol{O},\boldsymbol{m}',\boldsymbol{a}\,|\,\Lambda)-\ln Q(\boldsymbol{m}')-1 \\
&\quad= \sum_{\boldsymbol{a}} Q(\boldsymbol{a})\ln P(\boldsymbol{O}^{(2)}\,|\,\boldsymbol{O}^{(1)},\boldsymbol{m}',\boldsymbol{a},\Lambda)+\ln P(\boldsymbol{m}'\,|\,\Lambda) \\
&\qquad +\ln P(\boldsymbol{O}^{(1)}\,|\,\boldsymbol{m}',\Lambda)-\ln Q(\boldsymbol{m}')-\mathit{const} \qquad (37)
\end{aligned}$$

The maximum of $\mathcal{F}$ occurs at a critical point subject to the constraint that $\sum_{\boldsymbol{m}} Q(\boldsymbol{m})=1$, and can be found using a Lagrange multiplier $\lambda_{\boldsymbol{m}}$. By setting for each of mixture number sequence $\boldsymbol{m}$

$$\frac{\partial\mathcal{F}}{\partial Q(\boldsymbol{m})}+\lambda_{\boldsymbol{m}}=0 \qquad (38)$$

the optimal approximation of the posterior distribution is derived as

$$Q(\boldsymbol{m}) \propto P(\boldsymbol{m} \,|\, \Lambda)\, P(\boldsymbol{O}^{(1)} \,|\, \boldsymbol{m}, \Lambda) \\ \times \exp\left[\left\langle \ln P(\boldsymbol{O}^{(2)} \,|\, \boldsymbol{O}^{(1)}, \boldsymbol{m}, \boldsymbol{a}, \Lambda) \right\rangle_{Q(\boldsymbol{a})}\right] \quad (39)$$

Similarly to the distribution $Q(\boldsymbol{m})$, the optimal distribution of sequence matching can be obtained as

$$Q(\boldsymbol{a}) \propto P(\boldsymbol{a} \,|\, \Lambda) \\ \times \exp\left[\left\langle \ln P(\boldsymbol{O}^{(2)} \,|\, \boldsymbol{O}^{(1)}, \boldsymbol{m}, \boldsymbol{a}, \Lambda) \right\rangle_{Q(\boldsymbol{m})}\right] \quad (40)$$

By inspection, equation (39) has the same structure as the posterior distribution of standard GMMs, therefore it can be it easily calculated. Moreover, equation (40) is composed of a first-order Markov chain, and it can also be calculated as the standard DP matching (forward-backward algorithm in the training of hidden Markov models). Using these approximate distributions, a new set of expectations can be compute as follows:

$$\begin{aligned} \gamma^{(1)}_{t^{(1)}}(i) &= \sum_{\boldsymbol{m}} Q(\boldsymbol{m}) \delta(m_{t^{(1)}}, i) && (41) \\ \gamma^{(2)}_{t^{(2)}}(t^{(1)}) &= \sum_{\boldsymbol{a}} Q(\boldsymbol{a}) \delta(a_{t^{(2)}}, t^{(1)}) && (42) \\ \gamma^{(2)}_{t^{(2)}}(t^{(1)}, i) &= \gamma^{(1)}_{t^{(1)}}(i) \gamma^{(2)}_{t^{(2)}}(t^{(1)}) && (43) \\ \xi^{(2)}_{t^{(2)}}(t^{(1)}, n) &= \sum_{\boldsymbol{a}} Q(\boldsymbol{a}) \delta(a_{t^{(2)}-1}, t^{(1)}) \\ & \quad \times \delta(a_{t^{(2)}}, t^{(1)} + n) && (44) \end{aligned}$$

4.2. Variational DAEM algorithm

The EM algorithm has the problem that the solution converges to a local optimum and the convergence point depends on the initial model parameters. In the variational EM algorithm, the decoupled posterior distributions are updated individually based not only on the initial model parameters but also on the other distributions, both of which are unreliable at an early stage of training. To avoid this problem, we apply the DAEM algorithm to the algorithm derived in the previous section.

In the DAEM algorithm, the problem of maximizing the log-likelihood is reformulated as minimizing the thermodynamic free energy defined as

$$\mathcal{L}_\beta = -\frac{1}{\beta} \ln \sum_{\boldsymbol{m}, \boldsymbol{a}} P^\beta(\boldsymbol{O}, \boldsymbol{m}, \boldsymbol{a} \,|\, \Lambda) \quad (45)$$

where $1/\beta$ called the "temperature" and this cost function can be rewritten by using Jensen's inequality:

$$\begin{aligned} -\mathcal{L}_\beta &= \frac{1}{\beta} \ln \sum_{\boldsymbol{m}, \boldsymbol{a}} Q_\beta(\boldsymbol{m}, \boldsymbol{a}) \frac{P^\beta(\boldsymbol{O}, \boldsymbol{m}, \boldsymbol{a} \,|\, \Lambda)}{Q_\beta(\boldsymbol{m}, \boldsymbol{a})} \\ &\geq \sum_{\boldsymbol{m}, \boldsymbol{a}} Q_\beta(\boldsymbol{m}, \boldsymbol{a}) \ln P(\boldsymbol{O}, \boldsymbol{m}, \boldsymbol{a} \,|\, \Lambda) \\ & \quad - \frac{1}{\beta} \sum_{\boldsymbol{m}, \boldsymbol{a}} Q_\beta(\boldsymbol{m}, \boldsymbol{a}) \ln Q_\beta(\boldsymbol{m}, \boldsymbol{a}) && (46) \\ &= \mathcal{F}_\beta(Q_\beta, \Lambda) && (47) \end{aligned}$$

where $-\mathcal{F}_\beta(Q_\beta, \Lambda)$ is the same form as the free energy in statistical physics, and maximizing $\mathcal{F}_\beta(Q_\beta, \Lambda)$ with a fixed temperature can be interpreted as the approach to thermodynamic equilibrium. In the algorithm, the temperature is gradually decreased and the function is deterministically optimized at each temperature. The procedure of the DAEM algorithm can be summarized as follows:

1 Give an initial model and set $\beta = \beta_{min}$

2 Iterate EM-steps with β fixed until F_β converged:

$$\begin{aligned} \text{(E step)} &: \quad Q_\beta^{(k+1)} = \underset{Q_\beta \in C}{\operatorname{argmax}} \mathcal{F}_\beta(Q_\beta, \Lambda^{(k)}) \\ \text{(M step)} &: \quad \Lambda^{(k+1)} = \underset{\Lambda}{\operatorname{argmax}} \mathcal{F}_\beta(Q_\beta^{(k+1)}, \Lambda) \end{aligned}$$

3 Increase β.

4 If $\beta > 1$, stop the procedure. Otherwise go to step 2.

where $1/\beta_{min}$ is an initial temperature and should be chosen as a high enough value that the EM-steps can achieve a single global maximum of $\mathcal{F}_\beta$. At the initial temperature, the entropy of Q_β is intended to be maximized rather than the $\mathcal{Q}$ function (the first term of equation (46)); therefore Q_β takes a form nearly uniform distribution. While the temperature is decreasing, the form of Q_β changes from uniform to the original posterior and at the final temperature $1/\beta = 1$, the negative free energy $\mathcal{F}_\beta$ becomes equal to the lower bound $\mathcal{F}$, accordingly the DAEM algorithm agrees with the original EM algorithm.

Similarly to the variational EM algorithm, the optimal distribution which maximizes $\mathcal{F}_\beta$ is given by

$$Q_\beta(\boldsymbol{m}) \propto P^\beta(\boldsymbol{m} \,|\, \Lambda)\, P^\beta(\boldsymbol{O}^{(1)} \,|\, \boldsymbol{m}, \Lambda) \\ \times \exp\left[\beta \left\langle \ln P(\boldsymbol{O}^{(2)} \,|\, \boldsymbol{O}^{(1)}, \boldsymbol{m}, \boldsymbol{a}, \Lambda) \right\rangle_{Q(\boldsymbol{a})}\right] \quad (48)$$

$$Q_\beta(\boldsymbol{a}) \propto P^\beta(\boldsymbol{a} \,|\, \Lambda) \\ \times \exp\left[\beta \left\langle \ln P(\boldsymbol{O}^{(2)} \,|\, \boldsymbol{O}^{(1)}, \boldsymbol{m}, \boldsymbol{a}, \Lambda) \right\rangle_{Q(\boldsymbol{m})}\right] \quad (49)$$

5. ML-Based Spectral Conversion

The converted feature sequence $\boldsymbol{O}^{(2)}$ can be obtained by maximizing the lower bound of the likelihood. Taking the derivative of $\mathcal{F}$ with respect to $\boldsymbol{O}^{(2)}$, the optimal sequence is given as the following equation.

$$\hat{\boldsymbol{O}}^{(2)}_{t^{(2)}} = \left(\sum_{t^{(1)}} \sum_i \gamma^{(1)}_{t^{(1)}}(i) \gamma^{(2)}_{t^{(2)}}(t^{(1)}) \boldsymbol{\Sigma}_i^{(2)-1} \right)^{-1} \\ \times \left(\sum_{t^{(1)}} \sum_i \gamma^{(1)}_{t^{(1)}}(i) \gamma^{(2)}_{t^{(2)}}(t^{(1)}) \boldsymbol{\Sigma}_i^{(2)-1} \bar{\boldsymbol{W}}_i \bar{\boldsymbol{O}}^{(1)}_{t^{(1)}} \right) \quad (50)$$

Although the proposed method can represent different length sequences of source and target features, the transition probability $P(\boldsymbol{a} \,|\, \Lambda)$ assumed in this paper is insufficient to generate the duration of the converted feature sequence. Therefore, one to one frame matching is used in the conversion process (i.e. $a_{t^{(2)}} = t^{(2)}$). Under this assumption, if $t^{(1)} = t^{(2)}$, $\gamma^{(2)}_{t^{(2)}}(t^{(1)}) = 1$, otherwise $\gamma^{(2)}_{t^{(2)}}(t^{(1)}) = 0$, therefore equation (50) can be rewritten as

$$\begin{aligned} \hat{\boldsymbol{O}}^{(2)}_{t^{(2)}} &= \left(\sum_i \gamma^{(1)}_{t^{(2)}}(i) \boldsymbol{\Sigma}_i^{(2)-1} \right)^{-1} \\ & \quad \times \left(\sum_i \gamma^{(1)}_{t^{(2)}}(i) \boldsymbol{\Sigma}_i^{(2)-1} \bar{\boldsymbol{W}}_i \bar{\boldsymbol{O}}^{(1)}_{t^{(2)}} \right) && (51) \end{aligned}$$

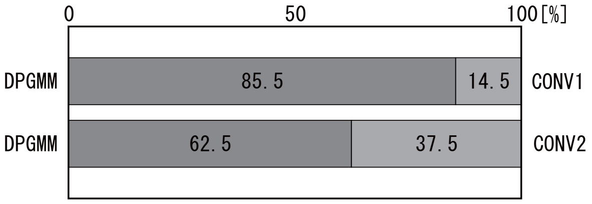


Figure 2: Results of perference test.

Given the temporal matching $\boldsymbol{a}$, the optimal converted feature sequence still depends on the posterior distribution of mixture number sequence $Q(\boldsymbol{m})$. Therefore, an iterative update procedure is required. The conversion procedure is summarized as follows:

1. Compute the expectation $\gamma^{(1)}_{t^{(1)}}(i)$ for each frame of the source feature sequence $\boldsymbol{O}^{(1)}$ (omitting the last term of equation (39), that is, $Q(\boldsymbol{m}) \propto P(\boldsymbol{m} \mid \Lambda)\, P(\boldsymbol{O}^{(1)} \mid \boldsymbol{m}, \Lambda)$ and equation (41)).
2. The converted feature sequence $\hat{\boldsymbol{O}}^{(2)}$ is obtained by using $\gamma^{(1)}_{t^{(1)}}(i)$ (equation (51)).
3. Update the expectation $\gamma^{(1)}_{t^{(1)}}(i)$ by using both the source feature sequence $\boldsymbol{O}^{(1)}$ and the converted feature sequence $\hat{\boldsymbol{O}}^{(2)}$ (equation (39), (41)).
4. If $\mathcal{F}$ is converged, stop the procedure. Otherwise go to 2.

6. Experiments

Voice conversion experiments on the ATR Japanese speech database were conducted. Two male speakers are selected as a source and a target speaker (source:mtk target:mht). Twenty sentences uttered by the both speakers were used for training and 200 sentences were used for evaluation. The speech data were down-sampled from 20KHz to 16KHz, windowed at a 5-ms frame rate using a 25-ms Blackman window, and parameterized into 24 mel-cepstral coefficients excepting the zero-th coefficients and their first order derivative were used as the dynamic features.

Although voice similarity to target speakers is primarily required in voice conversion, we conducted subjective preference tests in speech quality because the proposed method is expected to improve speech quality. In preliminary experiments, it is confirmed that the proposed method obtained the almost same or higher performance in voice similarity than the conventional method. The number of mixtures was set to four which achieved the best performance for the both conventional and proposed methods on objective tests using the mel-cepstrum distance. The number of subject was eight and each subject evaluates 25 sentences in total 200 sentences.

Figure 2 shows the results of the preference test. The notation “DPGMM” means the proposed method with the DAEM algorithm, and “CONV1” and “CONV2” indicate the conventional GMM-based methods without and with iterative updates of the DP matching and spectral conversion, respectively. The iterative procedure is as follows: the DP matching using the Euclid distance is conducted for each training utterance and initial alignments are obtained. The GMM parameters are estimated from the joint features constructed by using the initial alignments. Then source feature sequences are converted and new alignments are obtained by using the converted sequences instead of source feature sequences. These processes are iterated until convergence.

Comparing the proposed method “DPGMM” and the conventional GMM-based method “CONV1,” “DPGMM” is superior than the “CONV1” in the preference test. This means that the GMM-based method without iterative matching could not obtain appropriate alignments between source and target feature sequences because the matching is performed using only the similarity measure between two frames. Although the conventional method iterating the DP matching and the spectral conversion can improve the accuracy of spectral conversion, “DPGMM” was still better than “CONV2”. This is because the DP matching and training GMMs are simultaneously optimized based on the integrated objective measure. It could also be an advantage that “DPGMM” utilizes all frame combination of source and target features, since hidden variable sequences representing the DP matching are marginalized. Furthermore, the cost function of the DP matching was optimized based on the ML criterion, even though fixed cost was used in the conventional method.

7. Conclusions

This paper has proposed a new statistical model for voice conversion which includes matching between source and target feature sequences in the likelihood function. The proposed model provides an ML-based consistent algorithm for training model parameters, sequence matching and spectral conversion. In the experiments, it is confirmed that the proposed method achieved higher performance than the conventional GMM-based approaches. Investigation of the optimal model structure and spectral conversion including duration changes will be future works.

8. References

[1] Yining Chen, Min Chu, Eric Chang, Jia Liu, and Runsheng Liu, “Voice conversion with smoothed GMM and MAP adaptation,” *Proc. of EUROSPEECH*, pp.2413–2416, Sep. 2003

[2] Tomoki Toda, Alan W Black, and Keiichi Tokuda, “Spectral conversion based on maximum likelihood estimation considering global varinace of converted parameter,” *Proc. of ICASSP, vol.1*, pp.9–12, Mar. 2005.

[3] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Koyayashi, and Tadashi Kitamura, “Speech parameter generation algorithms for HMM-based speech synthesis,” *Proc. of ICASSP, vol.3*, pp.1315–1318, Jun. 2000.

[4] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola and L. K. Saul, “An Introduction to Variational Methods for Graphical Models,” Machine Learning, vol.37, pp.183-233, Jan. 1997.

[5] Z. Ghahramani, “On Structured Variational Approximations,” University of Toronto Technical Report, CRG-TR-97-1, 1997, revised 2002.

[6] N. Ueda, and R. Nakano, “Deterministic Annealing EM Algorithm,” Neural Networks, vol.11, no.2, pp.271–282, 1998.

Analysis of Affective Speech Recordings using the Superpositional Intonation Model

Esther Klabbers, Taniya Mishra, Jan van Santen

Center for Spoken Language Understanding
OGI School of Science & Engineering at OHSU
20000 NW Walker Road, Beaverton, OR, 97006, USA
klabbers@cslu.ogi.edu

Abstract

This paper presents an analysis of affective sentences spoken by a single speaker. The corpus was analyzed in terms of different acoustic and prosodic features, including features derived from the decomposition of pitch contours into phrase and accent curves. It was found that sentences spoken with a sad affect were most easily distinguishable from other affects as they were characterized by a lower F_0, lower phrase and accent curves, lower overall energy and a higher spectral tilt. Fearful was also relatively easy to distinguish from angry and happy as it exhibited flatter phrase curves and lower accent curves. Angry and happy were more difficult to distinguish from each other, but angry was shown to exhibit a higher spectral tilt and a lower speaking rate. The analysis results provide informative clues for synthesizing affective speech using our proposed recombinant synthesis method.

1. Introduction

Generating meaningful and natural sounding prosody is a central challenge in TTS. In traditional concatenative synthesis, the challenge consists of generating natural sounding target prosodic contours and imposing these contours on recorded speech without causing audible distortions. In unit selection synthesis, the challenge consists of selecting acoustic units from a large speech corpus that optimally match the phonemic and prosodic contexts required. When expanding a prosodic domain from a neutral reading style to more expressive styles, the size of the speech corpus grows exponentially.

We are developing a new approach to speech synthesis, called *recombinant synthesis* (also known as multi-level unit selection synthesis) in which natural prosodic contours and phoneme sequences are recombined using a superpositional framework [13]. The proposed method can use different speech corpora for selecting phoneme units and pitch contour components. As the prosodic space is expanded to include more speaking styles or sentence types (i. e. lists), more pitch contours can be added to the prosodic corpus. The prosodic corpus does not contain the raw pitch contours, as concatenating them would result in audible discontinuities [12], but rather contains phrase curves and accent curves that are derived from the original pitch contour. Recombinant synthesis has advantages over both traditional concatenative synthesis and unit selection in that (i) the pitch contours selected from the database are natural and smooth, leading to higher quality synthesis, and (ii) much smaller speech corpora are required as the coverage of acoustic and prosodic features is additive instead of multiplicative.

The goal is to select natural-sounding pitch contours that are appropriate for the given context and that are close enough to the original prosody of the selected phoneme units to minimize signal degradation due to pitch modification [5]. This paper discusses preliminary findings related to a set of affective recordings. There have been several studies analyzing affective speech for synthesis purposes [3, 1, 14, 9]. Typically they explore simple prosodic features such as the F_0 mean and range, and phoneme durations. Some studies [9] have gone further and examined pitch contour shapes in different affective conditions. The recordings used in our analysis are by no means complete, nor is the set large enough to make exhaustive predictions, but the analysis method and the acoustic features used to analyze the data will provide valuable information about distinguishing different affects and hopefully will be useful in generating appropriate affective speech. The relevance of acoustic features was analyzed using a repeated measures analysis of variance paradigm and paired t-tests were performed to determine the acoustic differences between pairs of affects.

2. Recordings

This study used a set of affective recordings that was collected for a previous study. A group of 42 actors read 24 sentences in 4 different affects: Angry (A), Happy (H), Fearful (F), and Sad (S). There was considerable variability within subjects with respect to expressing the different affects. For the purposes of speech synthesis of affective speech, one single speaker was chosen for analysis. The chosen speaker is an 8-year old girl who was the most consistent in her renditions of the different affects. This was established in a listening experiment, where 12 people listened to all sentences in random order and assigned affect labels and a confidence score to them.

The speakers did not produce neutral recordings for these 24 sentences. However, the sentences are semantically unbiased in their affective content, i. e., it is impossible to predict which affect is intended from the text alone. Because there are four different versions of each sentence, different affects can be compared side-by-side. The sentences consist of a single phrase 2–5 words in length. The sentences are preceded by short "vignettes" which cue the speaker to produce the correct affect. Table 1 presents 4 example vignettes for one of the sentences. The simulated vocal expressions obtained in this manner will yield more intense, prototypical expressions of affect [14], but for speech synthesis purposes this is desired to ensure correct

This research was conducted with support from NSF grant 0205731, "Prosody Generation in Child-Oriented Speech" and NIH Grant 1R01DC007129, "Expressive and Receptive Prosody in Autism".

Angry	Happy	Fearful	Sad
The parents had left their teenager home alone for the weekend and had come home to a house that had been turned upside down. The father said angrily:	Her best friend had moved away four months ago. She was contemplating this as the doorbell rang. It was her.	Suddenly the tornado made a turn, and now was heading for where John was standing. 'I'm gonna get killed by a tornado.	She cried when her parents told her that her best friend had been in an automobile accident and may never walk again. She was overcome with grief, and said:
"I don't believe it!"			

Table 1: Affective vignettes for the sentence "I don't believe it".

perceived affects. Moreover, the perception experiment showed that listeners could correctly recognize the intended affects, reflecting the fact that these recordings represent normal expression patterns.

3. Analysis

In this study we used analysis features based on pitch, duration, and energy to distinguish different affects. The pitch values for the recordings were computed using Praat [2]. The advantage of using Praat is that it is able to deal with high frequencies, which are more common in childrens' voices and it allows manual adjustments to the voicing flags on a frame-by-frame basis to obtain the best pitch contour. All resulting pitch contours were manually checked to make sure they were correct. The pitch was used to measure global features such as F_0 mean and range. In addition, more detailed features were computed relating to the phrase curves and accent curves obtained by decomposing the pitch contours according to the superpositional model. The decomposition algorithm will be described in more detail in 3.1.

Phoneme segmentation was performed using CSLU's phonetic alignment system [4]. The phoneme alignment was hand-corrected. The phoneme labeling was used to compute phoneme durations. In addition, the sentences were labeled according to their foot structure. A foot is defined as consisting of an accented syllable followed by all unaccented syllables until the next accented syllable or a phrase boundary. The foot structure could be different in each affect rendition, as the number of accents was not always the same. As a rule, foot labeling was based on the presence of audible emphasis on a syllable. The foot labels were checked by two colleagues to ensure consistency. Phrase-initial unstressed syllables are called *anacrusis*. The accent curves on anacruses were excluded from our analysis.

Variations in acoustic features between different speaking styles are not restricted to prosody, but also include spectral features such as spectral tilt and spectral balance. Spectral balance represents the amplitude pattern across four different frequency regions. These four bands are generally phoneme independent, and contain the first, second, third and fourth formant for most of the phonemes. Formants contain the largest portion of energy in the frequency domain. Moreover, when some prosodic factors change, e. g., from unstressed to stressed, the energy near formants will be amplified much more than those near other frequency locations. Choosing frequency bands according to formant frequencies has an important advantage for statistical analysis, because it will reduce interactions between phoneme identity and prosodic factors. For speech with 16 kHz sampling rate, the four bands are defined as: B1:0-800Hz, B2: 800-2500Hz, B3: 2500-3500Hz, B4: 3500-8000Hz. Previous research has shown systematic variations in spectral balance in phonemes when influenced by syllable stress, word accent, proximity to phrase boundary, and neighboring phonemes [11, 7]. The four band values were computed as an average of three data points nearest to the peak location in the foot. These points were always located in the stressed vowel. The overall energy was computed as a sum of the four bands. The spectral tilt was computed as {-2 * B1 - B2 + B3 + 2 * B4}. Previous studies have shown that our synthesis system is capable of synthesizing speech with different spectral balance profiles successfully without introducing additional signal degradation [11, 7].

3.1. Decomposition of pitch curves

In the general superpositional model of intonation, the pitch contour is described as the sum of component curves that are associated with different phonological levels, specifically, the phoneme, foot, and phrase level [10, 12]. To apply this model to the recombinant synthesis method, the pitch curves in the prosodic corpus need to be automatically decomposed into their corresponding phrase and accent curves. The phrase curve is the underlying curve that spans an entire phrase. It provides information about the baseline pitch and the global declination. The accent curves span the foot and they convey the amount of emphasis exerted on accented syllables.. The typical accent curve template is characterized by an up-down movement in the pitch, although there are also templates for negative accents and phrase-final accents containing continuation rises. Decomposing pitch curves is not trivial, since successive accents may overlap in time and we want to impose as few constraints as possible on the shapes of accent and phrase curves.

The proposed decomposition algorithm has been developed using increasingly more difficult sentences. The first step was to decompose synthetic F_0 contours that were generated with our implementation of the superpositional model and curves generated with the Fujisaki model [12]. The next step was to decompose natural F_0 contours from declarative all-sonorant sentences [8]. The last step involved decomposing natural F_0 contours from unrestricted declarative sentences containing continuation rises [6].

Figure 1 shows the decomposition of the F_0 contours for the sentence "I don't believe it" for all four affects. The estimated F_0 contours, as depicted by the solid continuous lines provide close approximations of the raw pitch contour. The decomposition algorithm optimizes the Root Weighted Mean Square Error (RWMSE) where the weights are determined by the amplitude and voicing flags. The overall RWMSE obtained for this database is 15.65 Hz, which is appropriate given the fact that the recordings are extremely expressive and come from a child whose F_0 excursions occasionally exceeded 800 Hz.

The decomposition takes place on a foot-by-foot basis. The

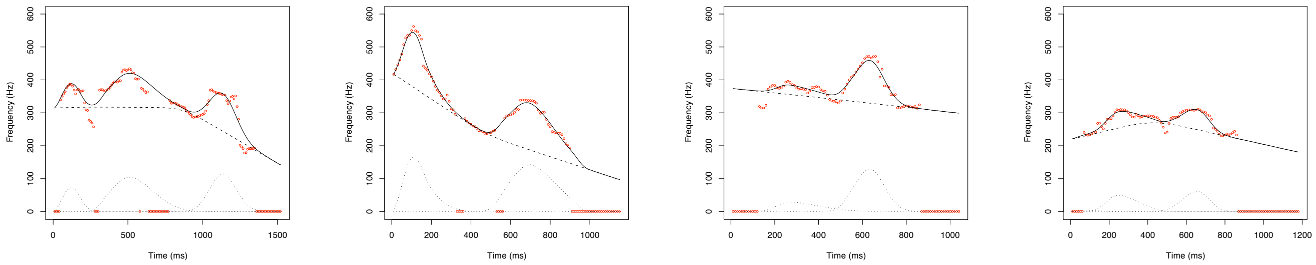


Figure 1: Decomposition of the F_0 contour into a phrase curve and accent curves for the sentence "I don't believe it".

Acoustic feature	F-value	p-value	Sig.
Average F_0	15.25	4.06e-08	$\ast\ast\ast$
F_0 range	21.85	9.58e-11	$\ast\ast\ast$
Phrase curve range	8.39	5.51e-05	$\ast\ast\ast$
Average phrase curve slope	5.57	0.0015	$\ast\ast$
Start of phrase curve	8.66	4.09e-05	$\ast\ast\ast$
End of phrase curve	3.95	0.011	$\ast$
Number of accents	1.85	0.14	
First accent amplitude	9.89	1.04e-05	$\ast\ast\ast$
Last accent amplitude	9.49	1.63e-05	$\ast\ast\ast$
Average accent amplitude	12.68	5.38e-07	$\ast\ast\ast$
Speaking rate	1.03	0.38	
Overall energy	29.18	2.62e-13	$\ast\ast\ast$
Spectral tilt	7.47	0.00016	$\ast\ast\ast$

Table 2: Results for Anova with repeated measures for each acoustic feature. Sig. stands for significance, where $\ast$ corresponds to a p-value < 0.05, $\ast\ast$ corresponds to a p-value < 0.01 and $\ast\ast\ast$ corresponds to a p-value < 0.001

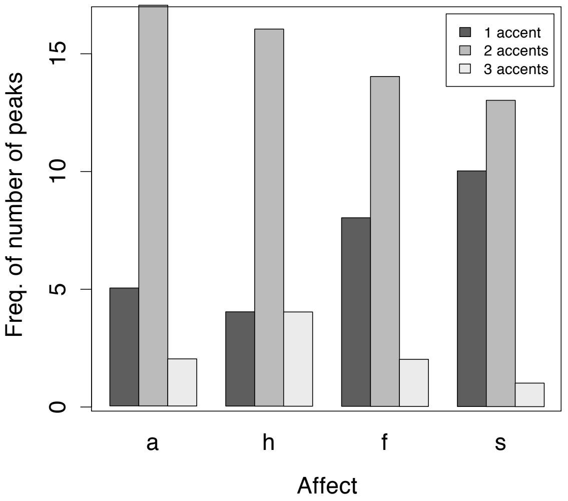


Figure 2: Number of accents per sentence.

phrase curve consists of piecewise linear segments that are smoothed to create a more natural looking curve. The accent curves are based on generic accent templates which are warped in the time and frequency domain to best match the target curve. Because the sentence content is known and phonemes and feet are labeled, the approximate locations of the accent curves are known. The algorithm requires an approximate location of the accent peak. We obtained initial peak location estimates automatically which were hand-corrected to ensure a close fit.

4. Analysis results

In order to determine which acoustic features were significantly different between affects, an analysis of variance with repeated measures was performed on each acoustic feature. Affect was the dependent variable and sentence number was the error term (because the acoustic features observed are not independent of the sentence content uttered). The analysis of variance results in Table 2 show that most of the features we examined were significantly different across affects. The only features that were not significantly different were the number of accents and the speaking rate. The end value of the phrase curve was only slightly significant.

Most studies on prosody in affective speech ignore the fact that the number of accents might be different across conditions. Informal analysis of the recordings exposed a tendency for speakers to emphasize more words in excited conditions such as angry and happy. Although the number of accents per sentence is not significantly different across affects for the current speaker, there is a clear trend visible in Figure 2. The fearful and sad sentences tend to have fewer accents than the angry and happy conditions. We believe that this trend will become more obvious with longer sentences and text material. The reason it is not signifcant in this corpus is that the number of stressable words is limited. The analysis of variance presents the overall significance of a feature, but it does not show differences between pairs of affects. Therefore, paired t-tests were performed for each acoustic feature comparing pairs of affects to determine which features were significantly different between each pair.

4.1. Overall pitch

The mean and range of F_0 are two popular features that have been reported on in many studies. Banse and Scherer [1] summarize previous findings as follows. Affects involving high arousal levels such as anger, fear, and happiness are characterized by an increase in F_0 mean and range whereas sadness is characterized by a decrease in F_0 mean and range. Cahn [3] reported a similar trend for F_0 range, but for F_0 mean her findings were much different in that fear showed the highest contribution followed by sad, then happy and angry. Figure 3 shows the mean differences between the affect pairs and the 95% confidence intervals for the F_0 mean for our speaker. The t-values and p-values were obtained by performing the paired t-tests. The F_0 mean values for this recording set were 279 Hz for happy, 261 Hz for angry, 250 Hz for fearful, and 177 Hz for sad. The sad affect is significantly lower in pitch than the other three emotions, in line with previous studies. Happy is slightly higher than fearful. The differences between angry and happy and between angry and fearful are not significant. The F_0 range shows the same picture as the F_0 mean in terms of the differences between the affect pairs. The average F_0 range is 581 Hz for happy, 544 Hz for angry, 431 Hz for fearful, and 309 Hz for sad. Note that these are recordings from a child, which explains the high range in F_0. All F_0 range differences between affect pairs are significant, except the difference between angry and happy.

The F_0 mean and range are not very informative features for describing the pitch contours. Using parameters derived from the phrase curves and accent curves as obtained from our decomposition algorithm, allows for a more detailed description of the differences between affects.

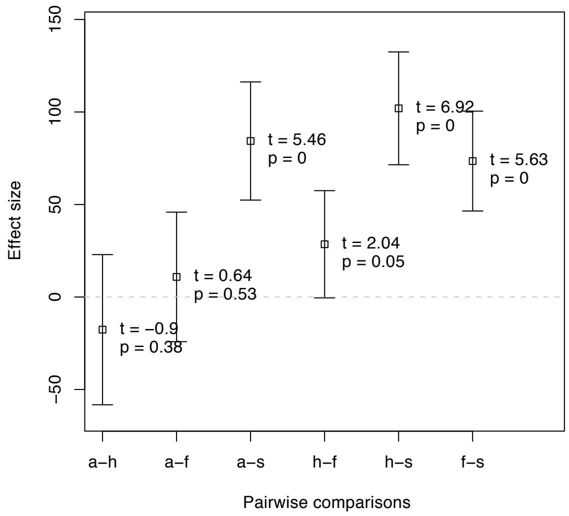


Figure 3: F_0 mean differences between affects.

4.2. Phrase curves

Due to the shortness of the sentences, there were no minor phrase boundaries and as such there was only one phrase curve per sentence. Anger and fear have been found to have more declination than happy and sad [1], although in a different study anger and sad were found to have a level contour slope and happy and fear had a rising contour slope [3]. The problem with these analyses is that they derive the declination slope from the raw pitch contour, the slope of which is polluted by the pitch accent prominences. The main advantage of our decomposition algorithm is that it allows for a separation of the declination in the phrase curve from the accent curves. Figure 4 shows differences in the average phrase curve range, which is defined as the difference between the maximum and the minimum value of the phrase curve. The results show that the differences in phrase curve range between angry and happy and between fearful and sad are not significant. However, both angry and happy have a significantly larger range than fearful and sad. The average phrase curve range is 188 Hz for happy, 200 Hz for angry, 120 Hz for fearful and 90 Hz for sad.

We also computed the average slope of the phrase curve (or declination). The results show the same trends as for the phrase curve range differences in that the differences between angry and happy and between fearful and sad are not significant. However, both angry and happy have significantly less declination than fearful and sad. The average slope of the phrase curve is -1.74 units for angry, -1.45 for happy, -0.29 for fearful and -0.71 for sad. The phrase curves for the fearful condition are almost flat.

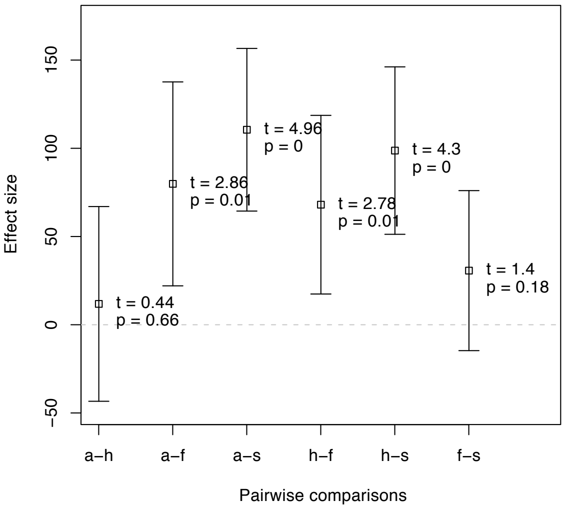


Figure 4: Average phrase curve range differences between affects.

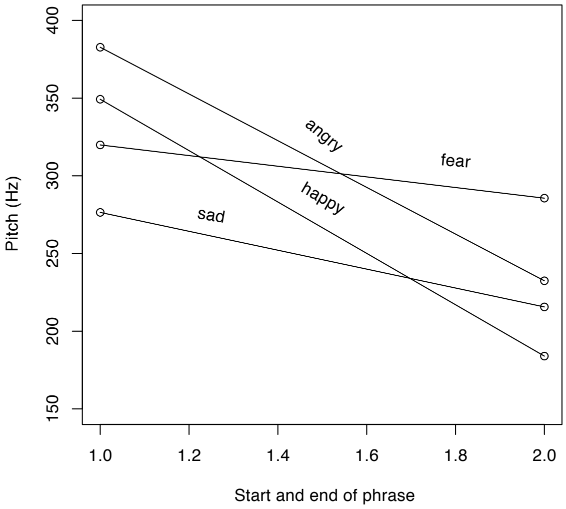


Figure 5: Average phrase curve start and end values for each affect.

Figure 5 displays the average start and end points of the phrase curve for each affect. The difference in slope is clearly visible between on the one hand the angry and happy and on the other hand the fearful and sad affects. The slope difference is mainly related to the end point of the phrase curve. The phrase curve on average starts higher for the angry affect than for happy, followed by fearful and sad. But the phrase curve ends highest for fear, followed by angry, sad, and happy. These findings will be very helpful for applying appropriate phrase curves to the phoneme sequences in our recombinant synthesis system.

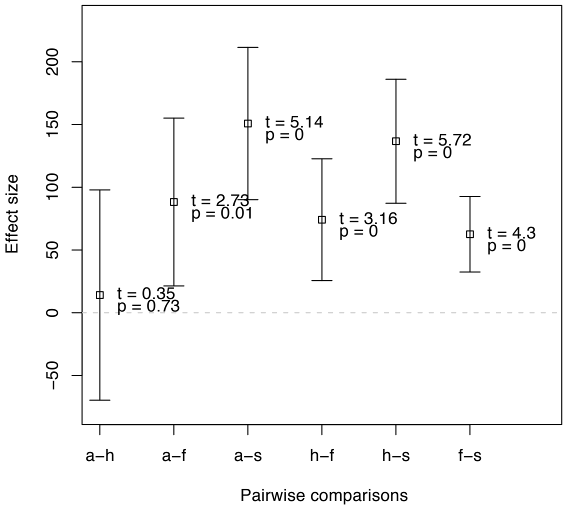


Figure 6: Average accent curve height differences between affects.

4.3. Accent curves

The start of the accent curve always coincides with the start of the foot, which is always a stressed/accented syllable. The end of the foot is located at the end of an unstressed syllable either right before the start of the following foot, or a phrase boundary. However, previous research has shown that the end of the accent curve does not need to coincide with the end of the foot, leading to overlapping accent curves [8]. We were able to provide a satisfactory fit to the pitch contours using accent curve templates for the basic up-down shape, negative accents and accents with continuation rises. We found some negative accents in our corpus, but the occurrence of negative accents was not significantly different between affects. Because the sentences were so short, there were no minor phrase boundaries and thus no continuation rises at those locations. But the speaker would sometimes end sentences in a continuation rise. Our hypothesis was that this occurred mostly in the fearful and sad affects, but no significant effect was found. For the measurement of accent curve amplitudes, the negative accents were excluded from the analysis.

Figure 6 displays the average differences in accent curve amplitudes between the affect pairs. The accent curve amplitude is measured at the peak location. It can be observed that the difference in accent curve amplitudes is not significant for the angry-happy comparison, but it is significant for all other comparisons. Both angry and happy have higher accent amplitudes than fearful and sad. Fearful has higher accent curve amplitudes than sad. The average values for the four affects are: 172 Hz for angry, 173 Hz for happy, 77 Hz for fearful and only 27 Hz for sad.

For sentences that had more than one accent, we also studied the average accent curve amplitude for the first accent versus that of the last accent. The averages are based on 60 out of 96 sentences. The first peak was on average 133 Hz for angry, 176 Hz for happy, 76 Hz for fearful and 29 Hz for sad. For the last peak the average values were 157 Hz for angry, 181 Hz for happy, 93 Hz for fearful and 18 Hz for sad. This shows that for all conditions except sad, the final accent had a higher amplitude than the first one.

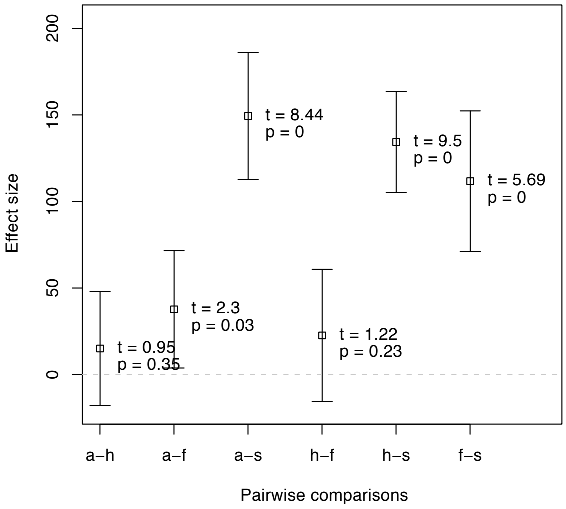


Figure 7: Average overall energy differences between affects.

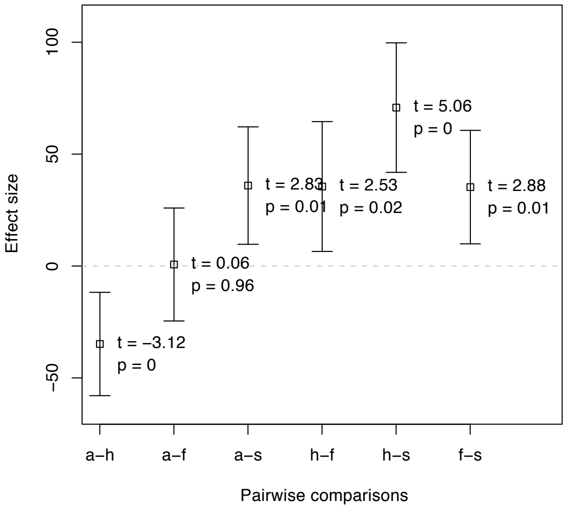


Figure 8: Average spectral tilt differences between affects.

4.4. Energy

Figure 7 shows the overall energy differences between affect pairs. The overall energy was computed as the sum of the four broad spectral band averages. As can be seen, the overall energy for sad is much lower than for the other three affects. Fearful is significantly lower than angry but its lower overall energy with respect to happy is not significant. Angry is louder than happy but again this difference is not significant. The average overall energy for angry is an order of magnitude of 409 for angry, 394 for happy, 372 for fearful and 260 for sad.

Although spectral tilt was not found to be a significant factor using the analysis of variance, we do include it here, as the paired t-test showed that there was an important difference in spectral tilt between angry and happy. This makes the spectral tilt one of the few parameters to distinguish angry from happy in our corpus. Figure 8 displays the average spectral tilt differences between affect pairs. The most important finding is that

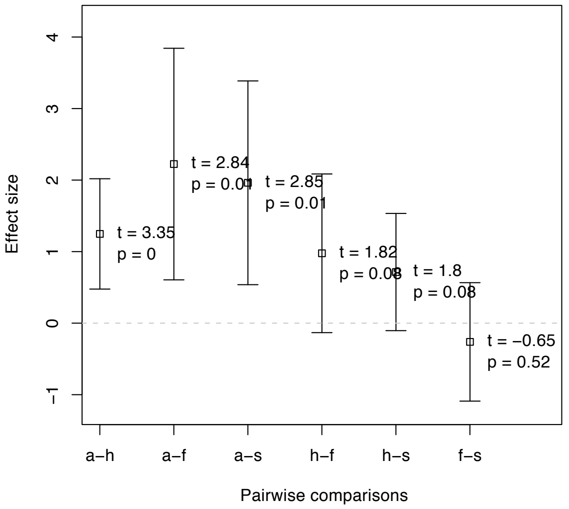


Figure 9: Average speaking rate differences between affects.

the spectral tilt in anger is significantly lower than in happy. The average values for spectral tilt were -87 units for angry, -53 for happy, -88 for fearful and -123 for sad. Thus, sad has the lowest amount of high-frequency energy whereas the other three emotions, all three of which are associated to higher arousal levels according to Banse and Scherer, have higher amounts of high-frequency energy, which is reported to be due to an increased vocal effort by the speaker [1].

4.5. Speaking rate

Phoneme durations and pause lengths are often included in an analysis of different affects. Because the sentences in our corpus are relatively short, there are no intermediate pauses that can be analyzed. We computed the average speaking rate by dividing the total phoneme duration (excluding pauses) by the number of phonemes. The average speaking rate was 140 ms/phoneme for angry, 127 ms/phoneme for happy, 117 ms/phoneme for fearful and 116 ms/phoneme for sad. This is surprising as we expected the angry affect to be faster than the other affects, but for this speaker that turned out not to be the case. We also considered other duration measures such as vowel durations and voiced portion durations, but the effects were similar to the speaking rate findings, so we don't go into detail here.

5. Conclusion

The sad affect presents the most distinct acoustic and prosodic features from the other three affects. The sentences have a lower overall energy and higher spectral tilt. The phrase curves are lower and the accent curve amplitudes are much lower than in other affects. The other three affects (angry, happy and fearful) are all high-arousal emotions and can be more easily confused with each other. However, our analysis has shown that we can distinguish the three affects for our speaker. Fearful is distinguishable from angry and happy by showing a lower F_0 range, a flatter phrase curve and lower accent curve amplitudes. Angry is distinguishable from happy by displaying a higher spectral tilt and a slower speaking rate.

The results provide a promising start to synthesizing expressive speech using our recombinant synthesis approach. The decomposition algorithm was shown to do a good job decomposing the pitch contours into phrase and accent curves, despite the fact that we were dealing with highly expressive children's speech. This demonstrates the fact that a prosodic corpus using neutrally read sentences can be used to select phrase and accent curves, which can then be warped using different warping functions for each affect to exhibit varying phrase curve slopes and ranges and varying accent curve amplitudes. The phonemic units selected from the acoustic corpus can be warped in the sinusoidal framework to display varying overall energy and spectral tilt profiles using the four-band representation.

6. References

[1] R. Banse and K. Scherer, "Acoustic Profiles in Vocal Emotion Expression", In Journal of Personality and Social Psychology, 70(3), pp. 614-636, 1996.

[2] P. Boersma and D. Weenink, "Praat: Doing phonetics by computer", [online http://www.fon.hum.uva.nl/praat/]

[3] J. Cahn, "Generating Expressions in Synthesized Speech", Master's Thesis, MIT, 1989.

[4] J. P. Hosom, "Automatic Time Alignment of Phonemes using Acoustic-Phonetic Information", PhD Thesis, Oregon Graduate Institute, Beaverton, OR, 2000.

[5] E. Klabbers and J. van Santen, "Control and prediction of the impact of pitch modification on synthetic speech quality", In Proceedings of EUROSPEECH'03, Geneva, Switzerland, pp. 317-320, 2003.

[6] E. Klabbers and J. van Santen, "Expressive speech synthesis using multilevel unit selection (A)", In J. Acoust. Soc. Am. 120(5), pp. 3006, 2006.

[7] Q. Miao, X Niu, E. Klabbers, and J.P.H. van Santen, "Effects of Prosodic Factors on Spectral Balance: Analysis and Synthesis", Speech Prosody 2006, Dresden, Germany.

[8] T. Mishra, J.P.H. van Santen, and E. Klabbers, "Decomposition of Pitch Curves in the General Superpositional Intonation Model", Speech Prosody 2006, Dresden, Germany.

[9] S. Mozziconacci, "Speech Variability and Emotion: Production and Perception", PhD Thesis, Technical University Eindhoven, 1998.

[10] J. van Santen and B. Möbius, "A quantitative model of F_0 generation and alignment", In A. Botinis (ed.), Intonation: Analysis, Modeling, and Technology, pp. 269-288, Kluwer Academic Publishers, Netherlands, 1999.

[11] J. van Santen and X. Niu, "Prediction and Synthesis of Prosodic Effects on Spectral Balance of Vowels", 4th IEEE Workshop on Speech Synthesis, Santa Monica, CA, 2002.

[12] J. van Santen, T. Mishra, and E. Klabbers, "Estimating phrase curves in the general superpositional intonation model", In Proceedings of the ISCA Speech Synthesis Workshop, Pittsburgh, PA, 2004.

[13] J. van Santen, A. Kain, E. Klabbers, and T. Mishra "Synthesis of prosody using multi-level sequence units", Speech Communication, 46(3-4), pp. 365-375, 2005.

[14] K. Scherer, "Vocal communication of emotion: A review of research paradigms", Speech Communication, 40, pp. 227-256, 2003.

Calliphony: A real-time intonation controller for expressive speech synthesis

Sylvain Le Beux, Albert Rilliard, Christophe d'Alessandro

LIMSI-CNRS, BP 133, F-91403, Orsay, France

{slebeux, rilliard, cda}@limsi.fr

Abstract

Intonation synthesis using a hand-controlled interface is a new approach for effective synthesis of expressive prosody. A system for prosodic real time modification is described. The user is controlling prosody in real time by drawing contours on a graphic tablet while listening to the modified speech. This system, a pen controlled speech instrument, can be applied to text to speech synthesis along two lines. A first application is synthetic speech post-processing. The synthetic speech produced by a TTS system can be very effectively tuned by hands for expressive synthesis. A second application is data-base enrichment. Several prosodic styles can be applied to the sentences in the database without the need of recording new sentences. These two applications are sketched in the paper.

Index Terms: prosodic modeling, prosodic perception, gestures, prosodic synthesis

1. Introduction

As speech synthesizers attain acceptable intelligibility and naturalness, the problem of controlling prosodic nuances emerges. Expression is made of subtle variations (particularly prosodic variations) according to the context and to the situation. In daily life, vocal expressions of strong emotions like anger, fear or despair are rather the exception than the rule. Then a synthesis system should be able to deal with subtle continuous expressive variations rather than clear-cut emotions.

Expressive speech synthesis may be viewed from two sides: on the one hand is the question of expression specification (what is the suited expression in a particular situation?) and on the other hand is the question of expression realization (how is the specified expression actually implemented). The first problem (situation analysis and expression specification) is one of the most difficult problems for research in computational linguistics, because it involves deep understanding of the text and its context. In this paper, only the second problem is addressed. The goal is to modify speech synthesis in real time according to the gestures of a performer playing the role of a "speech conductor" [1]. The Speech Conductor adds expressivity to the speech flow using Text-to-Speech (TTS) synthesis, prosodic modification algorithms and gesture interpretation algorithms.

This work is based on the hypothesis that human expressivity can be described in terms of movements or gestures, performed through different media, e.g. prosodic, body or facial movements. This question is closely related to musical synthesis, a field where computer based interfaces are still subject of much interest and development [2]. It is not the case for speech synthesis, where only a few interfaces are available for controlling in real time expressivity of spoken utterances. Existing gesture-controlled interfaces for speech production are either dealing with singing synthesis (cf. [3], [4]) or with full speech synthesis [5], but with a sound quality level insufficient for expressivity generation.

In this paper a new system for real-time control of intonation is presented, together with application to text-to-speech synthesis. This system maps hand gestures to the prosodic parameters, and thus allows the user to control prosody in a cross-modal way. As a by-product, the cross-modal approach of prosody generation represents a new way to generate and describe prosody and may therefore shed a new light on the fields of prosody systems and prosody description.

The paper is organized as follows. The real-time intonation controller is described in Section 2. The performances of the controller for real-time intonation modification are evaluated in section 3. Applications to expressive text-to-Speech synthesis are sketched in section 4. Section 5 discusses the results obtained so far, proposed future work and gives some conclusions.

2. Real-time intonation controller

2.1. Principle

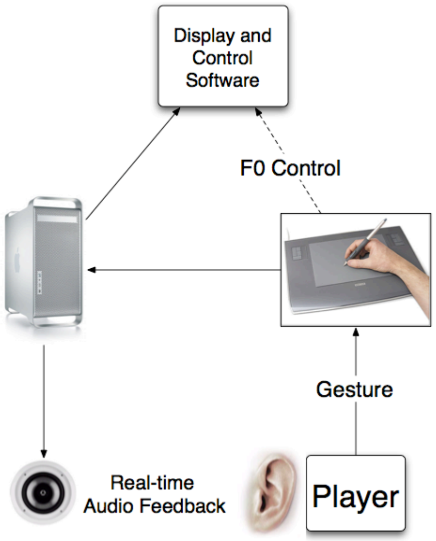


Figure 1: *Generic diagram of the system*

The real-time intonation controller operates in principle like a musical instrument. The loop between the player and the instrument is depicted in Figure 1. The player's hand movements are captured using an interface, and these movements are mapped on the input controls of the synthesizer. The sound is modified accordingly, played, and the player, who modifies his gestures as a function of the perceived and intended sounds, perceives this audio feedback.

2.2. Gesture interface: writing movements

Many devices, among which MIDI keyboard, Joystick and data glove; have been tested for capturing gestures with intonation control in mind.

Keyboard is not well fitted because it allows only discrete scales, although in speech a continuous control is mandatory. An additional pitch-bend wheel proved not very convenient from an ergonomic point of view.

As for the joystick and data glove, the precision in terms of position seemed insufficient: it proved too difficult to reach accurately a given target pitch. Such devices seem better suited for giving directions (as in a flight simulator) than precise values.

The graphic tablet has been chosen because it presents a number of advantages: its sampling frequency is high (200 Hz) and its resolution in terms of spatial position is sufficient for fine-grained parameter control (5080 dots per inches). Moreover, all the users are trained in writing since childhood, and are 'naturally" very much skilled in pen position control. Scripture, like speech, is made of a linguistic content and a paralinguistic, expressive content (in this case called "calligraphy"). There is a remarkable analogy between pitch contour and scripture. This analogy between drawing and intonation is very effective and intuitive from a performance point of view. Untrained subjects proved to be surprisingly skilled for playing with intonation using the pen on the graphic tablet, even at the first trial. For intonation control, only one axis of the tablet is necessary. The vertical dimension (Y-axis) is mapped on the F0 scale, expressed in semi-tones. The x-scale is not used: it means that very different gestures can be used for realizing a same intonation pattern: some players were drawing circle-like movements, when others preferred vertical lines or drawing similar to pitch contours. The second spatial dimension of the tablet will be used later for duration control in a second stage. Other degrees of freedom are still left in the tablet (pressure, switch) and will be use for controlling additional parameters, e.g. parameters related to voice quality.

Taking these observations into account, we decided to opt for a Wacom graphic Tablet, A4 size and we based our platform on a Power PPC Apple G5 Mac, 2.3 GHz bi-processor.

2.3. Real-time software

Real-time processing of information is a key point of the Calliphony system: as the user adapts his hand movement to perceived pitch at the output of the system, the delay has to remain inaudible. Calliphony is elaborated under the Max/MSP[1] software ([6], [7]), which is a graphical development environment intended to processes sound in real-time and which has already proven several years of reliable experience in real-time sound processing. Concerning the modification of speech pitch, we used a TD-PSOLA [8] Pitch-Shifter external provided by Tristan Jehan for Max/MSP environment [9].

As described on Figure 2, Calliphony takes as inputs the Y-axis position of the pen on the graphic tablet, and a recorded sound to be modified. It then maps the pitch value of the sound output to a value corresponding to the Y-axis value. This mapping is done on a logarithmic scale, such as the metric distance of each octave is the same. This corresponds analogously to the perception of the pitch by the human ear.

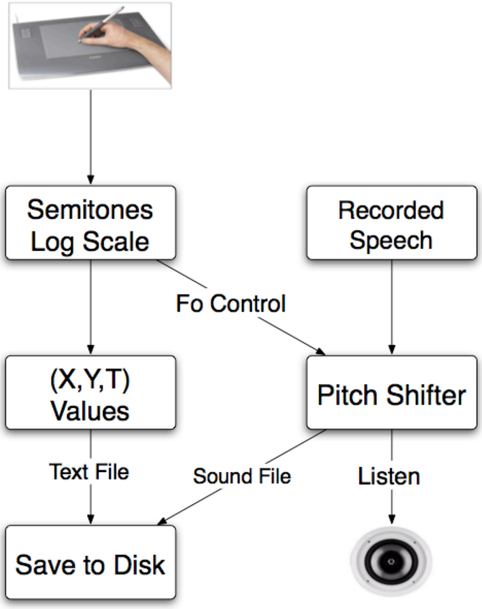


Figure 2: *"Calliphony" system description*

3. Evaluation of the controller

The use of handwriting movement to control pitch is not a priori straightforward. An evaluation procedure has therefore been developed, in order to assess the ability of a human to perform real-time control of speech prosody. The principle of this evaluation procedure is to measure the ability of the Calliphony player to imitate as closely as possible the prosody of an original sentence. The handwriting imitation performances are compared to the oral ability of the same user to imitate the same sentences. This work is described in more detail in a companion paper (cf. [10])

3.1. Prosodic imitation interface

A specific interface (cf. fig. 3) was developed to allow the subjects of the experiment to easily perform their imitation task. This interface encapsulate the Calliphony system, so that the user can listen to an original sentence, and then imitate the prosody both on a F0 flattened version of the sentence and vocally by recording his own voice.

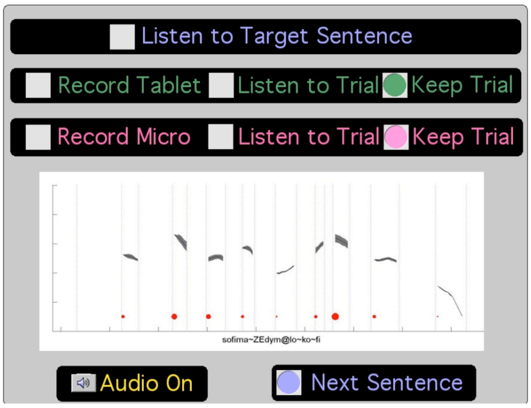

□ Listen to Target Sentence
□ Record Tablet □ Listen to Trial ● Keep Trial
□ Record Micro □ Listen to Trial ● Keep Trial
sofima~ZEdym@lo~ko~fi
Audio On
● Next Sentence

Figure 3: interface used for the handwriting imitation of prosody. Buttons allow to listen to the original sentence, record its own speech or the graphic tablet, listen to a recorded performance and save it. The current sentence's F0 is displayed.

[1] It is noticeable however that Max/MSP software is not multithreaded and consequently did not allows taking full advantage of the multi-processors architectures.

Table 1: *The 18 sentences of the corpus, from 1 to 9-syllable length.*

Nb syllable	Sentence	Phonetic	Sentence	Phonetic
1	Non.	[nɔ̃]	L’eau	[lo]
2	Salut	[saly]	J’y vais.	[ʒi vɛ]
3	Répétons.	[ʁepetɔ̃]	Nous chantons.	[nu ʃɑ̃tɔ̃]
4	Marie chantait.	[maʁɪ ʃɑ̃tɛ]	Vous rigolez.	[vu ʁigole]
5	Marie s’ennuyait.	[maʁɪ sɑ̃nɥijɛ]	Nous voulons manger.	[nu vulɔ̃ mɑ̃ʒe]
6	Marie chantait souvent.	[maʁɪ ʃɑ̃tɛ suvɑ̃]	Nicolas revenait.	[nikola ʁəvənɛ]
7	Nous voulons manger le soir.	[nu vulɔ̃ mɑ̃ʒe lə swaʁ]	Nicolas revenait souvent.	[nikola ʁəvənɛ suvɑ̃]
8	Sophie mangeait des fruits confits.	[sofi mɑ̃ʒɛ de fʁɥi kɔ̃fi]	Nicolas lisait le journal.	[nikola lizɛ lə ʒuʁnal]
9	Sophie mangeait du melon confit.	[sofi mɑ̃ʒɛ dy məlɔ̃ kɔ̃fi]	Nous regardons un joli tableau.	[nu ʁəgaʁdɔ̃ ɛ̃ ʒoli tablo]

As the aim of the evaluation is to investigate how close to the original the imitation can be, subjects are able to listen to the original sound when they need to, and to perform imitation until they are satisfied. Several performances can be recorded for each original sound.

3.2. Evaluation paradigm

3.2.1. Corpus

The evaluation procedure is based on a dedicated corpus constructed on 18 sentences, ranging from 1 to 9 syllables length (cf. table 1). Each sentence was recorded in its lexicalized version, and also in a reiterant delexicalized version, replacing each syllable by the same /ma/ syllable. Constraints on the corpus construction were: the use of CV syllable structure and absence of plosive consonant at the beginning of each word. Such constraints aimed at obtaining easily comparable prosodic patterns amongst the sentences and at avoiding important micro-prosodic effects due to plosive bursts.

Two native speakers of French recorded the corpus (a female and a male), according to three consigns: (1) to perform a declarative prosody, (2) to make an emphasis on one specific word of each sentence (generally on the verb) and (3) to perform an interrogative prosody. This results in 108 sentences, directly digitalized on a computer (41kHz, 16bits) for each speaker, using an USBPre sound device connected to an omni directional AKG C414B microphone placed 40 cm from the speaker mouth, and performing a high-pass filtering of frequency under 40Hz plus a noise reduction of 6dB.

3.2.2. Calliphony players

4 users have completed the experiment on a subset of 9 sentences ranging from 1 to 9 syllables, either lexicalized or reiterated, and using the three prosodic conditions (declarative, emphasized, interrogative), for the male speaker. All subjects are involved in this work and completely aware of its aims and are therefore familiar with prosody. Three out of the four subjects are trained musicians. One of the four subjects is the male speaker of the original corpus, who has therefore imitated its own voice vocally and by handwriting movements.

3.2.3. Prosodic parameters and distances measures

In order to evaluate the objective distance between the original and the imitated sentences, their pitch values have to be carefully extracted and computed. All the sentences of the corpus were manually analyzed. Their prosodic parameters were automatically extracted: fundamental frequency for vocalic segments (in semitones) and the corresponding voicing strength (calculated from intensity), syllabic duration and intensity thanks to Matlab (the yin script [11]) and Praat [12] programs.

The objectives distances between the prosody of the original sentence and the imitated prosody were calculated on the basis of the physical dissimilarity measures introduces by Hermes [13]: the correlation between the two F0 curves, and the root-mean-square (RMS) difference between theses two curves. The voicing strength was used (as suggested by [13]) as a weighting factor in the calculation of these two distances measures.

Objective distances between the original sentence and each repetition at the output of the Calliphony system were automatically calculated by using 10 ms spaced vector of F0 values for each vocalic segment. Then only the closest imitation, according to the weighted correlation measure and then the weighted RMS distance, was kept for the result analysis. This part of the work can be completely automated, as there is no duration change between the output of Calliphony and the original sentence. This is not the case for the oral imitations, which have to be labeled prior to extract F0 values for vocalic segments.

Moreover the distance computation supposes segments of the same length, a condition not met for vocal imitations. Therefore, only the distances between the original sentences and the gestural imitations have been calculated so far.

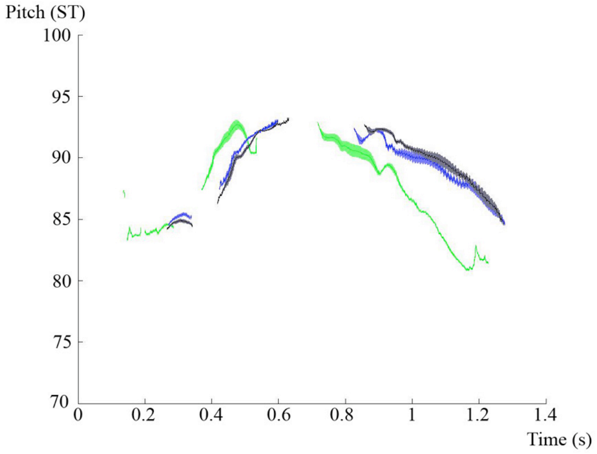


Figure 4: raw F0 value (in tones) for an original sentence (gray) and the two vocal imitations of one subject. Stimuli are not time-aligned.

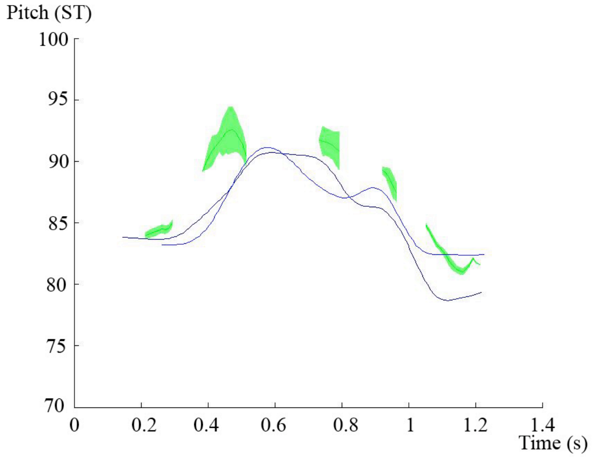


Figure 5: stylized F0 of an original sentence (the same as in fig. 4 – gray curve, smoothed values for the vocalic segment expressed in tones), and the value of the pitch parameter controlled by the graphic tablet for all the imitations performed by one subject. Stimuli are time-aligned.

Graphics with the raw F0 value of both the original and the vocal imitations have been produced in order to visually compare the performances of gesture vs. vocal imitations. Graphics with the stylized F0 of the original sentences (smoothed F0 for the vocalic segments) superimposed with the course of the pen on the graphic tablet were also produced in order to compare the two imitations' modalities (fig. 4 & 5).

3.3. Results

The mean objective distances are summarized in Table 2. There is no major difference between the four users, except for a higher RMS distance for AR, the only non-musician amongst the users (for a discussion about this issue cf. [10]).

Table 2: *mean distances for each subject and for all sentences imitated by handwriting movements.*

Subject	***R***	***RMS***
CDA	0.866	3.108
BD	0.900	3.079
SLE	0.901	3.091
AR	0.898	4.728
Total	0.891	3.502

The prosodic condition (declarative, emphasized, interrogative prosody) did not have a significant impact on the users' performances. The reiterant speech condition neither did.

The most influential factor in the experiment is the length of the sentence, as correlations continuously decrease while the number of syllable increase (cf. figure 6). This result can be explained either by an increasing difficulty of the user's task, or by an artifact due to the sentence length, because computation of correlation does not take into account any weighting for length compensation. More analyses would be needed before concluding on a sentence length effect.

Finally, the most important result of this evaluation procedure is the high overall correlation and low RMS distance obtained by all users. This result generally validates the ability of human users to imitate very closely an original prosody by using handwriting movements. Moreover, the observation of the imitated F0 curves shows a complete smoothing of any micro-prosodic variations: this indicates that users only reproduce prosodic movement at the level of the syllable or above, and that the task adequately matches prosody imitation and generation purposes.

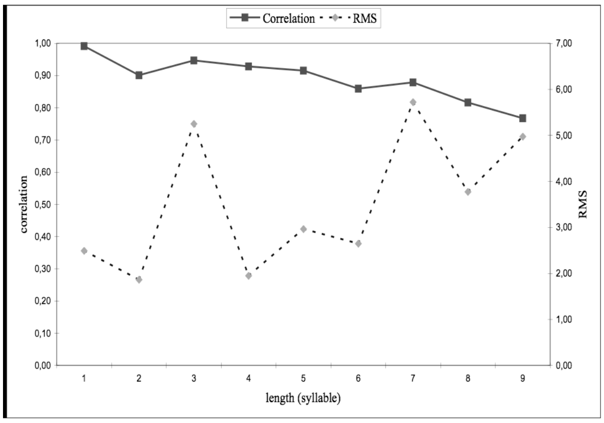


Figure 6: evolution of the two distances measures with the sentence's length. X-axis: length of stimuli, left Y-axis: correlations (plain line), right Y-axis: RMS difference (dotted line).

4. Application to expressive speech synthesis

Since the adequacy of a hand-driven interface to control speech prosody is validated, this section will explore some possible applications of this interface.

4.1. Intonation post-processing

A first application of the Calliphony system is directly derived from the scheme developed for the evaluation of the system: to allow a user to directly change the pitch of a spoken utterance. Such application can be useful in the field of speech synthesizers: as such devices have already reached a high degree of naturalness, they are now seeking for expressivity. The major problem is then to record and adequately model the huge corpora needed to be able to face any kind of expressivity for any sentences.

Our proposal is to give the end user the possibility to directly add the expressivity he needs on the output of his speech synthesizer thanks to the Calliphony system. This system is easy to use and only need little practice. Someone could then easily add e.g. a focalization on a desired word.

4.1.1. Assessment procedure

In order to assess the ability of our system to add such kind of expressivity to synthetic speech, a validation procedure has been set up, and is reported hereafter. It is based on exactly the same principle as the validation of the Calliphony system for prosody imitation reported above, with the only difference being that flattened speech (the input of the Calliphony system) has been replaced here with a synthetic sentence, produced with the Selimsi TTS system [14]. The player of Calliphony hears an original sentence from our corpus, carrying either a focalization on one word or an interrogative prosody. He has then to reproduce the pitch contour of the original sentence on the synthetic sentence, on a similar task that the one described above.

The major difference between the two experiments concerns the segmental duration of the modified stimuli: for the preceding evaluation, the segmental durations are exactly the same as the original, as it is only a flattened version of the natural stimulus, whereas the synthetic sentence has his proper segments' durations. It induces two major differences between the two protocols. The first one concerns the modification procedure: it is harder to perform an imitation when important lengthening is present in the original sentence. The second one concerns the distance measure between the original and the reproduced pitch contours. As the pitch values are compared for vowel only, and as synthetic and natural vowels don't necessarily have the same duration, instead of extracting one value of pitch for each 10 ms, 10 values for each vowel were calculated, regularly spaced along the vowel. These 10 values per vowel are then used to calculate correlation and RMS distance using the same formulae as those presented above.

Table 3: mean distance scores obtained for focalized sentence, interrogative sentences and for all sentences.

	Correl	RMS
Focalization	0,92	3,18
Interrog.	0,86	4,14
Mean	0,89	3,66

4.1.2. Results and analyses

The results obtained for this assessment are quite similar to those already exposed.

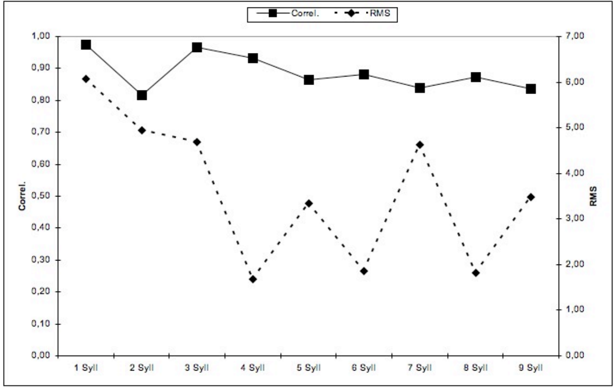


Figure 7: mean distances (correlation and RMS distance) obtained for sentences of all length, from 1 to 9 syllables.

Mean correlation and RMS distances are good (cf. tab 3), and indicate a close stylization of the pitch curve on the synthetic stimuli, even if there are durational differences. Mean score obtained for focalization vs. interrogation sentences are quite similar, with slightly better score for focalization. About the effect of the sentence's length (cf. fig. 7), the effect is a bit more complicated than the one observed with natural speech: if correlation decreases gradually with the sentence length, as it has already been observed, the RMS distance did not have any particular tendency, except for the 1-, 2- and 3-syllables length's sentences, that receive high RMS distances scores, contrary to natural speech.

The objective distance between modified prosodic parameters at the output of Calliphony and the original natural prosody is rather small, giving a very good idea of the system's performances at producing expressive speech.

However, it must be noted that the duration parameter is not dealt with in this first version of Calliphony. This is not satisfactory for high quality expressive synthesis, where durations' modification is mandatory. In addition, the sound quality is better for natural speech modification compared to synthetic speech modification. In our current implementation Calliphony results in two successive modifications of the signal (concatenation and PSOLA modification), a situation that is not optimal indeed. More work is still needed before to obtain a better sounding system, but we think that the ability of players to add expressivity to synthesizers has been convincingly demonstrated.

Considering the databases that are not previously tagged, the system can still be used online in a slightly different manner. When the purpose is only to produce some expressive sentences (for various perceptive experiments for example) then one is able to modify online the synthesized sentences and to record them directly after modification.

This gives the opportunity for someone not necessarily familiar with speech synthesis and processing, to produce expressive sentences in a convenient manner, without having to buy an expensive system or to acquire deep knowledge in speech processing. Moreover, one can use synthesized sentences from any TTS engine publicly available or can directly record sentences on its owns with a simple microphone and recording software, before achieving its modification thanks to our system.

As an extent, thanks to the good quality of expressive modifications on synthetic speech, it could find applications in various research and development situations, going from advertising to industrial mass media, or even animated cartoon characters voice synthesis.

4.2. Data-base enrichment

Another application of the Calliphony system to TTS concerns specifically data-driven speech synthesis. Synthesis systems based on selection/concatenation of non-uniform units need large corpora of recorded speech. Our system can be used for enrichment of the speech database, prior to synthesis. In this case, natural speech is modified, and a same sentence can be given several prosodic variations, as depicted in Fig.8

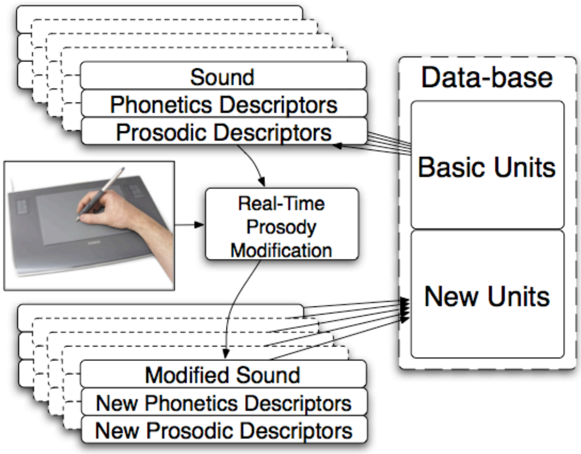


Figure 8: Enrichment of Data-Base with Non-Uniform Units

There are several steps to achieve this enrichment and it can be applied to various types of databases. There are no constraints on the content of the database. The system can then be used to add new expressions that were not recorded, or to have more utterances of a less represented expression.

Then the prosodic content of the database can be extended and/or improved without the need of new recordings. This is independent of the TTS system itself, because it is only a matter of database pre processing.

This application is in a preliminary stage: no formal evaluation of the synthetic speech obtained is available for the moment.

5. Discussion and conclusion

Speech instruments have been an important part of the history of speech synthesis, but have played only a marginal role in speech synthesis application or research. We think that high quality real-time speech modification algorithms and new high precision interfaces have the potential for dramatically changing the current situation.

In this paper, we explore the ability of handwriting movement for expressive speech synthesis. The system has been called "calliphony", i.e. expressive speech beyond phonemes by analogy with "calligraphy", i.e. expressive writing beyond graphemes. The results indicate even untrained players are almost as skilled for vocal imitation as for written imitation of expressive prosody.

Then, the system can be applied to TTS post processing and database pre-processing. TTS post-processing can be a useful extension of a TTS system for tuning synthetic speech utterance output without the need of deep engineering or expensive recordings. The quality obtained is basically the quality of the TTS system itself.

We are currently exploring the quality reached by database enrichment, a pre-processing for augmenting the prosodic content of a selection/concatenation TTS system, without recording new sentences.

Future work will be devoted to duration and tempo modifications. Our experiments show (or confirm) that changing intonation without changing duration or tempo is not enough in many situations. Changing voice quality is also required for more realistic prosodic modifications. Additional control parameters will then be needed.

Another path of research for future work is the interface itself. We are currently pursuing the study of the range of possibility offered by an ad-hoc controller called the Meta-Instrument. This controller offers up to 54 continuous controllers simultaneously, supervised by the fingers and the arms (see [15]).

6. References

[1] D'Alessandro, C., et al. (2005) "*The speech conductor : gestural control of speech synthesis*." in eNTERFACE 2005. The SIMILAR NoE Summer Workshop on Multimodal Interfaces, Mons, Belgium.

[2] http://www.nime.org/

[3] Cook, P., (2005) *"Real-Time Performance Controllers for Synthesized Singing,*. Proc. NIME Conference, 236Ð237, Vancouver, Canada.

[4] Kessous, L. , (2004) *"Gestural Control of Singing Voice, a Musical Instrument"*. Proc. of Sound and Music Computing, Paris.

[5] Fels, S. & Hinton, G. , (1998) *"Glove-Talk II: A Neural Network Interface which Maps Gestures to Parallel Formant Speech Synthesizer Controls."* IEEE Transactions on Neural Networks, 9 (1), 205Ð212.

[6] Puckette, M. (1991). *"Combining Event and Signal Processing in the MAX Graphical Programming Environment"*. Computer Music Journal 15(3): 68-77.

[7] http://www.cycling74.com/

[8] E. Moulines and F. Charpentier, (1990) *"Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones,"* Speech Communication, vol. 9, pp. 453–467.

[9] http://web.media.mit.edu/tristan/

[10] d'Alessandro, C., Rilliard, A. & Le Beux, S. (Submitted). *"Computerized chironomy: evaluation of hand-controlled intonation reiteration."* Proc. of InterSpeech 2007.

[11] de Cheveigné, A., Kawahara, H., (2002) *"YIN, a fundamental frequency estimator for speech and music"*, J. Acoust. Soc. Am. 111, 1917-1930.

[12] Paul Boersma & David Weenink, (2001)*"PRAAT, a system for doing phonetics by computer."* Glot International 5(9/10): 341-345.

[13] Hermes, D.J. (1998). *"Measuring the Perceptual Similarity of Pitch Contours"*. Journal of Speech, Language, and Hearing Research, 41, 73-82.

[14] Prudon, R. and C. d'Alessandro. (2001) "*A selection/concatenation text-to-speech synthesis system: databases development, system design, comparative evaluation*." in 4th ISCA/IEEE International Workshop on Speech Synthesis.

[15] Serge de Laubier, Vincent Goudard, (2006) *"Méta-Instrument 3 : a look over 17 years of practice",* in Proc. of the NIME Conference, IRCAM, Paris, France .

Epoch Synchronous Non Overlap Add (ESNOLA) Method based Concatenative Speech Synthesis System for Bangla

Shyamal Kumar Das Mandd, Asoke Kumar Datta

Centre for development of Advanced Computing (C-DAC), Kolkata, India
shyamal.dasmandal@kolkatacdac.in

Abstract

In the last decade there has been a shift towards development of speech synthesizer using concatenative synthesis technique instead of parametric synthesis. There are a number of different methodologies for concatenative synthesis like TDPSOLA, PSOLA, and MBROLA. This paper, describes a concatenative speech synthesis system based on Epoch Synchronous Non Over Lapp Add (ESNOLA) technique, for standard colloquial Bengali, which uses the partnemes as the smallest signal units for concatenation. The system provided full control for prosody and intonation.

1. Introduction

The most common mode of human communication is the oral mode. We are naturally conversant in communicating with other human beings in speech mode. In Indian languages machine to man communication has attained a reasonable level for direct practical application. Speech synthesizers in Indian languages are beginning to appear.

Speech synthesis is the process, which allows the transformation of a string of phonetic and prosodic symbols into a synthetic speech signal. The quality of the result is a function of the quality of the string, as well as of the quality of the generation process itself.

The first requirement of a text-to-speech (TTS) system is intelligibility and the second one is the naturalness. Actually the concept of naturalness is not to restitute the reality but to suggest it. Thus, listening to a synthetic voice must allow the listener to attribute this voice to some pseudo-speaker and to perceive some kind of expressivities as well as some indices characterizing the speaking style and the particular situation of elocution [1]. For this purpose the corresponding supra-segmental information must be supplied to the system [2].

Most of the present TTS systems produce an acceptable level of intelligibility, but the naturalness dimension, the ability to control expressivities, speech style and pseudo-speaker identity still are poorly mastered. The user's demands vary to a large extent according to the field of application: general public applications such as telephonic information retrieval need maximal realism and naturalness, whereas some applications involving professionals (process or vehicle control) or highly motivated persons (visually impaired, applications in hostile environments) demand intelligibility with the highest priority.

In the last decade there has been a significant trend for development of speech synthesizers using Concatenative based Synthesis techniques. This method of speech synthesis is one of the most successful approaches for synthesizing speech, which uses pre-recorded speech units for building the utterances. There are a number of different methodologies for Concatenative Synthesis like TDPSOLA, PSOLA, MBROLA and Epoch Synchronous Non Over Lapp Add (ESNOLA).

In the review by Klatt (1987) some of the early efforts on concatenative synthesis are included. Much earlier Peterson et al (1958) suggested that unit concatenation might be a possible solution for speech synthesis. Dixon and Maxey (1968) made a special effort to create a unit library for di-phone synthesis. Early synthesis research at AT&T based on "Diadic Units" (Olive, 1977) demonstrated an alternative to rule-based formant synthesis (Carlson and Granström, 1976, Carlson et al, 1982 and Klatt, 1982). Charpentier and Stella (1986) opened a new path towards speech synthesis based on waveform concatenation, by introducing the PSOLA model for manipulating pre-recorded waveforms. The current methods of using unit selection from large corpora, rather than using a fixed unit inventory to try to reduce the number of units in each utterance and solve context dependencies over a longer time frame, is gaining ground. Möbius (2000) gave an extensive review of corpus-based synthesis methods. In automatic unit selection method issues are mostly related to estimating target costs that match the perception of a human listener, so that the units chosen by the system are the best in terms of perceived speech quality. What is more, quality, when it is available, is still achieved at the expense of storage requirements (AT&T's system requires several hours of speech, i.e., several hundreds of Mbytes of speech data) and computational complexity (Speech Work's system won't work on your favorite PC, laptop or palmtop; users buy the right to run it on a server via the internet). This currently makes these systems unusable for low-cost general-purpose electronic devices.

This paper presents a new Concatenative text-to-speech (TTS) system for Standard Colloquial Bengali (SCB) using a new set of signal units in sub-phonemic level, namely, partnemes.

The Epoch Synchronous Non Overlap Add (ESNOLA) algorithm is developed for concatenation, regeneration as well as for pitch and duration (prosodic) modification. It may be noted that the prosody of the stored units is often not consistent with that of the target utterance and must be altered at the time of synthesis. Furthermore, several types of mismatches can occur at unit boundaries of the synthesized signal, which have to be properly truncated and matched. ESNOLA technique provides the complete control on implementation of intonation and prosody [3]. It allows judicial selection of signal segments so that smaller fundamental parts of the phonemes may be used as units reducing both number and size of the signal elements in the dictionary. Further the methodology of concatenation provides adequate processing for proper matching between different segments during concatenation [4][5]. The use of special type of basic signal segments makes the size of signal dictionary very small so there is a possibility of its implementation in low-cost general-purpose electronic devices.

2. Basic Working Principle of the Proposed Synthesizer

Figure 1 represents the basic block diagram of TTS System using ESNOLA Technique

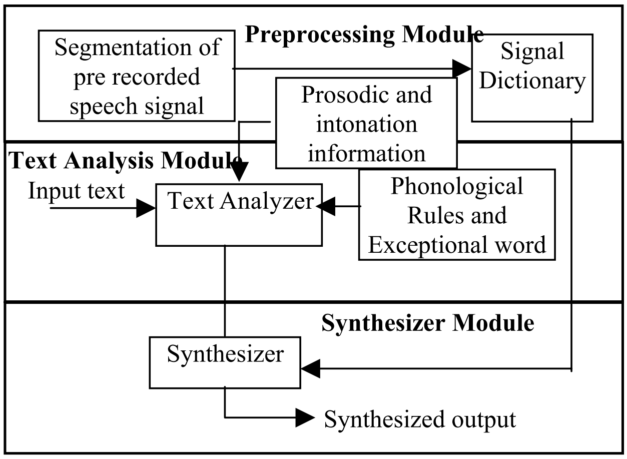


Figure 1: Basic Block Diagram of TTS System using ESNOLA Technique

The above block diagram (Figure 1) describes the basic part of the ESNOLA technique for the development of text-to speech synthesis system. It consists of three parts a) Preprocessing module, b) Text analysis module and c) Synthesizer module.

2.1. Preprocessing module

In this module the required speech segment (called pratnemes) database is created from the pre-recorded natural speech signal. The advantage of using partnemes as the basic unit is the simplicity of introducing intonation and prosodic rules into the synthesized speech signals. For the building of pratneme dictionary the following steps are required.

Step1: Creation of nonsense word set

This set of words must contain acoustic phonetic characteristics of all phonemes. A set of tetra-syllabic nonsense words of the forms CVCVCVCV, CVVCVVCVVCVVC is used for normal consonants and vowels. However, as /n/-/ŋ/ distinction in case of Bangla is not ascertained except in conjunction with appropriate consonant, an additional 8 syllabic form CVNVCVNVCVNVCVNV for two nasals /n/ and /ŋ/ are included. The choice of tetra syllabic words in case of Bangla is necessary because Bangla being a bound stress language with stress occurring normally at the first syllable. Another set of words has to be collected from the normal lexicon of the language where the different vowel-vowel combinations occur. Usually all possible combination may not be easily available. For the unusual combinations appropriate sentences are created where such combination occur at word juncture.

Step2: Recording

A professional speaker with good voice quality is used to utter the aforesaid set of words in a noise free environment. The utterances should be devoid of emphasis. Care is taken to ensure that the pitch of the recorded word remains almost same throughout the recording. Recording format is 16-bit PCM, mono, sampling frequency being 22050Hz.

Step3.Pitch Normalization

All signal segments is brought to exactly same fundamental frequency. This is necessary to avoid pitch-mismatch. However adjustment of pitch by manipulation of sampling frequency may be used only when the pitch difference does not exceed 10% of the original value.

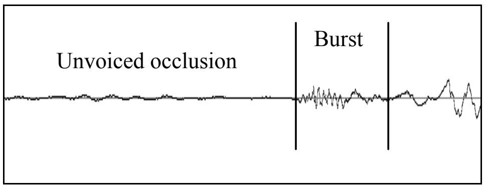
Step4. Amplitude Normalization

Amplitude normalization is performed with respect to the intrinsic amplitude of vowels. It is known that the vowels of equal amplitude do not sound equally loud. The amplitude of all the CV, VC and VV segments are normalized with respect to their associated vowel's intrinsic amplitude.

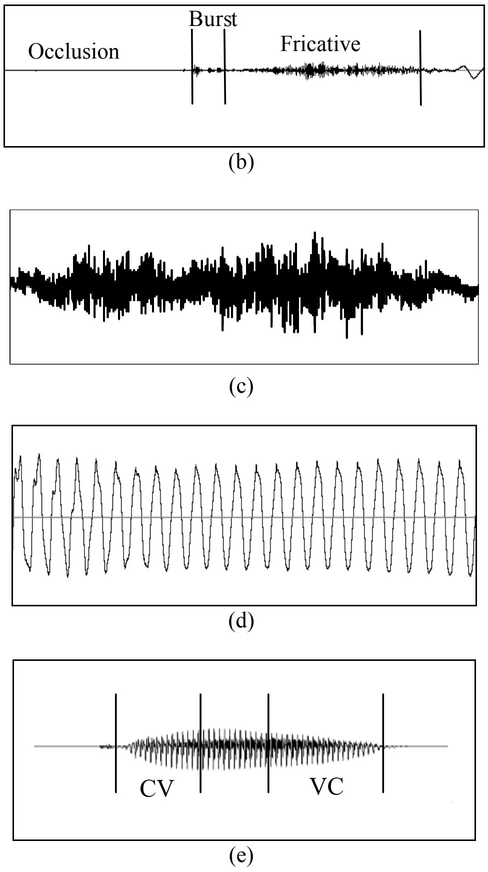
Step5. Segmentation

A set of basic speech units called partnemes (i.e. part of a phoneme) is used here. Partnemes include identifiable portions unique for phonemes as well as the segments representing co-articulation. The set of partnemes is divided into two sub-groups. The first group consists of the segments of occlusion or voice-bar along with the plosion or affrication, sibilants, nasal murmurs, laterals, semivowels and diphthongs. The second group has all CV, VC, and VV co-articulatory regions. It may be noticed that though VOT (Voice Onset Time) is an integral part of the plosives and affricates, it is not included in the consonantal parts for these phonemes. This is because during the VOT strong co-articulatory influences of the succeeding vowels are manifested in terms of aperiodic transitions.

Figure2 shows how partnemes are extracted from the VCV segments. For plosives the partneme consists of occlusion and burst (C) and for affricate the friction after the plosion is also included. The co-articulation between the vowel and consonant include the voice-onset-time (VOT) and the consonant vowel transition (CV). Vowel to consonant transition begins at the end of the steady state of the vowels up to the beginning of the occlusion of the next consonant or any other consonant marker.



(a)



(b)

(c)

(d)

(e)

Figure 2: Example pratneme of a) plosive b) affricate c) sibilant d) nasal murmur e) vocalic transition CV and VC

2.2. Text analysis module

The Text analysis module is the front-end language processor of the Text-to-Speech System, which accepts input text and generates corresponding phoneme string and stress markers. On many occasions the Text Analyzer consists of a natural language processing module (NLP), capable of producing a phonetic transcription of the text read, together with the desired intonation and rhythm (often termed as prosody).
The text analysis module has two broad sections one is the phonological analysis module and other is the analysis of the text for prosody and intonation. Bangla has a syllabic script. Grapheme-to-phoneme (phonological analysis) conversion is a formidable problem [6][7]. The phonological problems are mainly found in the pronunciation of the two vowels /a/ and /e/ as well as a number of consonant clusters. Even the semantic and the parts of speech of a word sometimes play a significant role in pronunciation. A comprehensive set of phonological rules including the exceptions is developed and implemented [8].
The naturalness of the synthesized speech out put depends on the suprasegmental feature (prosodic and intonation feature) of the speech signal mainly pitch variation, syllabic duration variation, amplitude variation and pause. The implementation of the variation of suprasegmental feature in synthesized speech depend on the two factor one generation of intonation and prosodic rule [8] along with the development of text parser for intonation and prosodic marking and the implementation of the suprasegmental feature variation in the synthesizer. The later part will only be discussed in this paper.

2.3. Synthesizer module

It is the task of the Synthesizer module to combine splices of pre-recorded speech and generate the synthesized voice output. A sequence of segments is first deduced from the phonemic input of the synthesizer. If required, the prosodic events may be assigned to individual segments based on the information extracted by the text analysis module.
The Synthesizer Module functions in the following way:
The Phoneme string input from the Text Analyzer is assigned tokens, based on the indexing of the segmented partneme voice signals. Modification of pitch, amplitude and duration of the vowels has to be done to implement the prosody and intonation. The selected segments are concatenated to get the raw output signal. Spectral smoothing is performed on the concatenation points to remove mismatch and other spectral disturbances
Rules for Token generation:

CVCV → C +CV+V+VC+C+V+Vo
VCV → Vi+V +VC+C+CV+V+Vo
CVYV → C +CV+V+VY+YV+Vo
CVV→C+CV+VV+Vo

Where Vi, Vo, Vand C represent respectively fade-in vowel, fade-out vowel, Medial vowel and consonant. The fade in and fade out operation is applicable for the terminal vowels only. In non-terminal cases Vo and Vi are to be treated as V.
In ESNOLA approach, the synthesized out put is generated by concatenating the basic signal segments from the signal dictionary at epoch positions. The epochs are most important for signal units, which represent vocalic or quasi-periodic sounds. An epoch position is represented in Figure 3.

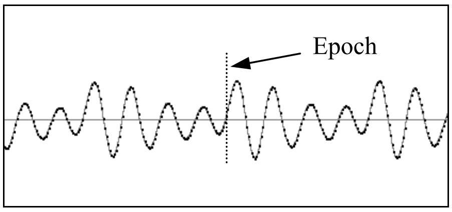


Figure 3: Epoch position of a speech segment

Steady states in the nucleus vowel segment of the synthesized signal are generated by the linear interpolation with appropriate weights of the last period and the first period respectively of the preceding and the succeeding segments. The generated signals require some smoothing at the point of concatenation. This is achieved by a proper windowing of the out put signal with out hampering the spectral quality. The equation of the window is as given below.

$$W(n) = \frac{1}{2}(1 - \cos(\pi * n / N)) \quad \text{for } 0<n<0.125N \qquad (1)$$

$$= 1 \qquad \text{for } 0.125N<n<0.625N$$

$$= \frac{1}{2}(1 - \cos(\pi * n / N)) \quad \text{for } 0.625N<n<N$$

Figure 4 represent a synthesized voiced out put for a given text input /b^{h}arɔt/

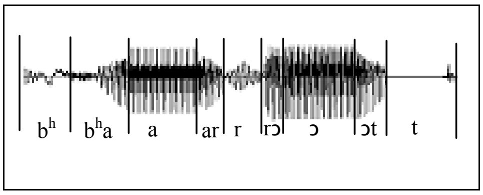


Figure 4: Synthesize out put for a given word /b^{h}ar□t/ using ESNOLA technique

2.3.1. Implementation of naturalness in synthesizer

Intensity modification (amplitude modification): this done by multiplying each of the sample value of the segment by the value specified by amplitude parameter of the corresponding token.

Duration modification: This operation in the present system is performed on steady state vowel segment. Length of the steady state of vowel segment depends on the syllable duration. It may be noted that the duration of consonants and the CV and VC transition are pre –specified.

F_0 modification: Pitch (F_0) modification of the synthesized signal is one of the important aspects to introduce intonation in the synthesized speech signal. In the segment dictionary the signal whose pitch have to be modified are the CV, VC, VV, nasal murmurs and laterals. Time scale pitch modification is done by changing the length of the period of the original signal.

In ESNOLA pitch (F_0) modification involves three steps. These are (1) Generation of short-time signals from original speech waveform, (2) Epoch synchronous modification brought to the short-term signals, and finally, (3) Synthesis by the concatenation of the modified signals. These three steps are described below.

2.3.1.1 Generation of Short-Time (ST) Signals

Let x(t) be the digitized speech waveform and let e_m: m = 1, 2, … represent the successive epoch positions in the signal. The intermediate representation of x(t) is a sequence of short-time (ST) signals $x_m^n(t)$, defined by

$$x_m^n = W_p(t)x(t - pT) \quad \text{for } 0<t<nT \text{ ....(2)}$$

Here, $W_p(t) = (1/\alpha)^{p-1}$ for positive integers p, n such that the value of p runs from 1 to n for each ST signal and α is an empirically chosen constant and it is greater than 0. T is the time interval between epoch positions e_{m-1} and e_m. In the equation 2, the value of p is 1 for the range $0 \le t < T$, the value of p is 2 for the range $T \le t < 2T$,… the value of p is n for the range $(n-1)T \le t < nT$ The physical implication of equation 2 is that the mth ST signal for the mth epoch points of the original signal constituted of n numbers of intermediate signals, constructed from the same Perceptual Pitch Period (PPP) in between (m-1)th and mth epoch points, but each time the amplitude is diminished by the factor $(1/\alpha)^{p-1}$ with increasing value of p. The length of the ST signal depends on the value of n.

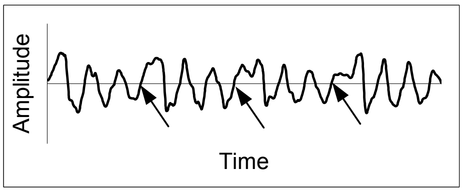


Figure 5: Epoch Positions Indicated by Arrows

The figure 5 shows the three consecutive epoch positions and let us denote the three as e1, e2, and e3 from left to right. Figure 6 shows the ST signal for the epoch e1 of the original signal. The ST signal is for n = 3 and α = 4. The ST signal constitute of three generated signal. The part of the signal, left to the left vertical line is for p = 1, that in between the two vertical line is for p = 2 and the right most one is for p = 3. It is to be noted that the number of generated ST signals is equal to the number of epoch points in the original signal.

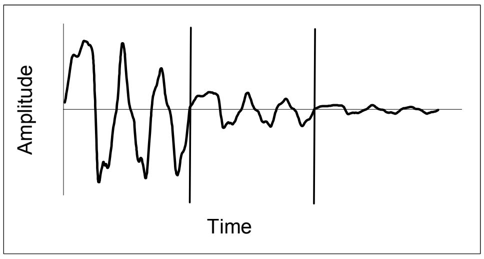


Figure 6: ST Signal for e1 in Figure 2.16 for n = 3 and α = 4

It is obvious that if α is chosen a large value, then the amplitude of the generated signals for p > 1 become negligibly small. The effect of it in the synthesized signal would be like that a glottal pulse is generated much after the dying down of the previous glottal pulse. This condition would create a creaky voice. Similar, if the value of α is much lower, then the effect of it in the synthesized signal would be like that a glottal pulse is generated much before the dying down of the previous one. Thus, this will create a breathy voice. Empirically the value of α is obtained 0.25 for the production of good synthesized output.

From this ST signal, the smallest pitch that can be generated is

$$f_m = \frac{1}{nT} \text{ ....(3)}$$

Each Short-Time signal is generated for the production of a single PPP of the synthesized speech signal. The value of n depends on the required pitch value of the synthesized signal. After generating the ST signal for a particular epoch points of the original signal, all the parameters are being reset and we shift to the next epoch point for the generation of the corresponding ST signal.

2.3.1.2 Epoch Synchronous Modification (ESM) of Short-Time signals

Epoch synchronous modification of $x_m^n(t)$ is described below. During pitch modification, the stream of Short-Time signals $x_m^n(t)$ is converted into modified stream of synthesized signals by placing a window appropriately and giving rise to a new set of epoch marks ${}_s e_m$. Let $\{{}_s e_m$: m = 1, 2, …} denote the epoch positions of the synthesized speech signal. The algorithm works out a mapping f: $\{e_m$: m = 1, 2, …} → $\{{}_s e_m$: m = 1, 2, …} between original and synthesized epoch marks such that the time difference between two consecutive epochs equals the corresponding synthesis pitch period. The modified stream of synthesized signals can be represented as:

$$x_m(t) = W_m^n(t) x_m^n(t) \quad \ldots (4)$$

In the above equation, the left side represents the synthesized speech signal for the mth ST-signal and $w_m^n(t)$ represents the window function for it. Note that this window is defined for every t less than or equal to the modified pitch period and it is zero beyond the pitch period. Selection of $w_m^n(t)$ and its consequence on ${}_s x_m(t)$ are described as in equation (2).

Now concatenating those changed pitch periods generate the required segment. This process creates a prominent striation and produces a perceptible mechanical horn like sound over and above the normal quality of the voice. This is because such concatenation produces exactly periodic wave instead of quasi-periodic ones. Normal human voice is not perfectly periodic. Two successive pitch cycles do not produce exactly the same pressure waves. The variations are random in nature and occur for pitch, amplitude and complexity, which are referred to as jitter, shimmer and complexity perturbations respectively. An optimum value of these produces natural sound. An excess of the perturbations makes the quality of sound rough or hoarse. Absence of these perturbations again produces an unnatural horn like sound. Addition of jitter and complexity perturbation almost removes the defect. A random variation of 2-3% in pitch period is introduced for jitter by introducing appropriate modification of T1. The complexity perturbation is introduced by randomly varying the sample value by ±1%.

3. Conclusions

In this paper, a system for concatenative speech synthesis has been described using ESNOLA technique. Partnemes are used as the smallest signal units in the paper. The theoretical analysis of the ESNOLA technique clearly shows its advantages in speech synthesis. The ESNOLA framework and partneme inventories altogether give a simple approach for the production of high quality synthesized speech, particularly useful for intonated concatenative synthesis system. Using only the epoch information of the voiced speech signal, the pitch and prosody can be manipulated by keeping the quality intact. The attractiveness of the present approach is its computational simplicity for pitch and duration manipulations. For prosody modification, it is also necessary to manipulate the pitch and duration in the CV, VC, murmur and laterals portions of the stored signals. The epoch detection algorithm is necessary for manipulating pitch and duration in these cases. But this can be avoided by an offline detection of the epochs and storing them in files.

Implementation of natural prosody and intonation need comprehensive rule for the spoken dialect. Unfortunately this is no yet available for SCB. Therefore system for flat speech using the technique has been developed for use. This is in the net where one cane hears the news from a Bangla daily newspaper, which is available in the net. Recently this system was used by the Election Communication for announcement of election results held in West Bengal.

4. References

[1] Deketelaere S., Deroo O., Dutoit T., “Speech Processing for Communications: What’s New?”(*) MULTITEL ASBL, 1 Copernic Ave, Initialis Scientific Park, B-7000 MONS(**) Faculté Polytechnique de Mons, TCTS Lab, 1 Copernic Ave, Initialis Scientific Park, B-7000 MONS

[2] Dan, T. K., Mukherjee B & Datta A. K. (1993). “Temporal approach for synthesis of singing (Soprano1).” *SMAC 93*, pp. 282-287, 1993.

[3] Datta A.K, Ganguli N.R and Mukherjee B. “ Intonation in segment concatenated speech” *Proc. ESCA Workshop on speech synthesis, France*, pp.-153-156, Sep 1990.

[4] Low P.H., Vaseghi S., “Synthesis Of Unseen Context And Spectral And Pitch Contour Smoothing In Concatenated Text To Speech Synthesis”, *ICASSP, Florida, USA*, Vol. 1, pp. 469-472, 2002.

[5] Das Mandal S. K, Datta A.K, Gupta B. “Spectral Matching of Epoch Synchronous Non-Over Lapping Add (ESNPLA) Method Based Concatenative Synthesizer”, *International Conference on Communications Devices and Intelligent System (CODIS-2004),* Jadavpur University, 2004, pp 729-732

[6] Chatterji Suniti Kumar “The Original and Development of the Bengali Language” *Published by Rupa.Co, 2002,* ISBN 81-7167-117-9, 1926.

[7] Sarkar Pabitra, “Bangla Balo” *Published by Prama prakasani, 1990.*

[8] Das Mandal Shyamal Kr, Saha Arup, Sarkar Indranil Datta Asoke Kumar, “Phonological, International & Prosodic Aspects of Concatenative Speech Synthesizer Development for Bangla,” *Proceeding of SIMPLE-05*, February 2005, pp56-60, 2005.

Syllable-based Thai Duration Model Using Multi-level Linear Regression and Syllable Accommodation

Chatchawarn Hansakunbuntheung[1]*, Hiroaki Kato*[2] *and Yoshinori Sagisaka*[1]

[1]GITI/Language and Speech Science Research Laboratory, Waseda University, Tokyo, Japan
[2]NICT/ATR Cognitive Infbrmation Science Labs, Kyoto, Japan
chatchawarnh@fuji.waseda.jp, kato@atr.jp, sagisaka@giti.waseda.ac.jp

Abstract

This paper proposes a syllable-based Thai duration model using multi-level linear regression and syllable accommodation. To build a timing model reflecting control characteristics directly, we introduce two analysis results on hierarchical control characteristics. First analysis result showed that syllable is highly correlated to higher-phone-level timing controls, while, phone differences by themselves do not affect higher control and contribute to local timing control only. Second one on the syllable accommodation showed that phone duration highly depends on local phone factors. These analysis results support a syllable-based hierarchical model proposed in this paper. Duration prediction experiments of 5-fold cross validation showed 46.73 and 32.37 ms in RMS error, and, 0.905 and 0.811 in correlation between measured and predicted duration at syllable and phone levels, respectively. The comparison of prediction precision showed that the proposed syllable-based multi-level duration model better performed than a conventional single-level phone duration model.

1. Introduction

To generate natural speech, an accurate duration model is required for assigning appropriate duration to each speech segment. We can find quite a few segmental duration models based on either phone or syllable in previous works [1]-[14]. In phone-based model, single phone was considered as a primary unit for prediction and all effects of duration control factors were calculated at one time at phone level [1]-[4]. This one-level calculation made it easy to globally optimize duration prediction errors.

While in syllable-based model, syllable was used as a primary unit [5]. In the syllable-based modeling, first, syllable duration was calculated and then phone duration was determined from syllable duration by accommodating its constituent phones using their mean durations and standard deviations showing intrinsic elasticity. In the previous work, the effects of constituent phones were not fully used in the first calculation of syllable duration. Only some generalized characteristics such as syllable complexities were employed instead of full combinations of constituent phones to reasonably reduce the control factors. It also has some drawbacks from computational viewpoints, First, its two-step calculation cannot be optimized at one time and finer mutual contribution between two levels gives duration errors as pointed out previously [7][8]. However, by separating control into inter-syllable level and intra-syllable level, this two-level calculation directly implemented control hierarchy and served for better understanding of the underlying timing control characteristics that could be applied to other non-stress timing duration characteristics [6].

In this paper, we propose a syllable-based duration model for Thai using multi-level linear regression, referred as multi-level model from now on, to predict syllable and phone duration. By linear property of linear regression (LR) method itself, it may not give the closest duration value. However, it can serve our purpose on observing underlying effects of desired control factors in the model. In Japanese duration modeling [4], it also shows that LR model can provide good prediction results. By using LR, two hierarchical LR models at syllable and phone levels are adopted for optimizing the prediction at both levels, and understanding of underlying timing controls at each level. In syllable-level model, constituent phones are also taken into account as mutual controls from phone level. In phone level model, only phone factors are used in modeling. In addition, syllable accommodation using absolute duration and duration ratio is also studied. In the following sections, the details of the multi-level duration model are depicted in Section 2. Then, duration control factors used in the model are described. Section 3 gives the details of speech data for experiments. In Section 4 and 5, we present the duration prediction experiments and results followed by discussions and conclusions.

2. Thai duration model using multi-level linear regression and syllable accommodation

2.1. Overview of the proposed duration model

The overview of the proposed duration model and control factors is illustrated in Figure 1. The proposed model is a syllable-based duration model with two LR sub-models at syllable and phone levels. The model considers syllable as a primary timing unit, and phone represents constituent timing constraints. In syllable-level model, syllable duration is modeled from control factors ranging from phone-level to breath-group levels. In phone-level model, duration modeling is calculated in two steps. First, phone-level LR model using only syllabic constituent phones estimates generalized phone duration for each constituent phone context. Then, constituent phone duration is generated by syllable accommodation using predicted syllable duration from the syllable-level. As introduced, phone and syllable levels contribute timing controls on each other. To cope with the mutual controls, phone context with syllable boundary information are shared between two levels. At syllable-level, it takes intra-syllabic and inter-syllabic phones into account. On the other hand, at phone level, only intra-syllabic constituent phones are considered.

Comparing to conventional syllable-based model [5], there are two main differences. First, controls of syllable-level and phone-level models are not entirely separated. They share syllabic constituent phones to cope with mutual controls. Second, phone duration is not directly determined from syllable accommodation of mean duration and standard deviations. Instead, generalized phone duration for each constituent phone context is used in syllable accommodation. This generalized duration is estimated by the phone-level LR model using current phone and constituent phone context. Hence, the generalized phone duration of all constituent phones have contained intrinsic duration of current phone and duration characteristics of their constituent phone context.

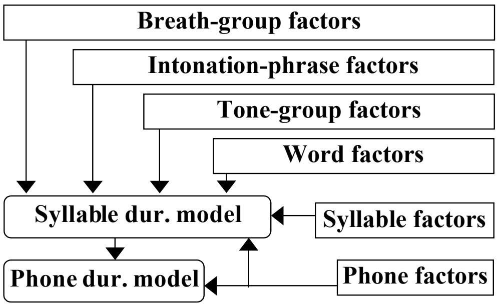


Figure 1 Multi-level LR duration model

2.2. Linear regression model

To model and analyze duration control, a linear regression-based model is adopted for sake of ease on observing underlying control effects. In this work, two linear regression (LR) models are applied to syllable-level and phone-level models. These LR models have been employed in Japanese duration modeling [4], and expressed in Eq. 1.

$$\hat{y}_i = \bar{y} + \sum_f \sum_c x_{fc}\delta_{fc}(i) \quad i = 1,2,3,\ldots,N \qquad (1)$$

where N, $\hat{y}_i$, $\bar{y}$, x_{fc}, and $\delta_{fc}(i)$ represents the number of data, the predicted duration of the i^{th} sample, the mean duration of all samples, the regression coefficient of category c of control factor f, and the characteristic function respectively. The characteristic function represents existing of control factors in the i^{th} sample. The function is set to 1 if the considering factor exists, otherwise, it equals 0. The regression coefficients x_{fc} can be interpreted as control effect of the factor. It can be calculated by minimizing equation (2) using a conventional multiple linear regression method.

$$\sum_i (\hat{y}_i - y_i)^2 \qquad (2)$$

2.3. Control factors

Regarding to control factors, six hierarchical levels of control factors are considered here as shown in Figure 1. Table 1 presents the lists of control factors used in the syllable-level model. The factors range from phone to breath group levels. Concerning to constituent phones and syllabic neighboring context, syllable is designed in the form of onset-nucleus-coda. Onset and coda represent single consonants or consonant clusters. Nucleus covers short and long vowels, and, short and long diphthongs. Table 2 presents that only current phone and syllabic constituent phones are adopted in phone-level model. The current phone covers all phones that defined in onset, nucleus, and coda. In case of the constituent phone factors, they are applied at phone level to contribute syllabic structure information in phone modeling.

Table 1 A lists of control factors for syllable-level model

Control level	Control factors
Breath group	- Length (syllable count in 10 scales) - Position in unit (initial, mid or final)
Intonation phrase	- Length (syllable count in 10 scales) - Position in unit (initial, mid or final)
Tone group	- Length (syllable count in 10 scales) - Position in unit (initial, mid or final)
Word	- Length (syllable count in 10 scales) - Position in unit (initial, mid or final) - Part of speech (47 types)
Syllable	- Current-syllable Tone (Tone 1-5) - Contextual tones (2-preceeding and 2 succeeding tones) - Stress level (stressed/unstressed)
Phone	- Syllabic constituent phones (onset, nucleus and coda) - Syllabic neighboring context (leading-syllable nucleus or coda, and succeeding-syllable onset)

Table 2 A list of control factors for phone-level model

Control level	Control factors
Phone	- Current phone - Syllabic constituent phones (onset, nucleus and coda)

2.4. Training syllable-level and phone-level duration models

To obtain a full model, we start from the syllable-level training. At syllable level, we can directly train the model by employing the control factors listed in Table 1 to linear regression in Eq.1 with the linear optimization as stated in Section 2.2.

In the phone level modeling, the predicted syllable duration from the syllable-level model is used to adjust the predicted phone duration to the predicted syllable one. To train the phone-level LR model, constituent phone duration of the predicted syllable is needed. Since we can obtain only constituent phone duration of the predicted syllable, syllable duration adjustment is needed to the constituent phone durations. For this adjustment, we simply use linear scaling as shown in Eq. 3.

$$y_{ph,acc}(j) = y_{ph}(j) \times \left(\hat{y}_{syl} \middle/ \sum_{i=1}^{N} y_{ph}(i) \right) \qquad (3)$$

where $y_{ph,acc}(j)$, $y_{ph}(i)$, $y_{ph}(j)$, and $\hat{y}_{syl}$ represent syllable-accommodated phone duration of the j^{th} measured phone in mother syllable, measured phone duration of the i^{th} and j^{th} measured phone in mother syllable, and predicted duration of mother syllable.

After the syllable adjustment, the recalculated phone durations were used for training of the phone-level model.

3. Speech data

In this work, 635 phonetically balanced sentences selected from TSynC corpus [15] were adopted. This data contains fluently read speech recorded by a Thai female announcer. Thus, this reading style is more fluent than general reading style and it sounds like announcing speech rather than reading one. The total data length excluding silences is approximately 70 minutes. In syllable-level data set, it contains approximately 20,900 syllables. In phone-level data set, it contains approximately 55,200 phones. Phones are segmented automatically using HMM-based segmentation. The segmentation errors were corrected by hand.

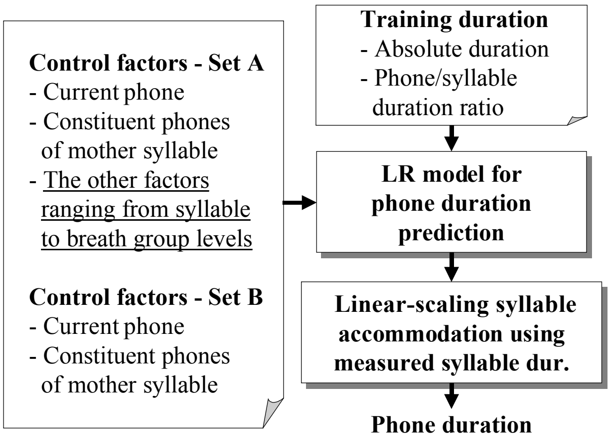
To evaluate the proposed model, both data sets were equally divided into five subsets for evaluation using 5-fold cross-validation. Each subset was used once as a test set and the rest was for training. These data sets were used throughout the paper.

4. Duration prediction experiments

We conducted two experiments to measure the following characteristics: a) effects of higher-phone-level control factors on phone duration b) effects of duration types on syllable accommodation and phone duration. Moreover, we compared the proposed multi-level duration model and a conventional single-level phone-based model.

4.1. Effects of higher-phone-level control factors, duration types and syllable accommodation

To confirm the reasonableness of constituent phone duration calculation discarding higher-level control factors, we analyzed dominant factors in phone duration control. Figure 2 shows the experiment configuration. This experiment sets up a LR phone duration model using two sets of control factors. The first set, referred as Set A, includes all higher-phone-level control factors as input. In contrast, the second one, referred as Set B, discards the higher-level factors. In addition, these two sets share a set of common control factors i.e. current phone and syllabic constituent phones.



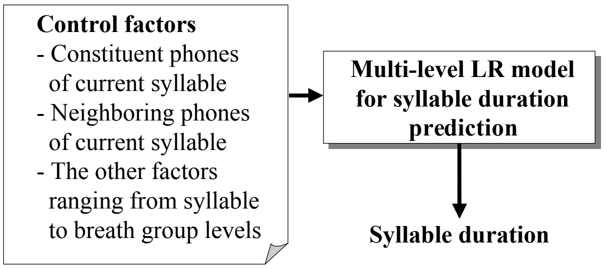
Training duration
- Absolute duration
- Phone/syllable duration ratio
Control factors - Set A
- Current phone
- Constituent phones of mother syllable
- The other factors ranging from syllable to breath group levels
Control factors - Set B
- Current phone
- Constituent phones of mother syllable
LR model for phone duration prediction
Linear-scaling syllable accommodation using measured syllable dur.
Phone duration

Figure 2 Experiment on the effects of higher-phone-level control factors in phone duration

Since constituent phones have to properly fit into syllable frame, syllable accommodation adjustment was evaluated. To fit the phones into the mother syllable, two types of fitting are evaluated. The first method adopts absolute phone duration with linear scaling. The other one is based on duration ratio between constituent phone duration and the mother syllable duration. In this method, all phone duration in the same mother syllable is normalized by duration of the mother syllable. The syllable ratio was used for every constituent phones. Thus, summation of all constituent duration ratios in a syllable always equals one unit. During syllable accommodation, syllable duration measured from the corpus is used as duration template for fitting. To compare the results, both duration types are applied to both duration models.

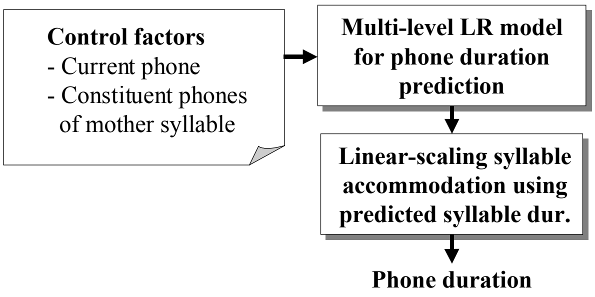
4.2. Multi-level and single-level modeling comparison

To evaluate the proposed multi-level duration model, we compared the proposed one and a conventional single-level one. Figure 3 shows the configuration of syllable-level model. To predict duration, we employed the following control factors as input; constituent phones, syllabic neighboring context, current-syllable tone, contextual tones, stress level and all control factors from other higher levels ranging from word to breath group levels. Syllable duration predicted from the model was used for the phone-level model. Figure 4 shows the phone-level model as explained in Section 4.1 with the factors Set B. As stated, absolute phone duration is used in this model.



Control factors
- Constituent phones of current syllable
- Neighboring phones of current syllable
- The other factors ranging from syllable to breath group levels
Multi-level LR model for syllable duration prediction
Syllable duration

Figure 3 Syllable-level control factors in the multi-level duration model



Control factors
- Current phone
- Constituent phones of mother syllable
Multi-level LR model for phone duration prediction
Linear-scaling syllable accommodation using predicted syllable dur.
Phone duration

Figure 4 Phone-level control factors in the multi-level duration model

For comparison, we predicted the phone duration using the conventional single-level phone-based model employing LR model expressed in Eq. 1. Figure 5 shows the control factors for a conventional single-level phone duration model. In this phone-based model, we discarded syllable boundary information and treated every phone boundary in the same manner. In this model, we adopted size-equivalent moving windows of five phones centered at the mid position as input to cover longer context. To make the model comparable, the other control factors that apply to the multi-level model were also included in this model.

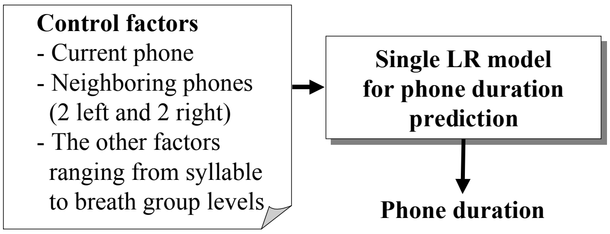

Control factors
- Current phone
- Neighboring phones (2 left and 2 right)
- The other factors ranging from syllable to breath group levels
Single LR model for phone duration prediction
Phone duration

Figure 5 Control factors employed for a conventional single-level phone duration model

5. Experimental results

We evaluated prediction precision of the proposed duration model using 5-fold cross validation to avoid any bias problem in selecting the training and test sets. RMS errors and Pearson's product-moment correlation between measured and predicted durations are employed as evaluation measures. The average values of the results from the cross validation are used for analysis.

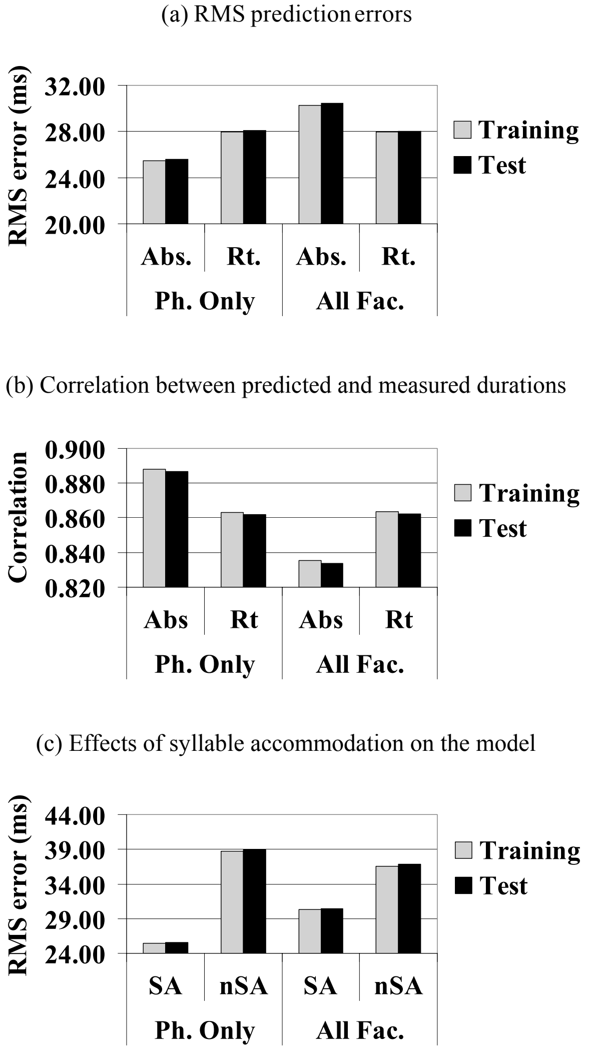
5.1. Analysis results supporting multi-level modeling

RMS prediction errors of the phone duration model are shown in Figure 6 (a). As shown in the Figure, the model using only phone factors gave equivalent or better prediction than the model using all control factors. Phone duration calculation using absolute duration gave the better results than the conventional one using ratio. The correlations between the predicted and the measured phone duration showed the same tendency as shown in Figure 6 (b). These results suggest that phone duration is highly correlated with local phone factors, and that absolute duration gives better results than ratio used in conventional phone duration calculation.

As shown in the results, it is to be noted that the model using only phone factors performs better than the one that includes all factors. This fact supports the reasonableness of the proposed multi-level formulation since syllable performed as a timing frame for constituent phones.

Moreover, we also compared the predicted phone duration before and after syllable accommodation using the same experiments described in Section 4.1. Figure 6 (c) shows that the model with syllable accommodation gave better prediction results than the one without syllable accommodation. As shown in the Figure, we could find that, before adjusting syllable accommodation, the model including higher-phone-level factors gave better prediction than the one using only phone factors. After accommodating syllable with constituent phones, the model using phone factors only gave better results, instead.

Thus, these results support the idea that the high-phone-level control factors mainly contribute on syllable duration, and, that phone duration is mainly controlled by local constituent phone factors. Furthermore, this show possibility to separate a phone duration model from other level models.


(a) RMS prediction errors

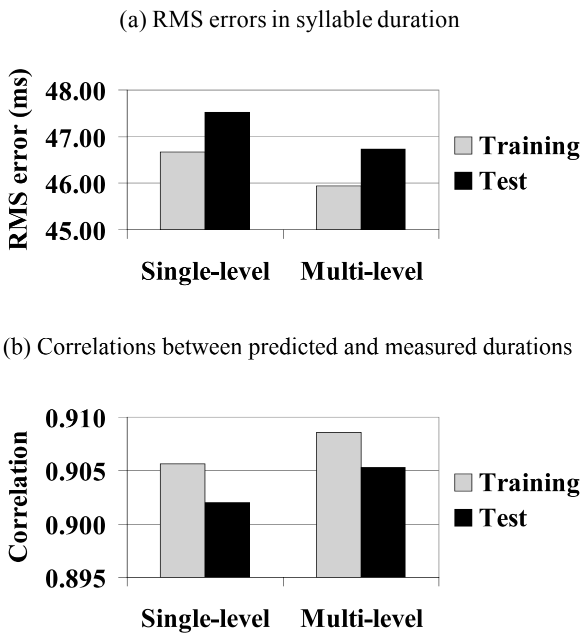
(b) Correlation between predicted and measured durations

(c) Effects of syllable accommodation on the model

Figure 6 Prediction results of the phone duration model (Ph. Only and All Fac. stand for using phone factors only, using all control factors, respectively. Abs. and Rt. mean phone duration calculation using absolute duration, duration ratio respectively. SA and SA stand for with, without syllable accommodation, respectively.)

5.2. Comparison between the proposed model and a conventional one

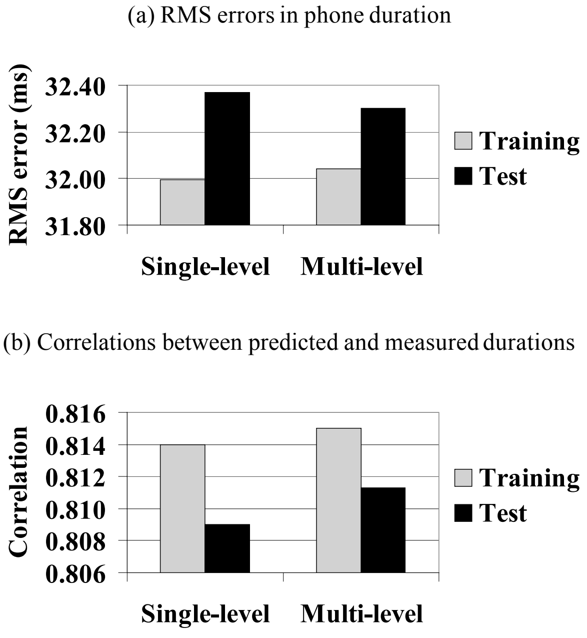
The RMS errors between predicted duration and the observed one were calculated both for syllable and phone durations. Figure 7 (a) shows syllable-level prediction errors between the multi-level and single-level models. As shown in the Figure, the multi-level model gave better prediction precision than the single-level duration model. Figure 7 (b) shows that the proposed model gave a higher prediction correlation than the single-level one.


(a) RMS errors in syllable duration

(b) Correlations between predicted and measured durations

Figure 7 The comparison of prediction precision in syllable duration between the multi-level model and the single-level model

At phone level, both prediction errors and correlation of the proposed model were better than those of the single-level model as shown in Figure 8. The different of prediction accuracy of both models were not so big. However, the better results at both syllable and phone levels support that the proposed model outperforms the single-level model in duration modeling.


(a) RMS errors in phone duration

(b) Correlations between predicted and measured durations

Figure 8 The comparison of prediction precision in phone duration between the multi-level model and the single-level model

To analyze correspondence between measured and predicted syllable duration, we showed the scatter plot of measured syllable duration versus predicted one in Figure 9. As shown in this Figure, it clearly showed two duration groups centered approximately at 200 and 500 ms, respectively. We found that these groups corresponded to syllables at non-final-phrase and final-phrase positions, respectively. A large lengthening effect of final-phrase position about twice of non-final-phrase position is presented. We could see well correspondence between the measured and the predicted duration throughout the duration range of the non-final-phrase duration group. In contrast, the final-phrase duration group showed small duration deviation from the equivalent line between the measured and predicted syllable duration. These results suggest that the syllable model underestimated syllable duration at final phrase.

Figure 10 shows a scatter plot of measured phone duration versus predicted one. As shown in the Figure, strong correspondence between measured and predicted phone durations that approximately less than 200 ms are found. The phones in the range mainly correspond to the phones in non-final-phrase syllables. In contrast, the phone longer approximately than 200 ms showed underestimated prediction on phone duration. We found that the phones mainly correspond to the nucleuses (vowels) and some voiced codas; nasals and glides, in final-phrase syllables, which tend to be arbitrarily lengthened unlike those at other positions. Thus, these findings suggest that the underestimation of final-position vowel duration is one cause of modeling problems. It also suggests that the duration control characteristics at final phrase are different from the other positions, and, they should be modeled with additional controls.

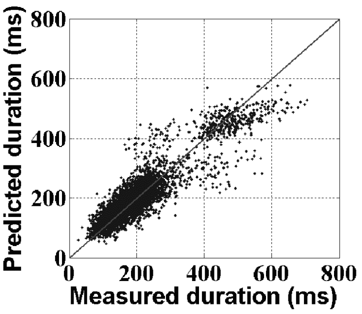


Figure 9 Comparison between measured and predicted durations at syllable level

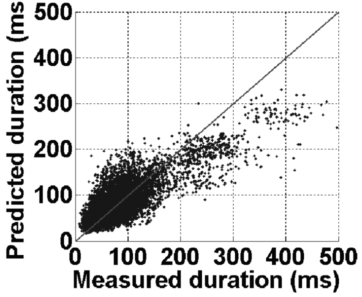


Figure 10 Comparison between measured and predicted durations at phone level

6. Conclusions

In this paper, we proposed a statistical-based Thai duration modeling using multi-level linear regression model and syllable accommodation. The duration prediction experiments showed that the multi-level LR model using absolute duration gave more accurate prediction than the single LR model. This model can serve with less complex and more structural model which better reflect mutual control duration characteristics of phone and syllable duration control. From the viewpoint of duration control, higher-phone-level factors mainly contribute on syllable. The results also suggested that syllable is highly correlated to higher-level timing controls, while, phone differences by themselves do not affect higher control and contribute to local timing control only.

In future works, to better understand the mechanism, duration modeling of higher timing control unit such as stress group or tone group will be investigated. In addition, further study on tempo and rhythm modeling will be carried out by integrating the knowledge in computational duration modeling. So far, the proposed model performs very high correlation on prediction but it still gives big prediction errors. We speculate that the limitation of LR method itself on coping interaction among control factors is one of the reasons. Thus, more optimized methods, e.g. Constrained Tree Regression (CTR) [11], will be employed. Furthermore, selection and optimization on control factors are also to be refined.

7. Acknowledgements

Work supported in part by Waseda University RISE research project of "Analysis and modeling of human mechanism in speech and language processing" and Grant-in-Aid for Scientific Research A, No. 16200016 of JSPS.

8. References

[1] Klatt, D. H., "Review of text-to-speech conversion for English", *Journal of the Acoustical Society of America*, Vol. 82(3), pp. 737-793, 1987.

[2] van Santen, J. P. H., “Analyzing N-way tables with sums-of-products models”, *J. Mathematical Psychology*, 37:327-371, 1993.

[3] Riley, M.D., "Tree-based modeling of segmental durations", *Talking Machines*, North-Holland, 1992.

[4] Kaiki, N., and Sagisaka, Y., “The control of segmental duration in speech synthesis using statistical methods”, *Talking Machine*, pp. 255-263, 1992.

[5] Campbell, W.N., *Multi-level speech timing control*, Ph.D. dissertation, Univ. of Sussex, 1992.

[6] Sagisaka, Y., “Modeling and perception of temporal characteristics in speech”, *15th ICPhS 2003*, pp.1-6, 2003.

[7] van Santen, J., and Shih, C., "Suprasegmental and Segmental Timing Models in Mandarin Chinese and American English", *J. Acous. Soc. Am.*, Vol. 107 (2), pp. 1012-1026, 2000.

[8] van Son, R., and, van Santen, J., "Modeling the interaction between factors affecting consonant duration", *Proc. EUROSPEECH*, pp. 319-322, 1997.

[9] Tseng, C., Pin, S., Lee, Y., Wang, H., and, Chen, Y., “ Fluent speech prosody: Framework and modeling”, *Speech Communication,* 46(3-4), pp. 284-309, 2005.

[10] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T., “Simultaneous Modeling of Spectrum, Pitch and Duration in HMMBased peech Synthesis,” *Proc. of EUROSPEECH, vol.5*, pp.2347–2350, 1999.

[11] Iwahashi, N. and Sagisaka, Y., "Statistical Modeling of Speech Segment Duration by Constrained Tree Regression", *Trans. IEICE*, E83-D, pp. 1550-1559, 2000.

[12] Hansakunbuntheung, C., and Sagisaka, Y., “Studies on the effects of intonation structure on Thai segmental duration”, *ALOHA workshop*, p. 34, 2006.

[13] Hansakunbuntheung, C., and Sagisaka, Y., “The comparison between Thai and Japanese temporal control characteristics using segmental duration models”, *J. Acous. Soc. Amer., 4th ASA-ASJ meeting*, 3295, 2006.

[14] Hansakunbuntheung, C., and Sagisaka, Y., “Multi-level linguistic and prosodic control factor analysis on Thai segmental duration modeling”, *Proc. ASJ spring meeting*, pp.281-282, 2007.

[15] Hansakunbuntheung,C., Tesprasit, V., Sornlertlamvanich , V., “Thai Tagged Speech Corpus for Speech Synthesis”, *The Oriental COCOSDA 2003*, 97-104, 2003.

Linguistic and Mixed Excitation Improvements on a HMM-based speech synthesis for Castilian Spanish

Xavier Gonzalvo, Joan Claudi Socoró, Ignasi Iriondo, Carlos Monzo, Elisa Martínez

GPMM - Grup de Recerca en Processament Multimodal
Enginyeria i Arquitectura La Salle. Universitat Ramon Llull
Quatre Camins 2, 08022 Barcelona (Spain).
{gonzalvo,jclaudi,iriondo,cmonzo,elisa}@salle.url.edu

Abstract

Hidden Markov Models based text-to-speech (HMM-TTS) synthesis is one of the techniques for generating speech from trained statistical models where spectrum and prosody of basic speech units are modelled altogether. This paper presents the advances in our Spanish HMM-TTS and a perceptual test is conducted to compare it with an extended PSOLA-based concatenative (E-PSOLA) system. The improvements have been performed on phonetic information and contextual factors according to the Castilian Spanish language and speech generation using a mixed excitation (ME) technique. The results show the preference of the new HMM-TTS system in front of the previous system and a better MOS in comparison with a real E-PSOLA in terms of acceptability, intelligibility and stability.

1. Introduction

One of the main problems of concatenative text-to-speech (TTS) systems is the degradation of quality when the database does not comprise the best units to be synthesized. Hence, larger databases are required for these kinds of systems. As the database grows up, it is more suitable to contain a unit closer to the target and more likely to have a better join [1]. In order to reduce errors, this database could become difficult to process. Therefore, a common solution is to use a limited domain context where text to be synthesized is under control (e.g. Virtual Weather man [2]).

Thence it follows that the final objective is to improve quality and naturalness in applications for general purpose. The main feature of the HMM-TTS is the statistical modelling of units producing a smoothed and natural speech that have been shown to be a possible advantage in front of the quality discontinuities in the concatenative systems [3]. Moreover, the main benefit of HMM-TTS is the capability of modelling voices in order to synthesize different speaker features, styles and emotions and perform interesting adaptations of speech [4]. Furthermore, HMM for speech synthesis could be used in new systems able to unify both approaches and to take advantage of their properties [5]. At this point, interesting work was presented by [6] to develop a fused system and last contributions have been presented in [7].

The aim of this paper is to present the advances throughout the development of a high-quality HMM-TTS for Castilian Spanish based on HTS engine [8]. Previous work for Spanish [9] identified the common problems that affect the HMM-TTS systems and other languages as well: vocoder, modelling accuracy and over-smoothing [7]. The following improvements are related to linguistic and vocoder issues which try to solve or alleviate these problems.

Firstly, the following linguistic features have been updated. In the one hand, the unit clustering has been upgraded using new contextual factors with respect to the previous approach [9], where the HMM training was presented to use a decision tree-based context clustering in order to improve models training. Also, clustering is able to characterize phoneme units introducing a counterpart approach with respect to English [3]. On the other hand, grapheme-to-phoneme conversion now uses a rule-based system to fix pronunciation errors instead of the Festival Spanish voice [10]. Secondly, synthesis quality has been increased by applying a mixed excitation (ME) technique using well defined models of the parametrized residual excitation [17]. The system is based on a source-filter model approach to generate speech directly from HMM itself. One of the drawbacks of these systems is the non ideal speech reconstruction due to the parametric representation of speech that the ME technique can solve by adding extra excitation parameters to the model.

This paper is organized as follows: Section 2 describes HMM system workflow and parameter training for spectrum, pitch, ME and duration. Section 3 concerns to synthesis process description. Section 4 presents measures, section 5 discusses results and final section presents the concluding remarks and future work.

2. HMM-based TTS system training

As in any HMM-TTS system, two stages are distinguished: training and synthesis. Figure 1 depicts the classical system training workflow (dotted lines stand for parameters modelled within the HMM).

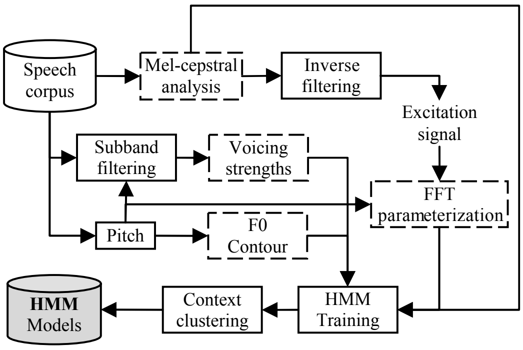


Figure 1: *Training workflow*

First, mel-cepstral analysis of the speech is performed. The first step estimates the HMM for isolated phonemes (each HMM represents a contextual phoneme) and each of these models will be used as an initialization of the contextual phonemes. Then, similar phonemes are clustered by means of a decision tree using contextual information and previously designed questions. Unseen units during the training stage can be synthesized using these decision trees. Each contextual phoneme HMM definition includes spectrum, state durations, F0, ME FFT parameters and the voicing strengths (VS) coefficients. During the analysis of these information, pitch is used for subband filtering and FFT parametrization.

Topology used is a 5 states left-to-right with no-skips. Each state is represented with 4 independent streams, one for spectrum, one for pitch and two more for mixed excitation part which comprises both FFT and VS. Each parameter is completed with its delta and delta-delta coefficients. The modelling information is structured in table 1.

Table 1: *Information modelled in the HMM.*

Feature vector streams			
c	Δc	Δ^2c	Spectrum
p	Δp	Δ^2p	F0
me	Δme	Δ^2me	FFT parameters for ME
v	Δv	Δ^2v	Voicing strengths

2.1. Spectrum modelling

The system is based on a source-filter model and spectrum parameters are modelled as multivariate Gaussian distributions [11]. Depending on the type and number of coefficients used on the vocoder, the quality of the synthetic speech can significantly vary. In this work, spectrum is updated to be modelled from 12^{th} to 24^{th} order mel-cepstral coefficients which generate speech with the MLSA (Mel Log Spectrum Approximation) filter [12]. The advantage of mel-cepstral in front of standard MFCC is that spectrum is better represented, so it gives a better performance of speech during synthesis [12]. Mel-cepstral has presented good results improving the basic HMM system in languages such as Arabic [13].

Last advances in high quality HMM-TTS used the STRAIGHT-based vocoding [14]. This analysis/synthesis technique is considered a high-quality solution initially used for speech morphing though it has been successfully applied to HMM-TTS (e.g. Blizzard 2005 [15]). Although it presents the advantage of performing pitch-adaptative spectral analysis, it was shown in [15] that MLSA filter was the most computationally efficient synthesis approach.

2.2. Mixed excitation

The aim of using a mixed excitation is to mimic the characteristics of natural human speech. It was first used in the LP vocoder (MELP) [16], a low bit rate speech coding and later integrated in a HMM-TTS for Japanese [17]. The reason for the vocoded speech quality is attributed mainly to the insufficiency of the binary source signal model which switches exclusively either the impulse train or the white noise. To solve this, the mixed excitation is implemented using a multi-band mixing structure.

As in the case of spectrum, STRAIGHT has also been used for the design of the mixed excitation as it weights a sum of a pulse train with phase manipulation and Gaussian noise. Other interesting schemes proposed the design of ME using wavelet [18].

The main information used to train the HMM is the following:

- Bandpass voicing strengths. The speech signal is filtered into five frequency bands considering a sample rate of 16k Hz [17] (see figure 1). The voicing strength in each band is estimated using normalized correlation coefficients around the pitch lag. In spite of correcting pitch estimation simultaneously with correlation, first the pitch is marked up and later, the correlation in each band is computed.
- Fourier magnitudes. In this work, the FFT parameters are the first thirty magnitudes of the centred pitch period of a 20ms excitation frame. The residual excitation is obtained by inverting the exponential filter transfer function [12] and filtering.

2.3. Pitch, mixed excitation and duration modelling

Pitch marks are crucial in order to obtain a good synthesis as they affect the representation of various parameters and the posterior training of the models. On the one hand, F0 contour is simultaneously modelled within the HMMs, hence estimated contour is dependent on the correctness of the pitch marks. On the other hand, mixed excitation FFT coefficients are estimated based on the determined pitch sequence. Thus, the Spanish corpus pitch analysis has been performed using an approach that automatically reduces the mark-up errors by using dynamic programming [19]. Moreover, this algorithm reduces discontinuities in the generated F0 curve for synthesis.

F0 model (table 1) is a multi-space probability distribution [11] that must be used in order to store continuous logarithmic values of the F0 curve and a discrete indicator for voiced/unvoiced. As in the case of spectrum, FFT magnitudes and voicing strengths are modelled as multivariate Gaussian distributions.

State durations of each HMM are modelled by a multivariate Gaussian distribution [20]. Its dimensionality is equal to the number of states in the corresponding HMM.

2.4. Phonetic data

The Spanish female voice was created from a corpus developed in conjunction with LAICOM [21]. Speech was recorded by a professional speaker in neutral emotion. Time boundaries segmentation was performed using an embedded HMM training, segmented and finally revised by speech processing researchers.

Phonetic labelling was performed in the previous work [9] using the Festival [10] Spanish voice. In order to resolve some incorrect transcriptions, a tested rule based approach (SinLib [22]) has been applied for text analysis in this work.

The grapheme-to-phoneme conversion has been extended from 31 to 36 units (see table 2) with one model of silence (types of silences are POS-tagged). It is important to notice that the system has the feature of a continuous transcription, so rules are applied between words (e.g. /**b**arko/ and /mi**B**arko/, translated as, “ship” and “my ship”).

- **Vowels**. Models for vowels are different either if they are stressed (capital letters) as also used in other approaches [23]. The system distinguishes various types of vowels:

semi-vowel, half open, open, closed and half closed including the main group of table 2.

- **Consonants**. New consonants (emphasized in bold) are used to avoid some pronunciation errors and improve intelligibility. Apart from the main groups, the system is also able to consider dental, velar, bilabial, alveolar, palatal, labio-dental, inter-dental, pre-palatal and voiced/unvoiced.

Table 2: *Castilian Spanish consonants and vowels inventory (SAMPA [24]).*

Vowels	
Frontal vowels	j,i,I,e,E,a,A
Back vowels	o,O,u,U,w
Consonants	
Plosive	p,b,t,d,k,g
Nasal	m,n,J,**N**,**M**
Fricative	**B**,f,tS,T,**D**,s,x,**G**
Lateral	l,L
Rhotic	R,r

2.5. Contextual factors

Input text is converted into a complete list of contextualized phonemes and each one is represented by a HMM. As the contextual information increases, HMMs will have less training data. To solve this problem during the training stage, similar units are clustered using a decision tree [11].

Extracted contextual information is language dependent and it serves as the features (attribute-value pairs) to construct the clustering decision trees. These trees are constructed using a set of questions designed in base of the contextual factors and the unit features using a yes/no based decision. Information referring to spectrum, F0, duration and ME is independently clustered by different trees.

Basically, the new approach in this work is focused on intonational improvement. English HMM-TTS included the ToBi tags which have been widely studied and applied to many systems [25]. In our case, we apply two groups of phonemes (Accentual group (AG) and Intonational group (IG)) in order to better represent the expressiveness. These parameters presented good results in a F0 estimator based on a machine learning approach applied to Spanish [9]. New information related to prosody events is the following:

- AG. Incorporates syllable influence and is related to speech rhythm. The type of AG is specified in Spanish as *agudo*, *plano*, *esdrújula* and *sobre-esdrújula* depending on the position of the accented syllable in the word.
- IG. Structure at this level is reached concatenating AGs. There are three types: interrogative, declarative and exclamative.
- AGs and IGs start/end flags.
- Syllable and word start/end flags.

New features are related to flags for syllable, words and intonational groups boundaries (SinLib system also controls these boundaries) and Part-of-speech (POS) that has been upgraded using Freeling [26] (a morphological engine). The following parameters are used to design the questions for the tree-based clustering and are presented in hierarchical order:

1. **Phonemes**. Current phone, left and before left phones and identical for the right side. Each kind of phoneme is labelled independently depending on the characteristics of table 2.
2. **AG**. The number of phonemes in current, previous and next AG; start/end flag and type of AG.
3. **IG**. Start/end flag and types of IG.
4. **Syllable**. Stress of current, previous and next syllables; position forward and backward of current syllable in current word and in current phrase; number of stressed syllables with respect to contextual syllables (this comprises 4 factors); vowel of the syllable and start/end flag.
5. **Word**. POS of the current, next and previous words; the number of syllables of current, next and previous words and position (forward and backward) of word in phrase and start/End flag.
6. **Phrase**. Number of syllables and number of words in current, previous and next phrases; positions (forward and backward) of current phrase in the utterance.
7. **Utterance**. Number of syllables, words and phrases in the utterance.

3. HMM-based TTS system synthesis

Figure 2 shows the synthesis workflow. Once the system has been trained, it has a set of phonemes represented by contextual factors. The first step is devoted to produce a complete contextualized list of phonemes from a text to be synthesized. Chosen units are converted into a sequence of HMM.

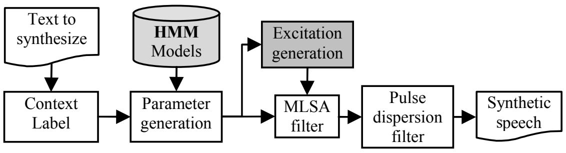


Figure 2: *Synthesis workflow*

Necessary parameters to synthesize are generated from the HMM using the algorithm proposed in [27]. The HMM is composed of the data and its Δ and Δ^2 features (see table 1). By taking into account the constraints between static an dynamic features, the algorithm avoid generating identical parameters for each state of the same HMM which results on an improved and smoothed speech envelope. Generated data are mel-cepstral, F0 and ME parameters. Duration is also estimated to maximize the probability of state durations.

Excitation signal is generated from the F0 curve, voiced and unvoiced information and the FFT parameters. Figure 3 presents the scheme to generate the mixed excitation (dotted lines indicates parameters generated from HMM). The pulse excitation is calculated from Fourier magnitudes using an inverse DFT of one pitch period in length. The bandpass filter for voiced and unvoiced parts are given by the sum of all the bandpass filter coefficients for the voiced and unvoiced frequency bands respectively. Voicing strengths are used to decide whether each filter coefficients belong to the voiced or unvoiced part. The excitation is generated as the sum of the filtered periodic and noise excitations.

In order to reconstruct speech, the system uses spectrum parameters as the MLSA filter coefficients and excitation as the signal to filter. Finally, the obtained speech is filtered by a pulse dispersion filter which is a 130^{th} order FIR derived from a spectrally flattened triangle pulse based on a typical female pitch period. The pulse dispersion filter can reduce some of the harsh quality of the synthesized speech [16].

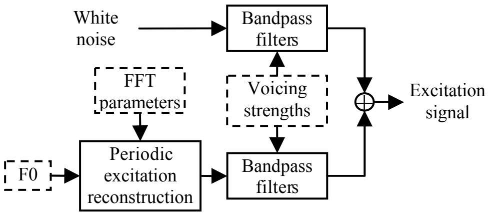


Figure 3: *Mixed excitation generation during the synthesis stage.*

One of the main problems during parameter generation is over-smoothing [28] that decreases the expressiveness and naturalness. Although the first solution would be to increase the size of the trees, its effect does not represent a substantial improvement in quality [9]. Another solution to improve the expressiveness could be to use an external F0 estimator though it can reproduce a forced intonation in some cases [9].

Last advances presented in [7] focus their study on reducing the error of the generated parameters. The HMM likelihood for a parameter trajectory generated by the conventional algorithm is too large compared with that for a natural one. This implies that is not only necessary to maximize the HMM likelihood [28]. For this case, minimum generation error (MGE) [29] or global variance (GV) [28] presented good results. GV introduces new constraints to the method of training and generation in order to avoid over-smoothing. The results reported were very good though at the moment is only showed to perceptually improve speech quality when applied to both mel-cepstral and F0.

4. Experiments

Experiments are conducted on a female corpus and evaluated using perceptual tests. The system was trained with HTS [8] using 620 phrases of a total of 833 (25% of the corpus is used for testing purposes). Contextual factors represent around 20000 units to be trained and around 5000 are unseen units.

Firstly, texts were labelled using contextual factors described in section 2.5. Then, HMMs are trained, decision trees for spectrum, F0, state durations and ME are built. Finally, HMM models are clustered. These trees are different among them because spectrum, F0 and states duration are affected by different contextual factors. Table 3 presents only two features to show the type of information in each tree. While spectrum tree is focused on phoneme features, excitation tree presents more high level information related to phrases (e.g. AG has increased the representation with respect to the spectrum tree).

Table 3: *Main contextual factors used for each tree.*

Feature vector	Contextual factors
Spectrum	Ph. 87%, AG 2%, Syll. 4%
Excitation	Ph. 45%, AG 16%, Syll. 10%
Durations	Ph. 76%, AG 8%, Syll. 5%
FFT	Ph. 21%, AG 11%, Syll. 28%
RV	Ph. 8%, AG 7%, Syll. 34%

It has been observed and discussed that RMSE is not a valid objective measure for F0 as it does not reflect real improvements showed by perceptual tests. For example, the generation algorithm considering GV usually causes larger errors compared with the conventional one [28] though GV increases the naturalness of synthesized speech. Meanwhile, subjective speech quality evaluation is generally seen to be the best measure of the aesthetic aspects [30] which is used to validate most of the TTS systems. Taking this into account, what follows presents a set of perceptual tests [1] to measure the improvements of the current HMM-TTS system.

In the first test, the systems with standard excitation (OLD-HMM) and the new system (ME-HMM) are evaluated. Figure 4 presents the preference of the new system in front of the old one. The effect of the ME (i.e. speech reconstruction buzzy is significantly reduced) is more important than the linguistic improvements. The preference tests evaluated single sentences by 15 listeners.

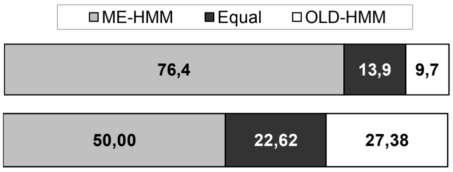


Figure 4: *Preference test for OLD-HMM and ME-HMM systems: (up) ME and linguistic improvement, (down) only linguistic improvements*

Once the new system has been validated, the second test (see figure 5) goal is to compare HMM-TTS systems with E-PSOLA [31] in terms of acceptability, intelligibility and naturalness. The perceptual comparisons were conducted using the same number of training sentences for both HMM-TTS and the E-PSOLA systems. Notice that the HMM-TTS systems model the F0 contour of a female voice with high variability (μ_{F0}=167 Hz, σ_{F0}=41 Hz) and the E-PSOLA version has real prosody from corpus as input.

The test was performed using a five steps (1-5) Mean Opinion Score (MOS) corresponding to the following quality evaluation: bad, poor, fair, good and excellent. The number of listeners were 25, most of them students of a technical degree and twenty phrases were randomly chosen for each system.

Different studies refer to acceptability as a measure of different components [30]. It is clear that in subjective user evaluations, at least intelligibility and naturalness play an important role. Subjective acceptability is not necessarily a simple consequence of intelligibility, and a distinction needs to be made between the aesthetic and functional aspects of synthetic speech.

1. **Acceptability.** Figure 5 shows that acceptability is higher for ME-HMM than for the other two systems, reaching a MOS of 2.8.
2. **Naturalness.** This measurement deals with quality and intonation as a measure of the extent to which a synthesizer sounds like a human [30]. In the one hand, the main

[1] See http://www.salle.url.edu/~gonzalvo/hmm, for some synthesis examples

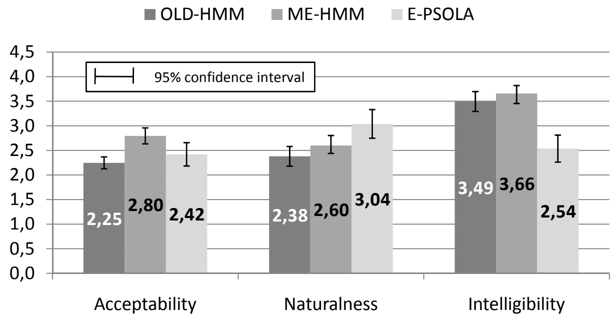


Figure 5: *Acceptability, intelligibility and naturalness MOS tests for ME-HMM, OLD-HMM and E-PSOLA systems.*

problem of the HMM-TTS is that produces a flat synthesis in some phrases. Moreover, although using a ME approach, the best example of a concatenative system still produces a better synthesis than the best HMM-TTS reconstruction [7]. On the other hand, E-PSOLA synthesis sounds more like a human but naturalness is affected by quality discontinuities. In any case, ME-HMM improves quality in comparison to the OLD-HMM due to the use of ME and new contextual factors (see section 2.5).

3. **Intelligibility.** This measurement marks the quality to distinguish the maximum number of words in a phrase. While E-PSOLA produces strong discontinuities that affect the comprehension of the phrases, HMM-TTS systems solve it by means of a smoother synthesis. This test also measures the effect of the linguistic changes (see section 2.4) with respect to the OLD-HMM.

Finally, as concluded for other languages (e.g. English [3] or European Portuguese [32]) HMM-TTS presents the most stable quality and although is less natural than E-PSOLA, it avoids quality discontinuities. In order to measure this, figure 6 shows the stability of the acceptability test in a bar graph. Notice that the E-PSOLA system is able to present more high-quality sentences but the probability of producing a bad synthesis is also higher than for the ME-HMM system. Stability of the ME-HMM system is then guaranteed thanks to a high probability "fair" zone.

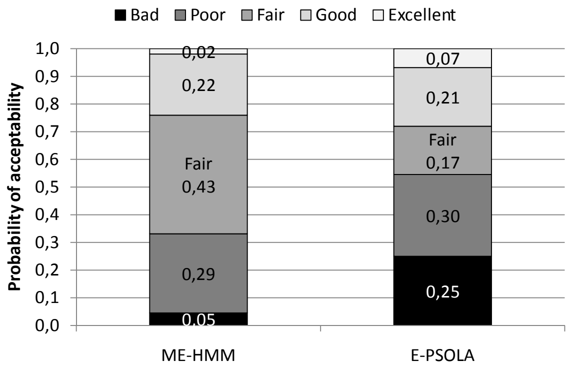


Figure 6: *Stability comparison based on the acceptability MOS results.*

5. Discussion

In order to analyse the concrete effect of the HMM-TTS, this work has presented a perceptual test in order to separate the factors that make a HMM-TTS preferable for general purposes applications controlling the length of the corpus.

In the one hand, the advantage of the HMM-TTS systems is its ability to maintain the synthesis quality for any text to be synthesized and the main drawback is the naturalness of the final produced speech. Using a HMM-TTS provides a high intelligibility system, that could even be more independent of corpus label errors than a standard concatenative system. In fact, the perceptual results could justify one of the possible aspects to make the acceptability be higher for system based on HMM-TTS, that is, the intelligibility and a quality able to reduce the vocoded speech.

Therefore, HMM-TTS systems used in a non limited domain applications provide stability. The intelligibility test could be the main reason because results have shown that smooth speech with a high intelligibility is preferable though a concatenative system still provided a higher naturalness.

6. Conclusions and future work

This work has presented the improvements on a Spanish HMM-TTS based on HTS updating new phonetic information, appending the AG and IG to contextual factor and integrating a ME scheme. With a set of tests we have compared the performance against a concatenative synthesis system. Subjective measures presented the advance of the system in terms of acceptability, intelligibility, naturalness and stability. The results have shown that the HMM-TTS for Spanish presents a better intelligibility and the ME reduced the buzzy vocoder quality. Also acceptability and stability of the system has presented an advantage in front of other kinds of synthesis in general purposes application.

HMM-TTS produces a flat synthesis caused by a smooth F0 contour and mel-ceptral parameters estimation. The conclusion from the results is that the HMM-TTS system is more suitable due to produce a continuous and more stable synthesis. However, although naturalness has been improved with regards to the previous system, it is still a lack and more expressiveness is still desirable. In this aspect, it seems to be necessary to integrate a parameter generation using minimum error to gain expressiveness and naturalness. New techniques and vocoders (e.g. Harmonic-Noise Model or STRAIGHT) have presented successful results in TTS systems, so a logical step would be to compare its performance with our current system. Moreover, it would be interesting to shape the HMM generated F0 contour with an external F0 estimation using an extended version of the system presented in the last approach [9].

Voice transformation and conversion techniques will be applied in the future. Finally, perceptual tests have been used to measure the subjective quality of the system. Due to RMSE is not a correct measure to objectively measure the improvements of the systems, it would be desirable to propose a new objective measure to evaluate the HMM-TTS systems quality that could also be extended to other types of synthesis. Voice quality descriptors could deal with this topic in the future.

7. Acknowledgements

This work has been developed under SALERO (IST FP6-2004-027122). This document does not represent the opinion of the European Community, and the European Community is not re-

sponsible for any use that might be made of its content.

8. References

[1] Black, A., “Perfect synthesis for all of the people all of the time”, Proc. of IEEE SSW, 2002.

[2] Alías, F., Iriondo, I., Formiga, Ll., Gonzalvo, X., Monzo, C. and Sevillano, X., “High quality Spanish restricted-domain TTS oriented to a weather forecast application”, Proc. of Interspeech, 2005.

[3] Tokuda, K., Zen, H. and Black, A.W., “An HMM-based speech synthesis system applied to English”, Proc. of IEEE SSW, 2002.

[4] Tamura, M., Masuko, T., Tokuda, K., Kobayashi, T., “Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR”, Proc. of ICASSP, pp.805–808, May 2001.

[5] Donovan, Robert E. and Woodland, P. C., “A hidden Markov-model-based trainable speech synthesizer”, Computer Speech and Language, vol.13, pp.223–241, 1999.

[6] Taylor, P., “Unifying Unit Selection and Hidden Markov Model Speech Synthesis”, Proc. of Interspeech, 2006.

[7] Black, A., Zen, H. and Tokuda, K., “Statistical Parametric Speech Synthesis”, Proc. of ICASSP, pp.1229–1232, 2007.

[8] Tokuda, K., Zen, H., Yamagishi, J., Masuko, T., Sako, S., Black, A.W and and Nose, T., “The HMM-based speech synthesis sysmte (HTS)”, http://hts.ics.nitech.ac.jp

[9] Gonzalvo, X., Iriondo, I., Socoró, J.C., Alías, F. and Monzo, C., “HMM-based Spanish speech synthesis using CBR as F0 estimator”, Proc. of NoLISP, 2007.

[10] Black, A. W., Taylor, P. and Caley, R., “The Festival Speech Synthesis System”, http://www.festvox.org/festival

[11] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi T. and Kitamura, T., “Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis”, Proc. of Eurospeech, pp. 2374–2350, 1999.

[12] Fukada, T., Tokuda, K., Kobayashi, T. and Imai, S., “An adaptive algorithm for mel-cepstral analysis of speech”, Proc. of ICASSP, pp.137140, 1992.

[13] Ossama, A-H., Sherif Mahdy, A. and Mohsen, R., “Improving Arabic HMM based speech synthesis quality”, Proc. of Interspeech, pp.1332-1335, 2006.

[14] Kawahara, H., Estill, Jo. and Fujimura, O., “Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system STRAIGHT”, MAVEBA, 2001.

[15] Zen, H. and Tomoki, T., “An overview of nitech HMM-based speech synthesis system for blizzard challenge 2005”, Proc. of Interspeech, pp.93–96, 2005.

[16] McCree, A. V. and Barnwell III, T. P. , “A mixed excitation LPC vocoder model for low bit rate speech coding”, IEEE Trans. Speech and Audio Processing, vol.3, no.4, pp.242-250, Jul. 1995.

[17] Yoshimura, T., Tokuda, K., Masukom,T., Kobayashi, T. and Kitamura, T., “Mixed excitation for HMM-based speech Synthesis”, Proc. of Eurospeech, pp.2259-2262, Sept. 2001.

[18] Aoki, N., Ifukube, T. and Takaya, K., “Implementation of MELP vocoder using lifting wavelet transform”, Proc. IEEE Region 10 Conf. TENCON, pp.194-197, Sept. 1999.

[19] Alías, F., Monzo, C. and Socoró, J.C., “A Pitch Marks Filtering Algorithm based on Restricted Dynamic Programming”, Proc. of Interspeech, 2006.

[20] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T. and Kitamura, T., “Duration Modeling in HMM-based Speech Synthesis System”, Proc. of ICSLP, vol.2, pp.29–32, 1998.

[21] Iriondo, I., Socoró. J.C., Formiga, L., Gonzalvo X., Alías F. and Miralles P., “Modeling and estimating of prosody through CBR”, Proc. of JTH, 2006. (In Spanish)

[22] http://www.salle.url.edu/tsenyal/english /recerca/areaparla/tsenyal_software.html

[23] Lambert, T. and Breen, A., “A database design for a TTS synthesis system using lexical diphones”, Proc. of Interspeech, pp.1381–1384, 2004.

[24] Llisterri, J. and Mario, J.B., “Spanish adaptation of SAMPA and automatic phonetic transcription”, UPC, ESPRIT PROJECT 6819 (SAM-A Speech Technology Assessment in Multilingual Applications), 1993

[25] Black, A. and Hunt, A., “Generating F0 contours from ToBI labels using linear regression”, Proc. of ICSLP, vol 3, pp.1385–1388, 1996.

[26] Atserias, J., B. Casas, E. Comelles, M. Gonzlez, L. Padr and M. Padr, “FreeLing 1.3: Syntactic and semantic services in an open-source NLP library”, Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC’06). Genoa, Italy. 2006.

[27] Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T. and T. Kitamura, “Speech parameter generation algorithms for HMM-based speech synthesis”, Proc. of ICASSP, pp.1315-1318, 2000.

[28] Toda, T. and Tokuda, K., “A Speech Parameter Generation Algorithm Considering Global Variance for HMM-Based Speech Synthesis”, IEICE Transactions, Vol. E90-D, No. 5, pp.816–824, 2007.

[29] Y.-J. Wu and R.-H. Wang, “Minimum generation error training for HMM-based speech synthesis”, Proc. of ICASSP, pp.89–92, 2006

[30] Lampert, A., “Evaluation of the MU-TALK Speech Synthesis System”, ICT Report, 2004

[31] Iriondo, I., Alías, F., Sanchis, J., Melenchón, J., “A Hybrid Method Oriented to Concatenative Text-to-Speech Synthesis”, Proc. of Eurospeech, vol. 4, pp.2953–2958, 2003.

[32] Barros, M. J., Maia, R., Tokuda, K., Freitas, D. and Resende Jr., F. G., “HMM-based European Portuguese Speech Synthesis”, Proc. of Interspeech, pp.2581-2584, 2005.

Inventory of Intonation Contours for Text-to-Speech Synthesis

Tetyana Lyudovyk, Valentyna Robeiko

International Research/Training Center for Information Technologies and Systems,
Kyiv, Ukraine

{tetyana_lyudovyk, robeiko}@uasoiro.org.ua

Abstract

This paper presents an intonation model which determines intonation contours over intonation phrases. The model is described by four elements: communicative type of an intonation phrase; number of accent groups in it; position of the nuclear accent group in it; and set of target intonation points. Individualization of the model is based on semi-automatic analysis of speaker database. The model was implemented in unit selection TTS system for Ukrainian.

1. Introduction

Modeling of intonation contributes to speech science by introducing clarity and defining more precisely the intonation system of a language. Adequacy of an intonation model can be tested by imposing it on synthetic speech.

The goal of modeling consists of analysis and generalization of intonation phenomena and their representation in a parametric form that is compact yet preserves the naturalness of intonation in the synthesized speech.

There is no generally accepted intonation model. Intonation models in TTS systems have varied from rule-based models derived from expert knowledge to data driven statistical models.

General disadvantage of rule-based methods of intonation modeling consists in their inflexibility and insufficient account of individual speaker peculiarities.

Data-driven methods of intonation modeling often do not promote the understanding of linguistic phenomena. Using empirically composed large sets of features with wide range of values sometimes makes modeling "blind", because the relative importance of different features is unknown [1].

Our approach to intonation modeling combines rule-based and data-driven methods by defining general set of intonation contours and a procedure of individualizing these contours based on the automated analysis of speech data.

The intonation model described in this paper has been realized in the TTS system for Ukrainian [2, 3].

2. Stages of general intonation model development

Speech communication is based on general models common for all people speaking a particular language.

In this work intonation modeling is based on the assumption that the intonation serves primarily the communicative function. Intonation can be understood as the systematic use of pitch for communication [4]. P. Taylor notes one of the reasons why good models of prosody have proved hard to develop is that researchers have often tried to study prosody without reference to its communicative function.

Three stages of intonation analysis have been carried out. First, we began with the acoustic-phonetic study of ten non-annotated speech corpora and six prosodically annotated speech databases of different speakers to examine Lobanov's intonation model.

2.1. Lobanov's model of intonation

Lobanov's intonation model [5] has been successfully used for a long time in TTS systems for Russian. According to this model, the minimal intonation unit is the Accentual Unit (AU), consisting of one or more words, having only one fully stressed syllable. An AU, in its turn, consists of the nucleus (the fully stressed syllable), the pre-nuclear part (all the phonemes preceding the fully stressed syllable) and the post-nuclear part (all the phonemes following the fully stressed syllable). Phonemic content and number of syllables in the pre- and post-nucleus do not influence significantly the intonation contour of a certain type of phrase intonation.

Phrase intonation is characterized by:

- phrase type (finality, non-finality, interrogation, exclamation etc.);
- number of AUs.

For example, a declarative phrase composed of 4 AUs is marked as F-4. The last AU in a phrase is considered the prominent one, because usually the distinct intonation movement is associated with a phrase end.

Each AU is described by a set of target intonation points, which determine F0 values. F0 values between these target points are calculated by means of linear interpolation.

The Lobanov's model has two significant advantages. The first one concerns the distinction between informationally important (nucleus) and non-important (pre-nucleus and post-nucleus) portions of an AU. The second advantage concerns the detailed description of the nucleus by six target intonation points, which allows to model slight but categorical details and thus contributes to the natural quality of generated intonation.

2.2. Preliminary study of Ukrainian intonation

The acoustic-phonetic study of 10 speech corpora revealed general regularities of Ukrainian intonation. Table 1 presents analyzed speech material. Special attention has been paid to the comparative analysis of material obtained from different speakers reading the same text.

Six speech databases have been created under the framework of unit selection speech synthesis on the basis of the six speech corpora: two from male voices (isolated

Table 1. Speech material analyzed at the preliminary stage

Speaker	Text type	Number of intonation phrases with neutral intonation				Number of intonation phrases with logical or emphatic accent
		Finality	Non-finality	Question	Exclamation	
Svyatoslav	isolated sentences	267	268	3	14	114
Olexandr	isolated sentences	5	6	—	17	—
Yuriy	isolated sentences	4	7	—	9	—
Larysa	isolated sentences	11	13	—	13	—
Dmytro	radio news	15	31	—	—	9
Anzhelika	radio news	11	4	—	—	—
Viola	radio news	14	12	—	—	—
Valentyna	instructions	18	8	15	—	6
Mykola	radio interview	—	—	23	—	—
Maryna	isolated sentences	74	78	7	6	3

sentences and radio news) and four from female voices (isolated sentences, radio news, and instructions).

Experiments with the TTS system for Ukrainian have shown that Lobanov's intonation model can be successfully used under the unit selection framework (earlier we used this model with a formant synthesizer).

2.3. Correction of intonation model

2.3.1. Units of intonation

Intonation phrase (syntagm) is considered the basic intonation unit. Intonation phrase (IP) is divided into accent groups (minor phrases). An accent group (AG) consists of one or more words united by one accent.

Pre-nuclear, nuclear and post-nuclear parts can be distinguished in an IP intonation contour, each of them carrying a different functional load. The nuclear part (nuclear, main, prominent AG) is an intonation center of an IP. It has a distinct F0 contour which allows to differentiate communicative types (discourse situations). Pre-nuclear and post-nuclear parts of an IP are optional.

The intonation model determines the intonation contour over the whole IP, rather than over syllables [6], or over words or phonemes. The intonation contour is represented by a sequence of target F0 points.

2.3.2. Model elements

Investigation of the speech material and experiments with synthesized speech shown poor intonation modeling results for IPs with logical or emphatic accent placed on any AG other than last. This led to the inclusion of one more element into the intonation model, namely position of a prominent AG in an IP.

The general intonation model determines the intonation contour over the whole intonation phrase and is described by four elements [3]:

- communicative type of an IP;
- number of AGs in an IP;
- position of the prominent (main, nuclear) AG in an IP;
- set of target intonation points.

The number of AGs is determined by the number of accented vowels in the IP. The position of the nuclear AG corresponds to the position of the last AG if there is no logical or emphatic accent. Otherwise the position of the nuclear AG corresponds to the position of the AG carrying the logical or emphatic accent.

Target intonation points determine F0 values. The accent center (nucleus) of an AG is its accented vowel. It is modeled by 6 target points. The part of an AG preceding the accent center is modeled by 2 target points. The same applies to the succeeding part of an AG. Thus, an IP with 3 AGs is described by 30 intonation points; an IP with 4 AGs is described by 40 intonation points, etc. F0 values between target points are calculated by means of linear interpolation.

2.4. Deeper study of intonation and testing of the intonation model

Aiming at reflecting the full range of communicative functions in synthesized speech, we selected for our work a Ukrainian fiction text with dialogues (80 minutes) read by the professional male speaker Valeriy.

2.4.1. Recordings and database development

The recording sessions were not monitored. In fact, a speaker received an orthographic text, made recordings in a quiet room within one day (two sessions), and supplied these recordings to the researchers. A speech database containing 18785 units (phones-in-context) was developed with manual correcting of automatically obtained transcription and segmentation into phones.

Stressed and unstressed vowels are treated as different phonemes.

The manual correction of segmentation assured appropriate pitch synchronous boundaries between phones. Segmentation of units into pitch periods was carried out automatically. Unvoiced phones were not segmented.

2.4.2. Intonation annotation in the speech database

The database annotation contains no high-level linguistic information nor symbolic prosodic labels like ToBI [1].

To annotate intonation, we rely only on objective low-level numerical feature that is the pitch period length. We claim that the sequence of pitch period lengths during an intonation phrase is the best intonation description independent of any intonation theory.

Borders between intonation phrases and between words are unmarked. We found that it was incorrect to mark borders relying on corresponding orthographic text (as in [7]), because the speaker often violates syntactic structure and ignores punctuation marks like question marks or points. To mark borders automatically without mistakes relying on acoustic cues (e. g. pauses) is also incorrect, because sometimes intonation phrases are not divided by a pause (3 % of borders between intonation phrases), while often there is an inner pause within an intonation phrase (25 %). Sometimes the speaker feels the need to place phrase breaks at equal intervals somewhat independently of the top down linguistic structure [4].

The analyzed speech material contains many cases of prominence, which are difficult to mark automatically.

3. Communication types of intonation phrases

Our goal is to analyze prosodic annotations of a speech database and to create an inventory of intonation contours related to this database and thus relative to the speaker intonation.

Two of the intonation model elements (number of AGs in an IP and set of target intonation points) can be found easily given the database annotation supplemented with breaks between IPs. But the other two elements, the position of the prominent AG in an IP and, most importantly, the communicative type of an IP, turned out to be difficult to identify not only automatically but even by phoneticians.

We began with a list of 10 communicative types: finality, non-finality, wh-question, yes/no question, exclamation, contrast, explication, parenthesis words, expressive finality, and not-identified type. In many cases it was difficult to distinguish between types, for example between non-finality and contrast, or between exclamation and expressive finality.

Thus we carried out a perception experiment aimed at analyzing what communicative types are assigned by listeners to different intonation phrases from real speech. 20 listeners (students and professors of linguistic university, all native speakers of Ukrainian) were asked to listen to 49 IPs selected from the investigated speaker's recordings. Listeners were supplied with a list of 9 communicative types and the orthographic text corresponding to the recordings, where punctuation marks were absent and all the words were in lower case. The task was to indicate the communicative type of each IP. Each IP was played three times, and the experiment lasted 30 minutes.

This experiment helped correct the list of communication types present in the speaker's recordings. Thus, communicative types of explication and contrast have been excluded. On the contrary, 3 communicative type have been added: enumeration; attributive relative clause with a relative pronoun; and first part of complex wh-question.

Table 2 represents the resulting distribution of IPs contained in the investigated read fiction text.

3.1.1. Stylization

In order to compare IPs with different segmental structure and to derive invariant intonation contours the stylization of F0 tracks was performed.

Stylization consists in determining of F0 values at the target intonation points. Intonation contour of one-accent IP is described by 10 F0 values:

$$F0_1^{\,1}, F0_1^{\,2}, \ldots, F0_1^{\,10},$$

where $F0_1^{\,1}$ is the F0 value at the first not unvoiced phoneme (vowel or voiced consonant) of the AG;

$F0_1^{\,2}$ is the F0 value at the last not unvoiced phoneme of the AG among phonemes preceding the accent center;

$F0_1^{\,3}$ ... $F0_1^{\,8}$ are F0 values at the accent center;

$F0_1^{\,9}$ is the F0 value at the first not unvoiced phoneme among phonemes succeeding the accent center;

$F0_1^{\,10}$ is the F0 value at the last not unvoiced phoneme of the AG.

Intonation contour of an IP divided into n AGs is described by $10n$ F0 values:

$$F0_1^{\,1},\ F0_1^{\,2}, \ldots,\ F0_1^{\,10},\ F0_2^{\,1},\ F0_2^{\,2}, \ldots,\ F0_2^{\,10}, \ldots,\ F0_n^{\,1},\ F0_n^{\,2}, \ldots,\ F0_n^{\,10}.$$

Table 2. Distribution of intonation phrases with different communicative types

Communicative type of intonational phrase	Total number of intonation phrases	Number of accent groups in an intonational phrase						
		1	2	3	4	5	6	7
neutral finality	350	49	94	113	67	18	4	5
expressive finality	207	11	57	51	37	39	7	5
non-finality	327	58	112	95	34	23	2	3
yes/no question	21	12	2	2	1	2	2	—
wh-question	20	—	6	10	3	—	1	—
exclamation	59	19	18	9	10	2	1	—
enumeration	17	5	8	—	4	—	—	—
parenthesis words	10	2	7	—	1	—	—	—
first part of complex wh-question	11	—	2	5	2	2	—	—
attributive relative clause with a relative pronoun	7	—	2	5	—	—	—	—
unidentified	26	8	8	7	3	—	—	—
total	1055	164	316	297	162	86	17	13

Figures 1 and 2 represent non-stylized and stylized intonation contours of the IP „А ви прислухайтесь," ("Lend your ear"), consisting of two AGs.

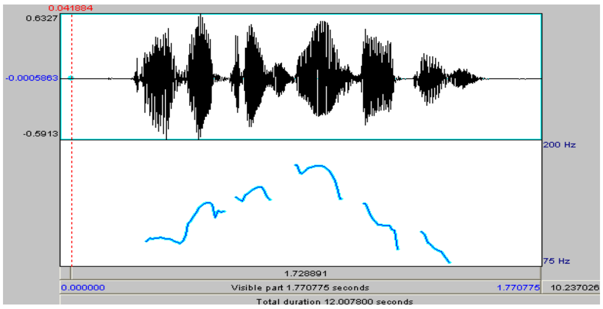


Figure 1: Oscillogram (top) and non-stylized intonation contour (down) of the intonation phrase „А ви прислухайтесь," ("Lend your ear") uttered by the speaker Valeriy.

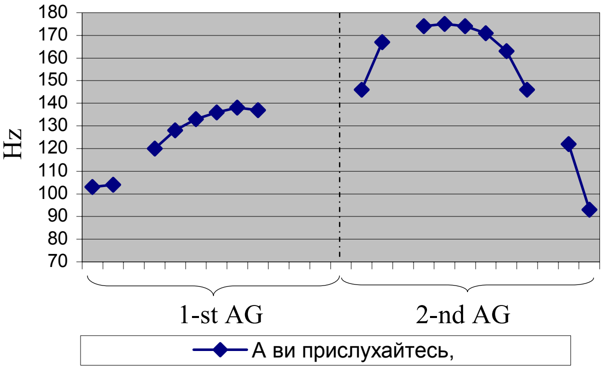


Figure 2: Stylized intonation contour of the intonation phrase „А ви прислухайтесь," ("Lend your ear") uttered by the speaker Valeriy.

Stylization allows comparing IPs with different segmental content abstracting from intonationally insignificant segments and microprosody influence, e.g. F0 change at consonants [8]. For example, there is an F0 lowering by ≈10 Hz (speaker Valeriy) and by ≈15-20 Hz (speaker Svyatoslav) at voiced fricatives. It is considered traditionally that F0 variation caused by segmental structure of speech segments are not perceived as intonationally significant. Experiments testify that F0 contour may be considerably simplified without a loss of intonation perception.

Now the stylization of intonation contours is performed in automated mode using speech database annotations containing the information about pitch periods lengths and, therefore, about F0 movement.

The most difficult non-automated stage of the stylization is the detection of IPs borders, because not all IPs are separated by pauses and not all pauses indicate such borders. We plan to analyze the dependence of the presence of IP borders on pause duration and on range and form of F0 contour at stressed vowels. First results in this direction allow us to automatically find potential IP borders.

3.1.2. *Classification*

The next step in the inventory of intonation contours deriving is classification of stylized intonation contours of all the IPs according to communicative types listed in Table 2.

Each communicative class is divided into sub-classes according to the number of AGs, and each sub-class is divided into sub-sub-classes according to the position of the prominent AG in the IP.

Each sub-sub-class is given a name consisting of three parts corresponding respectively to communicative type, number of AGs, and position of prominent AG: X_Y_Z.

The number of IP contours which intonation model can determine is equal to $l\sum_{n=1}^{m} n$, where l is the number of distinguished communicative types, m is the maximum number of AGs in an IP, and n is the position of prominent AG in the IP. Now we distinguish 10 communicative types. The maximum number of AGs in an IP is equal to 7. Then the proposed model can generate 280 different intonation contours.

We continued to investigate stylized F0 contours within each sub-sub-class, namely the direction of F0 movement (falling, rising or narrow). We discovered further subdivision into sub-types for finality (3 sub-types), non-finality (4 sub-types), and exclamation (3 sub-types). Figures 3, 4, and 5 represent averaged variants (regarded as models) of finality, non-finality and exclamation for IPs consisting of two AGs, the second one being the prominent one.

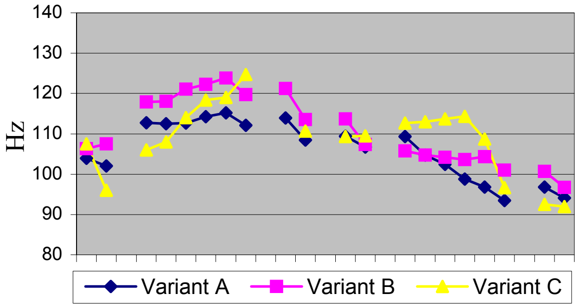


Figure 3: *Models for finality (speaker Valeriy).*

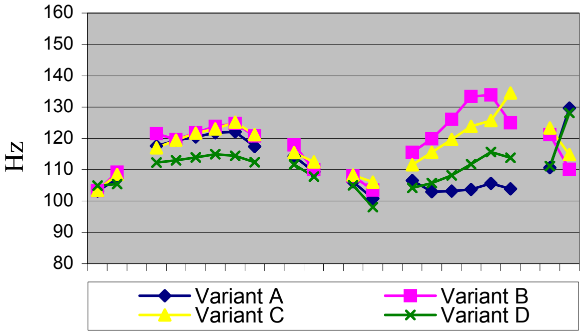


Figure 4: *Models for non-finality (speaker Valeriy).*

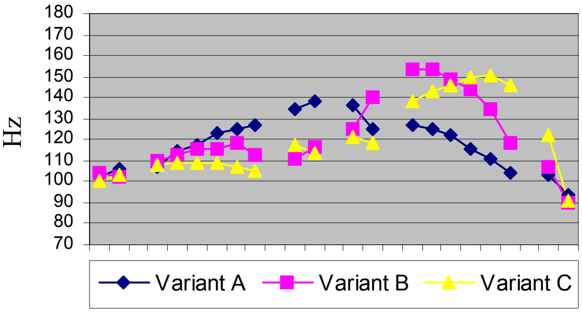


Figure 5: *Models for exclamation (speaker Valeriy)*

It should be added, that the proposed intonation model allows increase in the number of communicative types at the expense of a more detailed specification of communicative sense, e.g. differentiation between a proper question and a specifying question.

4. Individualization of Intonation Model

Synthesis of individualized speech implies the training of intonation model on speaker's data that is elaborating individual inventory of intonation contours differing in F0 range and shape. Training of the model is performed in semi-automatic way based on the annotation of the speaker's database. Intonation peculiarities other than F0 contours should be accounted for as well. In our case it concerns for example the insertion of an extra pause before the last AG of an IP. (This is characteristic of some actor's readings).

First, breaks between IPs are indicated, and then communicative sub-sub-classes are assigned. There is not enough knowledge at present to automate this step because neither the syntax structure nor even punctuation marks are helpful.

Second, the stylization of IPs according to the intonation model is carried out automatically on the basis of speech database annotation which contains the detailed description of F0 movement along each vowel and voiced consonant in the form of a sequence of pitch period lengths. Stylization consists in determining of F0 values at the target intonation points of an AG: two F0 values for pre-nucleus, six for nucleus and two for post-nucleus. Tables (where the rows correspond to IPs in sub-sub-classes, and the columns correspond to target points) and diagrams of intonation corespond are obtained and models of intonation contours are derived either by averaging or by selecting one representative contour.

Weak points of the described procedure are: the difficulty of automatic identification of an IP communicative type and prominent AG position; errors of automatic F0 values calculation; and lack of data for several communicative types.

The set of obtained intonation contours forms the individualized intonation model and describes the intonation of a particular speaker.

The challenge and the goal of the future work is to automate the process of breaks between IPs identifying, communicative type of an IP with indication of prominent AG marking, and variants of model contours within a set of IPs with equal communicative type, equal number of AGs and equal position of prominent AG discovering. For example, while speaker Valeriy has four variants of N_2_2 intonation contours (non-final IP with two AGs, the second being the prominent one), the speaker Svyatoslav has only three variants of N_2_2 contour. In Figure 6, averaged N_2_2 intonation contours of two speakers are presented.

We hope to advance in automation through detecting the correspondence between intonation contours on one hand and communicative nuances and lexical-syntactic structure of IPs on the other. For example, it is clear now that a change of a definite intonation contour occurs when the speaker is "uncertain" about what he is reading.

The most important yet difficult goal is to develop procedures for input text analysis with modeling of text interpretation by a particular speaker.

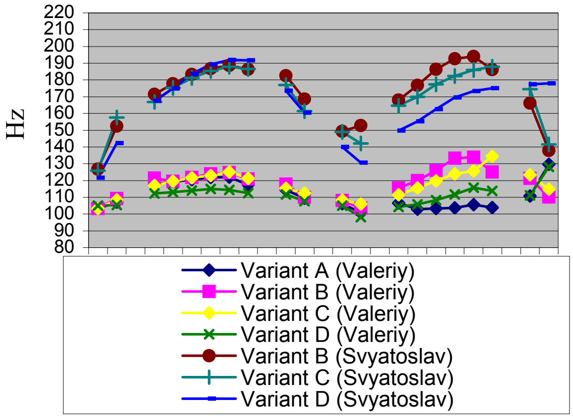


Figure 6: *Models of intonation contours for non-finality of two speakers Svyatoslav and Valeriy.*

5. Implementation in TTS system

Obtained individualized intonation models were used in the TTS system for Ukrainian.

Communicative type is assigned based on punctuation mark and some lexical cues. Sequences of F0 values corresponding to target intonation points, are calculated by the prosody generation module, and are characteristic of the speaker whose voice is used for synthesis. In unit selection module, the fundamental frequency is one of the main criteria of selection. Concatenation of selected phones with definite intonation is performed by the acoustic processor. Phone waves may be either modified according to calculated values of durations and F0 or concatenated as they are, without modification (pure unit selection).

6. Testing the intonation model

To test the intonation model incorporated in the TTS system, a formal listening test was carried out. 22 listeners (students and professors of linguistic university, specialists in Ukrainian language) were asked to listen to 60 synthesized passages containing IPs of 10 communicative types (Table 2). All the passages were taken from the Ukrainian translation of the Lewis Carroll's "Alice in Wonderland", because this text has a natural variety of prosody [1]. Test material was synthesized with Valeriy's voice.

Listeners were supplied with a list of 10 communicative types and the orthographic text corresponding to the synthesized passages, where punctuation marks were absent and all the words were in lower case. The task was to indicate the communicative type of each IP. Each passage of synthesized speech was played three times, and the experiment lasted 30 minutes.

The results are presented in Figure 7. The communicative type recognized the best was enumeration (89 %). This corresponds to the results of our experiment with real Valeriy's speech. Then the listeners noted that this speaker had a distinct intonation contour for enumeration. So, it was not difficult to implement this contour in our TTS system. On the contrary, there is no big difference between contours of finality, expressive finality and exclamation, which is reflected in corresponding recognition results. The poor recognition of "first part of complex wh-question" is due to the fact that some listeners judged the whole questions and recognized them as wh-questions.

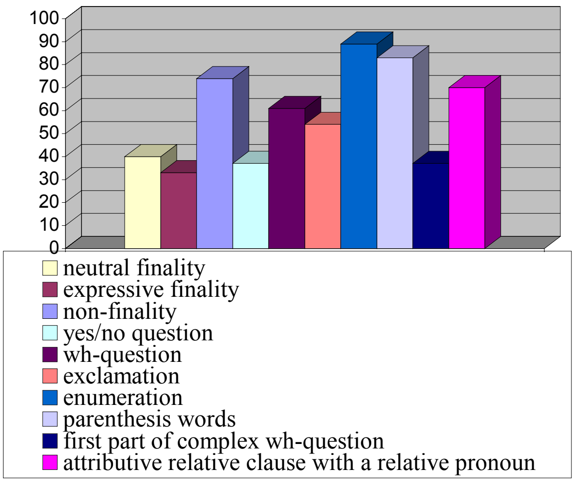


Figure 7: *Results of IPs communicative type recognition (%).*

Table 3 represents recognition results of some IPs not containing lexical cues (in Ukrainian). Communicative type "parenthesis words" was recognized by all listeners in all corresponding IPs.

Table 3. *Results of IPs communicative type recognition*

IPs (English equivalent for commodity)	Imposed communicative type	Recognized as
You'll see me there	expressive finality	expressive finality
Yes	expressive finality	neutral finality
Poor Alice	exclamation	exclamation
there's no use in crying like that	exclamation	expressive finality
Alice felt	non-finality	non-finality
I shall have to ask	non-finality	first part of wh-question
all dripping wet, cross	enumeration	enumeration
White (Rabbit) with pink eyes	enumeration	relative clause
pulling me out of the window	yes/no question	yes/no question
Not like cats	yes/no question	non-finality

7. Discussion

This work revealed the communicative polysemantics of information contained in texts to be read. Readers interpret texts according to situation, to audience and even to their own character. We studied several cases when, for example, some speakers break an IP in AGs and others do not, some add prominence, others do not while reading the same text. Thus, while synthesizing speech we have to model the reading of a text by a specific speaker.

Now the individualized analysis of input texts is not performed to the full extent. Decision about the communicative type of an IP is made based on punctuation marks and some lexical cues. Then a corresponding individual intonation model trained on speaker data is applied to create a speaker "tailored" target specification of an input text.

The attached audio file ("Alice.wav") presents the synthesized beginning of the Ukrainian version of "Alice in Wonderland".

8. Conclusions

The presented work concentrates on the study of intonation in communicative aspect. The full range of communicative types present in a large speech corpus was investigated. The presence of complex sentences allowed to discover specific types associated with parts of questions and attributive relative clauses with relative pronouns.

The derived intonation model based on communicative types and its individualization based on semi-automatic analysis of speaker data were implemented in unit selection TTS system for Ukrainian and tested during a formal listening test. The results testify that the listeners identify the communicative types of synthesized utterances.

The proposed model allows to synthesize speech in different styles (e.g. neutral and expressive) using the same speech database but different intonation contours (e.g. neutral and expressive finality, non-finality, questions, etc.).

It should be noted also that the synthesis technology under the framework of which the proposed intonation model is used, may be applied to languages other than Ukrainian. Similar approach to intonation modeling is used in [5] for Russian and Polish.

9. References

[1] Strom, V., Clark, R., King, S., "Expressive Prosody for Unit-selection Speech Synthesis", Proc. of INTERSPEECH 2006 – ICSLP, pp. 1296-1299.

[2] Lyudovyk, T., Sazhok, M., "Unit Selection Speech Synthesis Using Phonetic-Prosodic Description of Speech Databases", Proc. of International Conf. "Speech and Computer" (SPECOM'2004), St.-Petersburg, Russia, 2004, pp. 594-599.

[3] Lyudovyk, T., "Linguistic Processor Training on Speaker Data for Unit Selection Text-to-Speech", Proc. of International Conf. "Speech and Computer" (SPECOM'2006), St.-Petersburg, Russia, 2006, pp. 315–320.

[4] Taylor, P., "Text-to-Speech Synthesis", Manuscript, http://mi.eng.cam.ac.uk/~pat40/ttsbook_draft_2.pdf.

[5] Lobanov, B., Tsirulnik, L., Zhadinets, D., Karnevskaya, H., "Language- and Speaker Specific Implementation of Intonation Contours in Multilingual TTS Synthesis", Speech Prosody: Proc. of the 3rd International conference, Dresden, Germany, May 2-5, 2006, V. 2, pp. 553-556.

[6] Clark, R. A. J., King, S., "Joint Prosodic and Segmental Unit Selection Speech Synthesis", Proc. of INTERSPEECH 2006 – ICSLP, pp. 1312-1315.

[7] Colotte, V., Beaufort, R., "Linguistic features weighting for a Text-To-Speech system without prosody model", Proc. of INTERSPEECH 2005, pp. 2549-2552.

[8] Mertens, P., "The Prosogram: Semi-Automatic Transcription of Prosody Based on a Tonal Perception Model", Proc. of Speech Prosody 2004, Nara (Japan), 23-26 March, pp. 549-552.

Analysis Methods for Assessing TTS Intelligibility

H. Timothy Bunnell and Jason Lilley

Center for Pediatric Auditory and Speech Sciences
Alfred I. duPont Hospital for Children, Wilmington DE, USA
and
Department of Linguistics and Cognitive Science
University of Delaware, Newark DE, USA
{bunnell,lilley}@asel.udel.edu

Abstract

Semantically unpredictable (SU) sentences are often used to assess intelligibility of TTS systems, but analyses of listener responses to SU sentences can be a labor-intensive process. In this paper we compare several approaches to the analysis of data from an SUS task. Data from a study comparing five TTS systems were analyzed in a variety of ways ranging from string edit measures based on carefully hand-corrected phonetically transcribed responses to largely uncorrected words- or sentences-correct measures. Results suggest that a simple sentences-correct measure is adequate when only rank order information is of interest. However, the sentences-correct measure masks the magnitude of differences between systems and should be avoided when it is important to gage how large the difference in intelligibility is between systems. In preparing response data for analysis, careful human interpretation of listener response data can lead to higher intelligibility measures overall, but does not interact with TTS system or other factors and consequently does not lead to different conclusions when comparing multiple TTS systems. This suggests that largely automated scoring procedures are feasible.

1. Introduction

The use of syntactically well-formed but semantically anomalous sentences in assessing TTS systems was first described in [1]. More recently, [2] describe procedures for generating semantically unpredictable sentence (SUS) materials for evaluating TTS intelligibility. In [2], based on an earlier study ([3]), the recommended analysis procedure is to score whole sentences as either correct or incorrect, requiring every word of the sentence to be correct and in the correct sequence for a sentence to be scored as correct.

While the scoring procedure recommended in [2] is simple to implement, it is very strict (leading to generally lower measures of intelligibility for any given TTS system), and relatively coarse. Concerns with such a coarse measure include the opposing possibilities that it may either (a) mask relatively serious differences, or (b) amplify relatively subtle differences between two TTS systems, making one system appear to be much more or less intelligible than another. The former could happen, for example, when comparing a TTS system that makes about 2 errors per sentence to a system that makes 10 errors per sentence. The latter could happen when comparing a TTS system that makes phonetically subtle errors with relatively higher frequency to a system that makes gross pronunciation errors with somewhat lower frequency. In such cases, it is possible that scoring responses at a more fine-grained level would yield different intelligibility rankings of TTS systems, or would provide a more accurate measure of the differences between systems than would responses scored at the sentence level.

A second and more general concern related to analysis of SUS response data is the question of how to interpret ambiguous response data, and whether interpretation of ambiguous data influences conclusions to be drawn from a study using SUS material. This is particularly an issue when, as in the study described here, listeners respond by typing their responses into a computer. Typed responses contain a variety of errors. Some, such as simple typos and spelling errors, are of little interest and are presumed to be randomly distributed with respect to the synthesizers being compared. However, such errors may result in responses that are probably correct to be scored as incorrect. If using a between-subjects design, differences in the spelling or typing ability of subjects across groups could artificially increase or decrease real differences between TTS systems. On the other hand, other errors, such as attempts to “phonetically” gloss tokens perceived as non-words, are indicative of real intelligibility problems and may yield valuable information about the strengths and weaknesses of the system under study.

In the following, we explore the consequences of some of these factors on a set of data collected to compare five synthesis systems.

2. Method

2.1. Dataset

The perception experiment in which the current data set was collected was briefly described in [4]. The study was intended to compare the intelligibility of a new TTS system to four existing commercially available systems. A more complete description of that study is in preparation. Here we outline the overall study design to lay out the structure of the data collected.

2.1.1. Subjects

The subjects were 30 University of Delaware students who received a $10 gift card to a local bookstore in exchange for participation. All listeners were native speakers of American English and reported having normal hearing.

2.1.2. Stimuli

The stimuli were 100 SU sentences generated by each of the five TTS systems. Since this report is not concerned with the specifics of the TTS systems, they will be referred to simply as systems A, B, C, D, and E.

Per recommendations in [2] the SU sentences were constructed using words of minimal length (all one-syllable) within five distinct sentence frames. Examples of sentences generated for each of the five syntactic frames are shown in Table 1.

Table 1. Examples of each sentence frame. Words in italics are randomly assigned within the frame represented by words in normal font.

FRAME	EXAMPLE SENTENCE
1	The *trip talked* in the *old stage*.
2	The *state spared* the *claim* that *wept*.
3	The *thin aid brushed* the *part*.
4	Why does the *strength trust* the *dark sound*?
5	*Waste* the *shape* or the *hand*.

Synthetic renderings of all sentences were generated by each of the five TTS systems. Because all five synthesizers were sufficiently SAPI-compliant to be installed on the same Windows computer, sentences were generated directly to waveform files for storage and later presentation.

To reduce the possible effect of amplitude differences inherent to the five synthesizers, all synthetic speech files were adjusted to 72.0 dB RMS amplitude (calculated over the entire synthetic speech file). For all synthesizers, speaking rate and average F0 were left at default levels (in some cases, these were not adjustable). While consistent differences in speaking rate (as measured by raw waveform duration) existed between synthesizers, the differences were not perceptually prominent. The overall average sentence duration was 2.2 seconds and varied from 1.9 seconds (system B) to 2.4 seconds (system E).

2.1.3. Procedure

The five hundred synthetic sentences (100 sentences by 5 synthesizers) were split into five sets for presentation. Each set contained 20 sentences of each syntactic frame. Of the 20 sentences of each frame, four sentences were produced by each of the five synthesizers. This blocking ensured that each trial set of 100 sentences contained an equal number of sentences of each syntactic frame produced by each synthesizer without any duplication of sentences. Each listener was assigned to one sentence set and, hence, was never presented with the same sentence twice. In all, six listeners were assigned to each of the five sentence sets.

Listeners wore headphones and were seated at a computer in a quiet room. After hearing a sentence one time, listeners were provided as much time as necessary to type the sentence as they understood it. When finished, listeners clicked a *Next* button to initiate the next trial.

2.2. Data Reanalysis

The originally reported analysis of data from this experiment was based on the number of content words correct in each sentence [4]. For the present analysis, we have reanalyzed all data using measures based on the edit distance between the listener responses and the original sentences. The edit distance between two strings is defined as the minimum total "cost" of transforming one string into the other using insertion, deletion, and substitution operations, each operation being associated with its own cost. For the present analysis, the costs of all operations were set to one, so that the edit distance is simply the total number of insertions, deletions, and substitutions. The same general measure can be used whether the strings being compared are strings of discrete word tokens or discrete phone tokens.

The various edit distance measures reported are based on both word-level and phone-level measures. In both cases, we started by designing a dictionary to map both stimuli and listener input onto response tokens. The raw listener input (typed sentences) and stimulus sentences were first tokenized into a set of input word tokens, where a word token is defined as an uninterrupted sequence of alphabetic characters or apostrophes.

For word-level analyses, the response tokens in the dictionary were usually exactly the same as the input tokens, but in some cases, the dictionary would map multiple possible input tokens onto a single response token. For instance, the input tokens *rows*, *rose*, and *roze* were all mapped to the token *rose* (the word form given to the TTS systems in generating the sentence).

For phone-level analyses, a similar dictionary was used to map input tokens onto strings of phonetic symbols, including a word boundary symbol. For this experiment, two phone-level dictionaries were created, an “uncorrected” one and a “hand-corrected” one. The former was generated by running the tokenized listener input through the letter-to-sound rules of one of the TTS systems with a bare minimum of additional hand editing. For example, the default TTS pronunciation of nonce forms was used unless the system chose to spell out the form. In the latter case, an experimenter-supplied pronunciation was used. The “hand-corrected” dictionary was created by further editing the “uncorrected” dictionary, and attempting to interpret the intention of the respondent. For example, all nonce forms were corrected if the automatic transcription did not agree with the experimenter’s interpretation of what the listener intended. When the respondent's intention could not be determined completely, a transcription that would result in the best match between stimulus and response was used.

In addition, strings of phonetically uninterpretable listener input (e.g., a string such as *????*) were mapped onto a word boundary symbol with no other phonetic content. This approach allowed us to retain word boundary location information to the extent that it was recoverable from the response data.

Once all sentences were mapped, the word-level and phone-level edit distances between each pair of stimulus and response sentences were computed. For the phone-level edit distances, we chose to disregard several phonetic differences that were represented in the TTS symbol set we used. Specifically, the difference between front and back schwa was ignored, as was the distinction between a flapped /d/ and either an unaspirated /t/ or a full /d/. We also disregarded the distinction between aspirated and unaspirated voiceless stops. Additionally, in computing edit distances, word boundaries could be inserted or deleted, but they could not be substituted with other segments.

Finally, a sentence-level error score (1 = incorrect; 0 = correct) was also computed for each response sentence. A response sentence was scored correct only if the phonetic edit distance between it and the stimulus sentence was zero.

3. Results

To analyze the data, scores were derived by summing edit distances or sentence errors over the four sentences of each frame type from each synthesizer per listener. This resulted in 25 scores (5 frames X 5 synthesizers) per subject that we treated as a completely within-subjects design. Preliminary analyses revealed that the between-subjects factor SENTENCE SET (as described in 2.1.3) was not significant, and consequently it will not be discussed here.

So that sentence, word, and phone-level data were comparable, raw scores for each level were divided by the number of sentences, words, or phones within the sentences from which the score was derived. The resulting proportion data were highly non-normal in their distribution and were consequently arcsine transformed to improve their suitability for analysis of variance. All analyses of variance described below were conducted using the arcsine transformed data; however, only the original proportions are presented in figures.

3.1. Sentence-level scoring

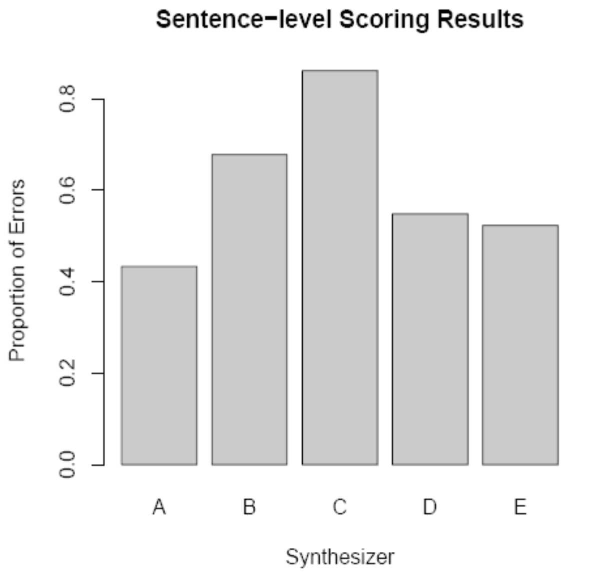


Figure 1: Overall ranking of synthesizer intelligibility when scored at the sentence level.

Sentence-level scoring produced results that closely resembled those originally reported in [4]. Overall, the main effect of synthesizer was significant (F[4,116] = 72.15, p < .001) as was the effect of sentence frame (F[4,116]=22.77, p < .001) and the interaction of synthesizer with sentence frame (F[16,464]=3.79, p < .001). Figure 1 displays the means underlying the significant main effect of synthesizer. System A clearly has the lowest error rate and system C the highest. Post hoc tests reveal that all differences among synthesizers except for the difference between systems D & E are significant.

This main effect of synthesizer was conditioned by a significant interaction with sentence frame, indicating that the relative ranking of synthesizers varied significantly over the various syntactic frames used in the study. Figure 2 illustrates this effect by plotting the individual synthesis systems as groups of bars within each sentence frame. As this figure shows, error rates tended to be lowest for frame 5 (the shortest frame) and are most representative of the overall results. System C has the highest error rate in all frames, and system A has the lowest error rate in all but one sentence frame. Systems B, D, and E tended to vary more in intelligibility as a function of the sentence frame.

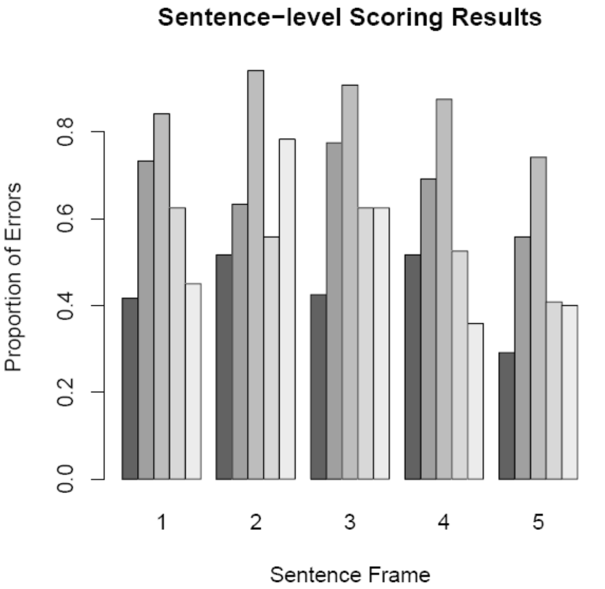


Figure 2: Means for interaction of synthesizer and sentence frame for sentence-level scoring. The shaded bars in each grouping represent the

3.2. Word-level scoring

The pattern of significant effects for the word-level analysis of variance mirrored the pattern for the sentence-level analysis with effects for synthesizer, sentence frame, and their interaction all significant (F[4,116]=161.25; F[4,116]=42.63; and F[16,464]=4.05 respectively, all p's < .001).

The means underlying these significant effects also patterned similarly to those from the sentence-level analysis. As with the sentence-level analysis, system A had the lowest overall error rate and system C the highest (see Figure 5). Systems D and E remained statistically equivalent, although error rates were slightly higher for system E than for system D, a reversal of the order seen in the sentence-level analysis. Two other differences are worth noting. First, as expected, there was an overall lower proportion of errors for all systems. Overall, 60.9% of the sentences in the sentence-level analysis contained errors. However, only 17.9% of the words were incorrectly identified in listener responses. Another noteworthy difference between the sentence-level and word-level analyses is revealed by considering the magnitude of the difference between the synthesizer with the highest error rate and the system with the lowest error rate. For the sentence-level analysis, system C had an error rate (86.2% of the listeners' response sentences had errors) about twice the magnitude of the error rate for system A (43.3% of the listener responses to sentence from system A had errors). By contrast, for the word-level analysis, the error rate for responses to system C (34.6%) was more than three times that of responses to system A (10.1%).

3.3. Phone-level scoring

We turn next to results from analysis of the phonetic edit distance measure. For this analysis we used the edit distances obtained using the largely uncorrected dictionary. Once again, the main effect of synthesizer was significant (F[4,116]=155.35, p < .001) as was the effect of sentence frame (F[4,116]=23.85, p < .001) and the interaction of sentence frame with synthesizer (F[16, 464]=3.30, p < .001).

Figure 3 shows the means underlying the main effect of synthesizer. Comparing this to Figure 1, it is clear that the relative ranking of TTS intelligibility is unchanged. However, the differences between the poorest and best systems are enhanced by using the edit distance measure. Thus, while Figure 1 shows that slightly more than 40% of the sentences for system A contained some error, Figure 3 shows that on average, these were due to errors on only about 4% of the phonetic segments within those sentences. Also of note once again is the relative number of errors on system C versus system A. At the phone level, listeners made more than 4 times as many errors transcribing utterances for system C compared to system A.

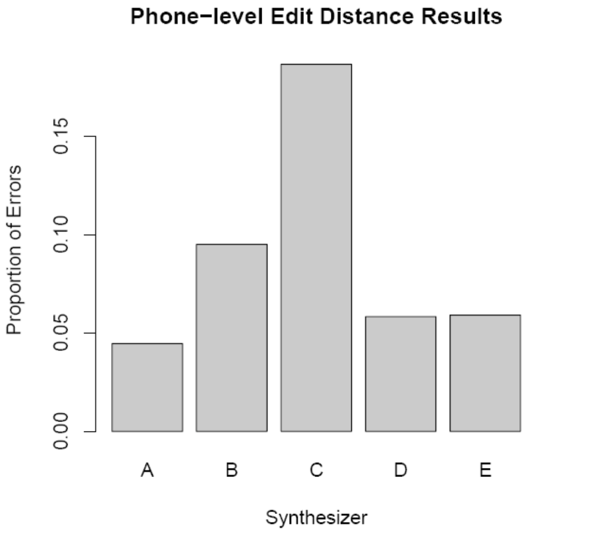


Figure 3: Overall ranking of synthesizer intelligibility when measured as phone-level edit distance.

The significant interaction between synthesizer and sentence frame for phonetic edit distance is illustrated in Figure 4. By comparison to Figure 2, we can see that differences between the systems are generally enhanced. While system C has the highest error rate in all sentence frames, there is greater variability among the other systems in Figure 4. For instance, in simple rank order, system A has the lowest error rate in 3 of the 5 frames which system E has the lowest rate in 2 of the five. As in the sentence-level analysis, however, system A is least variable over the five sentence frames.

3.4. Combined multi-level analysis

To further verify the impression that results from analyses at each level of analysis are qualitatively different, an additional analysis of variance was calculated combining data from all three levels of analysis as an additional within-subjects factor. Results of this analysis are given in Table 2 and Figure 5. As the ANOVA results shown in Table 2 indicate, level of analysis (LOA) was a significant main effect and participated in significant interactions with both synthesizer (SYN) and sentence frame (FRM).

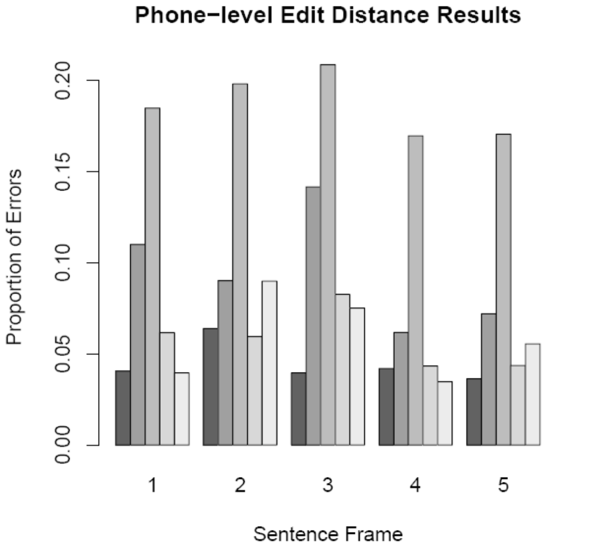


Figure 4: Means for interaction of synthesizer and sentence frame for phone-level edit distance. The shaded bars in each grouping represent the means for each synthesizer (A – E in alphabetical order) within each sentence frame.

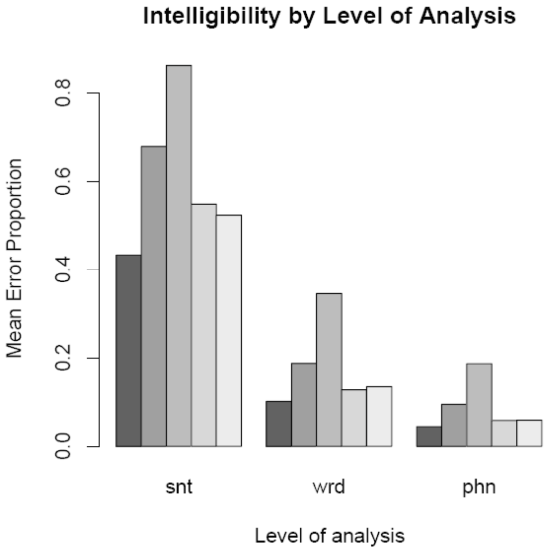


Figure 5: Comparison of mean proportion of errors for each synthesizer (shaded bars) at each level of analysis.

Table 2: Summary table from multi-level ANOVA. All terms are *p* < .001.

Term	Df	SSQ	MSQ	F
LOA	2	187.1	93.6	2714.6
SYN	4	34.4	10.9	121.8
FRM	4	7.7	1.9	29.4
LOAxSYN	8	6.8	0.9	30.9
LOAxFRM	8	2.7	0.3	17.4
SYNxFRM	16	5.2	0.3	4.1
LOAxSYNxFRM	32	2.1	0.1	3.3

3.5. Hand correction

To determine the consequences of carefully hand-correcting the phonetic transcriptions of listener response data, phonetic edit distances computed from hand-corrected versus uncorrected dictionaries were compared in an analysis of variance using synthesis system, sentence frame, and correction as factors in a 5 x 5 x 2 design. As expected from all the previous analyses, this analysis revealed significant main effects of synthesizer and sentence frame as well as a significant interaction of sentence frame and synthesizer. There was also a significant main effect of correction, with carefully corrected transcriptions having overall lower edit distances than did uncorrected data (F[1,29]=41.72, $p < .001$). Crucially, correction did not interact with any other factor. Thus, while hand correction of phonetic transcriptions for listener responses did result in lower edit distances overall, it had no further consequences for interpreting the results.

3.6. Study size

Study size can be varied either by changing the number of listeners involved, or by changing the number of stimuli per condition that are presented to each listener. In the latter case, reducing the number of stimuli per condition can reduce the total amount of time required to run a study, or allow more conditions to be explored with the same total number of stimuli. Reducing the number of listeners can also reduce the amount of time required to run a study (reducing cost if listeners are paid), or if listeners are grouped in different conditions, allow more conditions to be explored with the same total number of subjects.

We simulated the relative costs of reducing the number of subjects per condition versus reducing the number of stimuli by repeatedly randomly discarding 50% of the subjects or sentences in a balanced manner, and recalculating the results based on the randomly selected subset. Results from 50 such simulated smaller experiments are presented in Figure 6, which shows boxplots for the results when subjects are randomly discarded (left panel) and when sentences are randomly discarded (right panel). Each boxplot represents the median error proportion (horizontal line in each box), the interquartile range (box vertical extent), and the full range of results (whiskers) sans data points identified as outliers (circles). The amount of variability (as indicated by interquartile range) is clearly larger for simulations based on discarding sentences than for simulations based on discarding subjects.

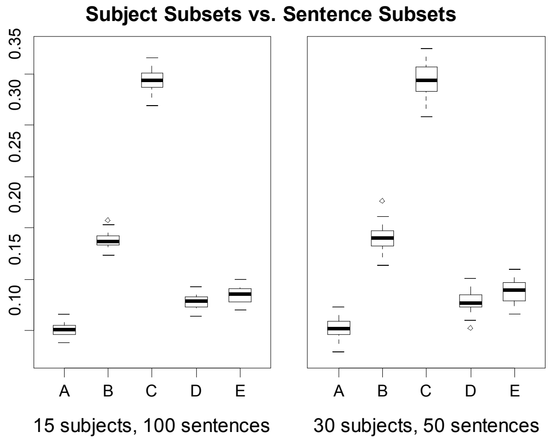


Figure 6: Comparison of simulations using a reduced number of subjects (left panel) versus a reduced number of stimuli (right panel).

Given that it is less costly—in terms of experimental power—to reduce listeners than to reduce sentences, we next simulated a series of studies of size ranging from one listener per group (total N=5) to 6 listeners per group (total N=30, i.e., the original study). In this simulation, we sought to determine how many listeners were needed to retain significant pair-wise differences between synthesis systems. The results of this analysis are shown in Figure 7 where each small panel presents the average t-values for 50 comparisons between one pair of TTS systems with various numbers of subjects. Red dotted lines indicate the nominally significant level ($p < .05$) of t (without correction for multiple tests). Error bars indicate to total range of t-values observed. Significant differences between systems D and E were never observed. Significant differences between system A and systems D and E are sometimes lost with even a reduction from 30 to 25 subjects. Virtually all other pair-wise comparisons remained significant with only 5 or 10 listeners (i.e., one or two listeners per group).

4. Discussion

A variety of different analyses were presented to examine intelligibility measures at different levels of analysis. At the sentence level, the absolute overall ranking of the five TTS systems differed slightly from the other two levels. System E had a lower proportion of errors than System D in the sentence-level analysis, but a higher proportion in other analyses. However, the differences between these two systems were extremely small and not statistically significant in any analysis. Hence, both systems should really be considered to share a single rank in all analyses. With this qualification in mind, it seems safe to conclude that for merely ranking the intelligibility of TTS systems, it makes little difference whether one uses a simple "sentences correct" measure or a more labor-intensive phonetic edit distance measure.

It is often important, however, to be able characterize how much more intelligible one system is compared to others. For instance, in selecting a TTS system for a specific application, one may want to consider multiple factors including intelligibility, naturalness, preference for a specific voice gender, etc. In weighing these factors to arrive at a final decision, knowing the amount of intelligibility difference between two systems is essential. That is, one may be willing to accept a small, but not a large, loss of intelligibility in favor of a more natural or pleasing voice. Our results here suggest that sentence-level scoring may obscure the magnitude of the differences between systems. While the present analysis was sufficiently well powered to detect the overall differences between most of the TTS systems at all levels of analysis, there is concern that screening studies run with fewer listeners and/or a smaller number of utterances per synthesizer would fail to detect differences with sentence-level scoring that would be detectable with word- or phone-level scoring.

It is, of course, clear that developers of TTS systems need analyses at the phonetic level to diagnose specific strengths and weaknesses with systems. In that case, an encouraging finding from the present study is the absence of secondary effects of carefully hand-correcting phonetic data. Although our efforts to carefully interpret the phonetic intent of the subjects did result in overall lower error rates for all systems,

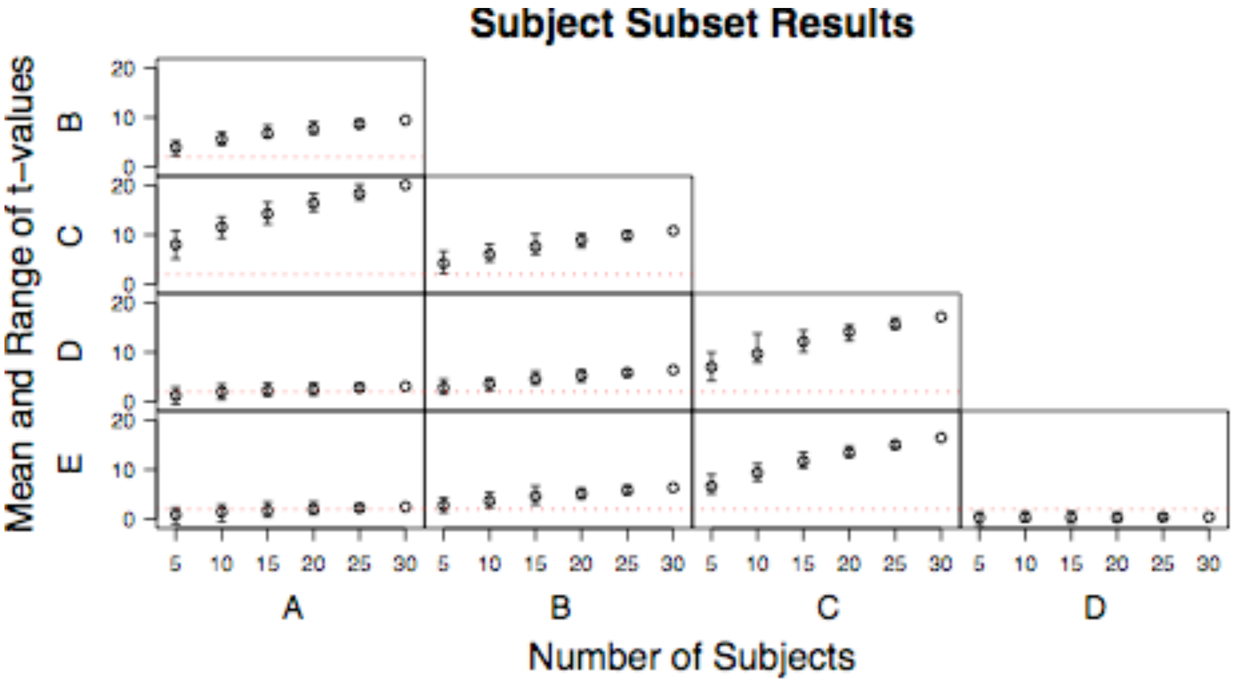


Figure 7: Average t-values for all pairwise comparisons among the five synthesizers as a function of the number of subjects. Error bars reflect total range of t-values obtained in 50 simulation trials. See text for explanation.

there is no evidence that these efforts would have consequences for conclusions drawn from the study. This in turn suggests that it should be possible to develop relatively automated scoring procedures based on dictionaries that include entries for highly probable typos, misspellings, and the like.

Another advantage to using phonetic edit distances is that the edit distance data can be further analyzed to diagnose specific phonetic strengths and weaknesses within a system, or to discover differences between systems with indistinguishable gross intelligibility scores. For example, while systems D and E in the present analyses have indistinguishable total edit distance scores, we found that responses to system E had a greater number of deletions, while responses to system D had greater numbers of insertions and substitutions. These differences in the types of errors listeners make on one system versus another may prove to be of diagnostic value. It is also a simple matter, during the computation of phonetic edit distances, to tabulate a confusion matrix of stimulus phones versus response phones, allowing systems to be examined and compared by phonetic classes. For example, responses to system E were more likely to delete voiced phones or replace them with voiceless ones, while responses to system D were more likely to subsitute labials with nonlabials.

Finally, simulations of smaller experiments with different numbers of listeners and stimuli highlighted the importance of using a large number of stimuli per synthesizer, relative to the number of listeners. It is interesting to speculate that this may be particularly true of studies using concatenative TTS systems because of the very large number of unique concatenation sequences such systems may employ.

5. Conclusions

If one is only interested in ranking the relative intelligibility of several TTS systems, sentence-level scoring of SUS response data may be adequate, but it may mask the magnitude of real differences between systems. For more detailed analyses, phonetic edit distance is a more attractive measure. While the amount of effort needed to obtain phonetic-level edit distances is greater than that needed for a words or sentences correct measure, we found that little is gained by investing large amounts of effort in screening and interpreting listener responses. Instead, a more automated (and probably more objective) approach yields slightly higher overall error rates, but does not otherwise appear to influence conclusions one might draw from the data.

6. Acknowledgements

Work supported by NIDCD grant # R42DC006193 and Nemours Biomedical Research.

7. References

1. Nye, P.W. and J.H. Gaitenby, The intelligibility of synthetic monosyllabic words in short, syntactically normal sentences. *Haskins Laboratory Status Reports on Speech Research,* 1974. **37/38**: p. 169-190.
2. Benoit, C., M. Grice, and V. Hazan, The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using Semantically Unpredictable Sentences. *Speech Communication*, 1996. **18**(4): p. 381-392.
3. Benoit, C., An Intelligibility Test Using Semantically Unpredictable Sentences - Towards the Quantification of Linguistic Complexity. *Speech Communication*, 1990. **9**(4): p. 293-304.
4. Bunnell, H.T., et al., Automatic personal synthetic voice construction. *Proceedings of InterSpeech 2005*, Lisbon, Portugal, 2005.

Understandable Production of Massive Synthesis

Brian Langner, Alan W Black

Language Technologies Institute
Carnegie Mellon University, Pittsburgh, USA
{blangner,awb}@cs.cmu.edu

Abstract

This paper explores *massive synthesis*, or synthesis of sufficiently large amounts of content such that its evaluation is challenging. We discuss various applications where massive synthesis may apply, and their related issues. We also outline factors related to those applications that affect the perceived quality and intelligibility of the speech output, and discuss modifications of those factors that can improve the understandability of the resulting synthetic speech. There is a discussion of the challenges of evaluating this work, and of the different possible metrics that may be appropriate. Finally, we show in a simple evaluation that our modifications improve the perceived quality of the synthesis.

1. Introduction

Speech synthesis is increasingly being used to deliver spoken information to people. As its use becomes more frequent, new applications which push the limits of viable synthesis become more desirable. One such application involves converting some large amount of text-based information into speech, for listening to in situation where reading is inappropriate or impossible, such as while driving or exercising – a sort of “automatic podcast generation” task. The requirements of this application are highly understandable speech that is of sufficiently high quality that people will listen to it.

The difficulty, of course, is that this task has an enormous amount of text to be synthesized, when the potential uses are examined, to the extent that it is impossible for a person, or even a group of people to realistically evaluate it before use. This is compounded by the likelihood that new content is being continuously generated, making optimizations based on prior evaluations potentially less useful.

We are calling this application task *massive synthesis* – synthesis of such a large amount of data that more typical evaluation methods are impractical because no single person will be able to listen to enough of it. The goal of our work is to identify potential problems and find solutions that maximize intelligibility and understandability with the least manual intervention possible.

Though similar, there are two different relevant concepts here. *Intelligibility*, or how well the words a synthesizer produces can be correctly recognized, is an important measure for determining the quality of the speech synthesis. *Understandability*, or how well knowledge, information, and concepts can be transferred from the speaker to the listener, is also of great importance when considering speech applications designed to provide information. For speech, understandability builds on the intelligibility, which provides a sort of ceiling for how understandable the speech will be; less intelligible speech, by default, will be less understandable as well. Both of these are important to the task of massive synthesis; though more challenging, the ultimate goal of this work is to produce understandable, and not just intelligible, synthetic spoken output.

2. Massive Synthesis

2.1. Potential Applications

We envision several possible applications that could be classified as massive synthesis. Tasks such as error reports, business case summaries, even a news reader, all have characteristics that synthesis for them would end up as massive synthesis. Producing speech in these domains, at least on a sufficiently large scale or sufficiently often, will result in too much audio to legitimately evaluate. However, the task requires high understandability in order to be a success - listening to a news story you can’t understand right away is not worthwhile, and most people would not bother.

All of these domains share the difficulty of having significant amounts of content, and generally, continuous generation of new content. However, each of the above tasks has some characteristics that may simplify them. For example, error reports are likely to have a standard format and fairly closed language, and while news stories typically have a few new or unknown words per day, they are otherwise fairly normal English text. Unfortunately, that can’t be said for all massive synthesis applications. Weblogs are another potential use of massive synthesis, and though they might be thought of as amateur-produced news articles, there can be some noticable differences, both in terms of the topics covered and the vocabulary used.

2.2. Example Content

2.2.1. Obtaining a Corpus

As mentioned above, there are several applications where there is more content to be synthesized than can reasonably be heard by any individual or small group. One of these, synthesis of weblogs, is interesting because of the large amount of continuously-generated content to synthesize, as well as a potentially large pool of users to listen to the synthesized output. Though each synthesized blog may have only a few listeners, the entire space here is quite large and is clearly suitable to the problem at hand.

It is fairly easy to collect data from a number of weblogs, though there are some concerns about making the content representative of a "generic" blog. Fortunately, there has already been an effort to create a large-scale corpus of weblog content. The TREC Blog06 corpus [1], a collection of over 100,000 RSS and Atom feeds collected over 11 weeks in late 2005 and early 2006, is an ideal example of a large corpus of this sort of text. The corpus was created by downloading the homepage and all new permalinks for each feed once a week, for a total of over 750,000 collected feeds over this time period. The corpus includes an appropriate amount of spam content for realism.

It should be noted that this corpus also has a non-negligible amount of non-English text, including French, Spanish, and German, among others. As we are only concerned with English at this time, this content was largely ignored.

2.2.2. Analysis

To examine the content of this corpus, we first did a small amount of text processing to extract the content from the surrounding HTML and meta-information from the corpus distribution. Removing this non-content information resulted in a 14 gigabyte collection of blog text. This is primarily the form in which we used the corpus.

We performed a word frequency analysis to determine how weblog text differed from other English text, such as news articles. Our hope was to find "blog frequent" words that would be unlikely to be synthesized well, either in terms of quality or intelligibility. Once the "unusual" frequent words were identified, we then would determine if they were present in the lexicon, and if not, if their predicted pronuciation is likely to be accurate. For words with implausible or incorrect pronunciation, they would be flagged and targetted for improvement strategies.

In general, our analysis found most of the text was typical for English, at least with the most frequent words, which is not surprising. The most frequent but atypical tokens, *html* and *blog* appeared 27th and 28th most frequently, respectively, but otherwise the top 50 words appear to be fairly normal for English text. Even the unusual words that are frequently seen tend to be normal English words, simply used more often than, say, in the Wall Street Journal. Other common words that are mishandled tend to be acronyms that should be spelled rather than pronounced (or vice-versa), such as "FAQ", or pluralized abbreviations such as "mp3s".

It is interesting to observe the frequency of "adult" content in this corpus. Though not overwhelmingly common, "porn" and variants appear several hundred thousand times in the data. This perhaps says something about what happens when content is produced anonymously, either through weblog posts or their comments.

3. Improving the Synthesized Content

For several reasons, speech output of this content is difficult to understand. Since the usefulness of a spoken report or article is very low if it can't be understood, this is a problem that must be solved. We believe there are several issues that cause the reduced understandability, but there also are likely solutions that can be implemented to mitigate the effects.

3.1. Relevant Factors

3.1.1. Non-standard Words

Though non-standard words [2] are present in many different applications, including news articles, it seems that weblog content has a higher incidence of these, and a wider variety. News articles are generally limited to numbers and some punctuation symbols, and perhaps some foreign names or words, whereas blogs can have a far greater range of non-standard tokens. These include technical jargon (particularly when the content is related to computing technology), what is termed *leetspeek* (or *l33t5p33k*), intentional or inadvertant typographical and spelling errors (such as "the-teh", "lose-loose" or "voila-viola"), *expressive spelling* (such as "soooo..."), self-censoring of expletives (as in "#*!%"), frequent usernames and handles that are often ambiguously pronounceable, as well as similar non-standard words as in news articles. To a certain extent this is due to the lack of a formal editor reviewing the content before publication, but the fact that weblogs tend to be treated more as informal conversation than a professional publication is also an influence on these trends.

Improperly rendering these non-standard words has a significant effect on the perceived quality and intelligiblity of the synthesized speech, reducing the overall understandability. For the listener to understand what they are hearing, the speech output must take into account these words, and produce something more like what a person would say when reading: "leet" rather than "el three three tee".

In many cases, these non-standard words can be grouped into classes, some of which may be quite large; for example, words containing numbers or punctuation substituted for letters. For these, it may be helpful to consider them as a foreign language of sorts, and approach learning their proper pronunciation in that fashion. Techniques as in [3] would prove useful in that situation, particularly if we can devise a system where users are capable of providing feedback while listening to the content.

3.1.2. Formatting / Text Structure

Because the bulk of the content we would be synthesizing in these applications is web-published material, there is an inherent structure embedded by use of markup languages. This structure likely will provide hints for appropriate ways to segment the content, even when presenting it as speech, rather than visually. Thus, a method that takes the text structure into account will likely be easier to understand.

Even if an individual post's content has no structure or formatting beyond simple paragraphs, the entire page containing the post almost certainly will: title, content sections, comment sections, archive links, links to other sites, ads, and other items. If the goal is to synthesize the content, removing or ignoring the parts of the structure that are unrelated or unnecessary should simplify the output and probably improve how it is perceived by the listener.

Similarly, how the text itself is formatted can be used as a guide for how it should be said. Words that are emphasized in the text should probably be emphasized when spoken. Expressive spelling, as mentioned above, is another example of text formatting signifying how it should sound when spoken. When this is done appropriately, it can make the resulting speech sound more like how a human would speak - and more understandable.

Other formatting issues can be more problematic than helpful. Improperly rendered HTML entities, for example, are likely to be very poorly understood when synthesized, and even if they can be understood, people will be unlikely to know (or care) what `&#8211;` (or as would be heard “ampersand hash eight two one one”) is supposed to represent.

3.1.3. Content Summarization

One issue that is likely to arise, particularly when synthesizing weblogs, is the problem of having very long articles, or related to that, several new articles, that should be spoken. Is it always appropriate to read very long articles in their entirety? Will condensing several new comments to the phrase “and there are 15 new comments” or similar be sufficient, or should all of those comments be heard? These questions and other similar ones do not seem to have obvious answers, but they are at the core of providing understandable speech to people.

Like most speech applications, the answers here likely depend in some way on either the user or the domain, or possibly both. Some users might prefer condensed summaries, while others insist on hearing everything. Summaries themselves can have several options. They can summarize the main article and just indicate there are comments, summarize both the article and any comments, just say how many new posts and comments there are, or something more abstract like “several pages of ravings from a barely literate teenager”, for example.

There are other, more intermediate options as well, such as *subsetting* the content. That is, speaking enough of the start to make it clear what the article is about, and then waiting for the user to indicate whether the system should continue or move on to something else. In this way, the user could more quickly “browse” through the content.

Though all of these can potentially help, the most appropriate option is almost certain to depend on user preferences.

3.1.4. Phrase Boundaries

It is fairly well known that improved phrase breaks can produce significant gains in the overall understandability of synthesized speech. This effect is likely amplified with informal writing, which is less likely to have consistent punctuation or other cues for identifying phrase breaks.

In some ways, weblog content – particularly very informally written content – can end up resembling “word soup” due to a lack of punctuation and grammatical sentences. The text, then, could be thought of in the same way as the output from machine translation engines, and synthesized appropriately. Because the language in the text is “unusual”, the default naïve method to determine phrase breaks will be less effective. Something more advanced, taking things such as part of speech into account, can probably provide improved breaks.

This problem is particularly noticable for non-sentence content, such as structural or navigational information on web pages. Sometimes the information provided is important, but simply reading it out without adding better prosody and phrasing makes it too difficult to understand.

3.1.5. Multiple Voices

Another possibility to improve intelligibility and understandability would be to use multiple voices, particularly with long utterances. Using different voices for different contexts - such as one for the main content, one or two others for other comments, and one for meta information or non-primary content - could provide audible cues to where content is changing. Those cues could, in turn, make the speech easier to follow, and thus, understand.

For situations where multiple different voices may not be appropriate or desired, a similar effect might also be obtained using a single voice but changing style, particularly combined with improved phrasing.

Also, though not strictly speaking a different voice, using non-speech sounds to render some text could also provide a more natural or understandable result. For example, turning “ROFL” into an appropriate laughter sound would probably be better than trying to turn that “word” into speech. Using non-speech sounds such as beeps to indicate shifts between different content can also provide a potential increase in understandability, though at the cost of decreased naturalness.

3.2. Identifying and Correcting Problems

Of course, in order to use the strategies outlined above, it is necessary to know when and where to apply them. The most likely method to find problems is to listen to the speech output, but as we have discussed above, massive synthesis is characterized by having too much content to listen to. However, evaluating *some* of it is likely to help, particularly if we select things which are more likely to have errors.

Determining whether the synthesis is correct is, in the end, always going to require someone to listen to the speech. This manual process is both slow and expensive, but necessary. To reduce the cost, we want to find as many potential problems *without* requiring a human listener as possible. There are some heuristics we can use here. First, though we want to select examples at random, we can start by selecting those examples with words not in the lexicon, such as those we flagged from the Blog06 corpus. Using this as a guide to select candidate examples for evaluations makes it for more likely to find errors.

Still, however, the amount of content to examine is likely to be large. Therefore, some method of gauging the severity of the potential errors would be ideal, in order to prioritize error correction. This is key, because trying to find and fix all errors is unlikely to be cost effective. The more optimal approach to error correction and resolution would be to concentrate on solutions that fix large classes of errors, and simple fixes that can be implemented quickly without much effort.

3.3. Evaluation

Like the problem for speech synthesis in general, it is difficult to describe a consistent, objective measure that can evaluate this speech with regard to its quality and/or understandability Typical approaches have included mean opinion scores, modified rhyme tests, semantically unpredictable sentences, and others, and in fact these have all been present in some fashion in the Blizzard Challenge [4] in previous years. However, though these approaches are suitable for comparing different synthesizers or methods, they are not as helpful for demonstrating improvement for a specific task, particularly with regard to understandability. Semantically unpredictable sentences are inherently an artificial task which may or may not have any bearing on understandability for a specific application.

There are other possibilities, however. Asking listeners to rate which of two or more examples they prefer, or "like more", could be a useful dimension presuming the voice being used is the same and the quality level is consistent across different utterances. However, such an open-ended criterion may not be capturing the desired information about quality and understandability, though a large evaluation with many examples and explicit directions should be able to demonstrate improvements over a baseline. Another option would be to design a test similar to reading comprehension tests for children; by providing the content, and then specific questions about what was present, it should be possible to identify differences in understandability. The drawback to this sort of approach is the effort and cost required to design and implement it; it is likely to be far more expensive than typical synthesis evaluations.

4. Simple Evaluation

4.1. Test Examples

Given all of the issues related to how the synthesis is perceived, as well as the cost-benefit analysis to dealing with them, we implemented a number of modifications to weblog-style text. These modifications include a set of "number-to-letter" rules that effectively translate common "leet" words into pronounceable English, rules for words such as "iTunes" that use case to identify syllable boundaries, and lexical entries for several common non-standard words like "pwn" and "kthx", among others.

To test our modifications, we synthesized random comments and articles from several blogs and content sources: Slashdot [5], MetaFilter [6], LiveJournal [7], as well as a random Wikipedia article [8] and text from the Blog06 corpus. We felt these were fairly representative of the types of content that we have been working with. All examples were selected randomly, with the only constraints on the content being non-pornographic, and total playing time under 40 seconds.

Each of the examples was synthesized with a default Festival [9] installation and using our modifications. We used one of the Nitech HTS Arctic voices [10], because we felt, based on the results of past evaluations, the HTS voice would provide consistent, good-quality synthesis and reduce perceived quality differences between multiple utterances. The original content was identical between the modified and unmodified versions, though obviously the modified output might contain different phrases due to the token modifications.

4.2. Task Setup

Subjects were asked to listen to 6 different content examples, one from each method, for a total of 12 different wavefiles. For each example, they were instructed to identify which of the two waveforms they felt was better, and then rate on a scale of 1 to 5 how much better. The order of presentation was randomized, such that the same method was not used to generate the first presented wavefile for all examples.

Five subjects, all of whom are familiar with speech synthesis, took part in this evaluation. Each was given a URL that outlined the task to them, and provided the wavefiles to listen to. Subjects could listen to the examples using either speakers or headphones, but were encouraged in either case to listen to each file as few times as possible.

4.3. Results

All subjects universally preferred the modified examples to the unmodified ones. Though we expected a clear preference to emerge, it is still somewhat surprising that this preference was complete in all cases.

There was less consistent cross-listener agreement in the degree of preference, however, with some examples showing strong agreement and others almost none. In general, the average preference was fairly weak, so despite a clear preference for the modified utterances, that preference does not seem to indicate a strong improvement over the baseline. The preference scores are shown in Table 1. These results are not statistically significant due to the limited sample size.

	Min Pref	Avg Pref	Max Pref
Ex 1	1	2.2	3
Ex 2	1	3	4
Ex 3	1	1.8	2
Ex 4	1	2	3
Ex 5	2	3	5
Ex 6	2	3	4

Table 1: Degree-of-preference scores from this evaluation.

5. Discussion

As the results from our evaluation show, it is clear that some fairly simple modifications will result in speech which is perceived as better to at least some degree. More thorough or complex changes might produce an even more obvious user preference. Our results, unfortunately, are lacking more detailed comments that would prove useful in how the speech was perceived. It may be that listeners found both examples to be poor or difficult, and one was simply "less bad" than the other.

It seems likely, based on some past evaluations and anecdotal experiences, that improved prosody will be required to have truly understandable synthesis of lengthy items. The machine-like qualities simply make it harder to concentrate on the speech, with the result being longer utterances are far more difficult to understand. On some level this is likely to be a memory issue – people have limited auditory memory [11], and even natural speech is hard to remember after hearing a long talk. However, the fact that people can routinely go to an hour-long lecture and come away having learned something suggests memory is not a valid excuse to hide behind. It seems highly unlikely that the same lecture, if delivered by a speech synthesizer, would be as well understood, or received by the audience, even with a modern, state of the art synthesizer. We would like to, with this work, be able to "close the gap" and reduce the understandability differences between synthetic and natural speech.

One area which we discussed but have not explored here is utilizing the structure of the content to help influence its synthesis and presentation. Other recent work [12] suggests this can be helpful, both in terms of resulting understandability, but also with summarizing complex information into something more suitable for spoken output. We feel that looking further into this has high potential for improving massive synthesis.

As we discussed above in regarding evaluation, a massive synthesis application will never be able to be quality-checked in the same fashion as, say, a limited domain synthesizer. To help alleviate that issue, we believe having users of these applications provide feedback (and if possible, corrections) can provide useful improvements to the spoken output. The drawbacks to this approach are that for truly useful feedback, the users must actually care about what they are listening to, and have a want or need to understand it. This becomes tricky since those types of users, besides being harder to find in the first place, are also the ones who are least likely to put up with doing error correction as part of using a system like this. However, it is important to be have this sort of feedback mechanism to drive improvements.

Even with user-provided feedback, however, it is unclear that there is a good evaluation metric on which to judge progress. On some level, receiving fewer error reports from users would be a reasonable measure (presuming that the user base stays constant). Other metrics such as token error rate may be useful as well, but there is still likely a perceptual component that needs to be considered.

Moving forward, we envision developing a prototype system which, given the URL or other location of a document, will parse the content and provide a "podcast" to listen to – in some sense, a web browser that instead of displaying the content on a screen, renders it as speech. Given the nature of this, some collaboration with groups working on web browsers for the blind might be beneficial.

6. References

[1] C. MacDonald and I. Ounis, "The TREC Blog06 Collection: Creating and analysing a blog test collection," Department of Computing Science, University of Glasgow, Tech. Rep. TR-2006-224, 2006.

[2] R. Sproat, A. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards, "Normalization of non-standard words," *Computer Speech and Language*, vol. 15, no. 3, pp. 287–333, 2001.

[3] J. Kominek and A. Black, "Learning pronunciation dictionaries: Language complexity and word selection strategies," in *HLT-NAACL*, New York City, USA, 2006.

[4] C. Bennett, "Large scale evaluation of corpus-based synthesizers: Results and lessons from the Blizzard Challenge 2005," in *Interspeech 2005*, Lisbon, Portugal, 2005.

[5] user RedBear, "slashdot.org article comments," [accessed 12-May-2007], http://slashdot.org/.

[6] user gsb, "MetaFilter article comments," [accessed 12-May-2007], http://www.metafilter.com/.

[7] user cdinwood, "LiveJournal article," [accessed 12-May-2007], http://www.livejournal.com/.

[8] Wikipedia, "Train surfing," [accessed 12-May-2007], http://en.wikipedia.og/w/index.php?title=Train_surfing&oldid=127993634.

[9] A. Black, P. Taylor, and R. Caley, "The Festival speech synthesis system," 1998, http://festvox.org/festival.

[10] H. Zen and T. Toda, "An overview of Nitech HMM-based speech synthesis system for blizzard challenge 2005," in *Interspeech 2005*, Lisbon, Portugal, 2005.

[11] A. D. Baddeley, N. Thomson, and M. Buchanan, "Word length and the structure of short-term memory," *Journal of Verbal Learning and Verbal Behavior*, vol. 14, pp. 575–589, 1975.

[12] B. Langner and A. Black, "uGloss: A framework for improving spoken language generation understandability," 2007, submitted to Interspeech 2007, Antwerp, Belgium.

The Online Evaluation of Speech Synthesis Using Eye Movements

Charlotte van Hooijdonk, Edwin Commandeur, Reinier Cozijn, Emiel Krahmer & Erwin Marsi

Department of Communication & Information Sciences
Tilburg University, The Netherlands
{C.M.J.vanhooijdonk; E.Commandeur; R.Cozijn; E.J.Krahmer; E.C.Marsi}@uvt.nl

Abstract

This paper* describes an eye tracking experiment to study the processing of diphone synthesis, unit selection synthesis, and human speech taking segmental and suprasegmental speech quality into account. The results showed that both factors influenced the processing of human and synthetic speech, and confirmed that eye tracking is a promising albeit time consuming research method to evaluate synthetic speech.

1. Introduction

The evaluation of synthetic speech in terms of intelligibility has primarily been done with offline research methods. For example, the Modified Rhyme Test [1] has been used to investigate the segmental intelligibility of synthetic speech [2]. In this test, listeners are presented with spoken words and are instructed to select the word they heard from a set of alternatives that differ only in one phoneme. Another example is the Mean Opinion Score [3] in which listeners have to rate the quality of spoken sentences on scales (i.e., excellent - bad).

A disadvantage of offline research methods is that no insight is obtained in how listeners process synthetic speech. Online research methods, like eye tracking, give a direct insight in how speech is processed incrementally. In the "visual world paradigm", participants are asked to follow spoken instructions to look up or pick up objects within a visual display (e.g., [4, 5]). The fixation patterns on the objects within the display are used to draw inferences about the processing of spoken instructions. Eye tracking might give an idea of how similar the processing of synthetic speech is, compared to the processing of human speech. This idea was first explored by Swift et al. [6] in a study concentrating on acoustically confusable words (e.g., beetle, beaker, and speaker) to see if the "disambiguation" point was processed at comparable time windows for two instances of synthetic speech and human speech. The results showed that both human speech instructions and synthetic speech instructions were indeed processed incrementally. Moreover, when hearing the onset of the target noun (e.g., beaker), the listeners were more likely to look at the cohort competitor (e.g., beetle) than at the rhyme competitor (e.g., speaker). Finally, the listeners identified the target more rapidly in the human speech condition than in the two synthetic speech conditions.

The intelligibility of speech does not only depend on its segmental quality but also on the quality and the appropriateness of the prosodic information in the speech signal (i.e., suprasegmental quality) [7]. The visual word paradigm has more recently been used to investigate how humans process prosodic information. For example, Weber et al. [8] used eye tracking to investigate how prosodic information influences the processing of spoken referential expressions. In two experiments, participants followed two consecutive instructions to click on an object within a visual display. The first instruction mentioned the referent (e.g., purple scissors). The second instruction either mentioned a target of the same type but with a different colour (red scissors) or of a different type and a different colour (red vase). The instructions were either realised with an accent on the adjective (e.g., Click on the PURPLE scissors, Click now on the RED scissors) or on the noun (e.g., Click on the purple SCISSORS, Click now on the red SCISSORS). The results showed that the listeners were affected by this prosodic difference. When the first instruction was realized with an accent on the adjective (e.g., Click on the PURPLE scissors), listeners anticipated the upcoming target, i.e., before the onset of the target noun, listeners looked at the target of the same type as the referent but with a different colour (red scissors). When both instructions were realized with an accent on the adjective (e.g., Click on the PURPLE scissors, Click now on the RED scissors) this anticipation only increased. However, when the instructions were realized with an accent on the noun (e.g., Click on the purple SCISSORS, Click now on the red SCISSORS), listeners did not anticipate the upcoming target.

Both segmental and suprasegmental quality are important factors in processing synthetic speech. In this paper, we therefore extend on the work by Swift et al. by focusing on both segmental and suprasegmental aspects of speech. In our evaluation experiment, the participants were given two consecutive spoken instructions to look at a certain object within the visual display. These instructions were presented in three speech conditions: diphone synthesis, unit selection synthesis, and human speech. Diphone synthesis is based on concatenating prerecorded diphones (i.e., phoneme transitions), followed by signal processing to obtain the required pitch and duration. Unit synthesis is also based on concatenation, but on a much larger scale, where units are of variable size (e.g., sentences, constituents, words, morphemes, syllables, and diphones). As larger units of natural speech are exploited, requiring less concatenation, the segmental quality of unit synthesis is in general significantly higher than that of diphone synthesis. At the same time, the prosody may be inadequate, because the intended realisation of, for example, pitch accents, may not be available in the speech database. Thus, while quality of diphone synthesis is in general inferior to that of unit synthesis, it has the

*The current research is performed within the IMIX-IMOGEN (Interactive Multimodal Output GENeration) project sponsored by the Netherlands Organisation for Scientific Research (NWO). The authors would like to thank Lennard van der Laar, Pascal Marcelis, and Marie Nilsenova for their help in setting up the experiment, Marc Swerts for discussing the findings of the experiment and Carel van Wijk for his statistical advice.

advantage that it can always produce contextually appropriate prosody (albeit by human intervention). In this experiment, we investigated this trade-off between segmental quality on the one hand and contextually appropriate prosody (i.e., suprasegmental quality) on the other from the perspective of humans processing synthetic speech. The human speech condition was added as a baseline to compare processing of natural and synthetic speech.

2. Method

2.1. Participants

Thirty-eight native speakers of Dutch (13 male and 25 female, between 18 and 33 years old) were paid to participate. They had normal or corrected-to-normal vision and normal hearing. None of the participants were colour-blind and none had any involvement in speech synthesis research.

2.2. Stimuli

Fifteen pairs of Dutch monosyllabic picturable nouns were chosen as stimuli. These nouns shared the same initial phonemes (e.g., *vork - vos*, fork - fox). Each experimental trial consisted of a 3x3 grid with four objects in the corner cells, see Figure 1[1].

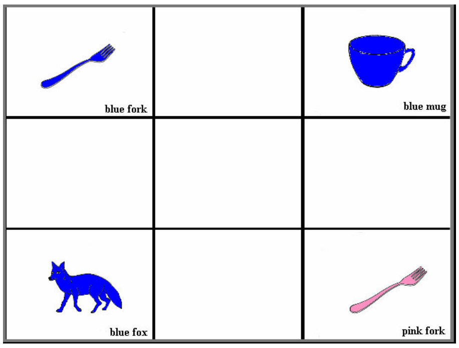


Figure 1: An example of a visual display

For every grid, the participants were given two consecutive spoken instructions each referring to a certain object within the grid. In both instructions, the nouns were modified with a colour adjective (blue or pink). The first instruction mentioned the **referent** (e.g., *Kijk naar de roze vork*, Look at the pink fork). The second instruction mentioned the **target**. The target could either be of the **same type** as the referent modified with a different colour adjective (e.g., *Kijk nu naar de blauwe vork*, Now look at the blue fork), or of a **different type** as the referent modified with a different colour adjective (e.g., *Kijk nu naar de blauwe vos*, Now look at the blue fox). A fourth object was added as a **distractor** (e.g., *blauwe mok*, blue mug). The distractor did not share the form of the other objects, but did share the colour with the two targets. The distractor was never mentioned in the experimental trials. The colours blue and pink could occur in both instructions and were randomized across the trials.

The first instruction was realised with a standard, neutral intonation contour. In the second instruction, the adjective and noun were both accented (e.g., *BLAUWE VOS*, BLUE FOX). In half of the cases the second instruction had a contextually appropriate double accent pattern while the other half had not, see Table 1. The second instruction had an appropriate accent pattern when it mentioned a different colour adjective and a different object type as the referent in the first instruction. The second instruction had an inappropriate accent pattern when it mentioned a different colour adjective but had the same object type as the referent in the first instruction [9, 10]. Note that the choice of a double accent pattern was forced by the output of the unit selection synthesizer, as it typically produces these double accents.

Table 1: Example of the instructions

First instruction	*Kijk naar de roze vork* Look at the pink fork
Second instruction contextually appropriate double accent pattern	*Kijk nu naar de BLAUWE VOS* Now look at the BLUE FOX
Second instruction contextually inappropriate double accent pattern	*Kijk nu naar de BLAUWE VORK* Now look at the BLUE FORK

The instructions were realised in three speech conditions, i.e., diphone synthesis, unit selection synthesis, and human speech. A female voice was used for all three speech conditions. The diphone stimuli were produced using the Nextens[2] TTS system for Dutch, which is based on the Festival TTS system [11]. The input consisted of words and prosodic markup. Pitch accents were phonetically realised with a rule-based implementation of the Gussenhoven & Rietveld model for Dutch intonation [12]. For the unit selection synthesis a commercially available synthesizer was used. The instructions were obtained through an interactive web interface of the synthesizer. The output that was given by the interface was stored. Note that it was not possible to control the accent patterns of the instructions, as this type of synthesis is dependent on the intonation of the selected units in the database of the synthesizer. The instructions in the human speech condition were recorded by a native speaker of Dutch (the first author) in a quiet room at Tilburg University. The instructions were digitally recorded, sampling at 44 kHz, using Sony Sound Forge™ and a Sennheiser™ microphone (type SKM 135 G2). The instructions were recorded multiple times and the best realisations were chosen. An independent intonation expert checked the utterances using PRAAT [13] to make sure that the intended accents in the second instructions were properly realised. All instructions in the three speech conditions were normalized at -16 dB, using Sony Sound Forge™, and stored in stereo format.

We checked whether there were durational differences between the target nouns mentioned in the second instruction between the various conditions. It turned out that speech condition did not affect the duration (F< 1). Also, the target object type (same object type vs. different object type) mentioned in the second instruction did not affect its duration

[1] The textual descriptions in figure 1 are only added for illustrative purposes, they did not occur in the actual experiment.

[2] http://nextens.uvt.nl

(F< 1). Finally, there was no interaction for duration between speech condition and target object type (F< 1).

In addition to the 90 experimental trials (15 stimuli × 3 speech conditions × 2 target object types), 20 filler trials were constructed to add variety to the visual display, and the accent pattern of the second instruction. In the filler trials, either the adjective or the noun mentioned in the second instruction was accented (i.e., *ROZE mok*, PINK mug or *roze MOK*, pink MUG), and they were only realised in human speech and diphone synthesis.

Three lists were constructed in a semi-Latin square design, each containing 90 experimental and 20 filler trials. In each list, the stimuli were mixed up and presented as one block to the participants. Thus, the participants were presented with all three speech conditions and both target object types during the experiment.

2.3. Procedure

Each participant was invited to an experimental laboratory, and was seated in front of a computer monitor. First, the participants were familiarised with the objects that occurred within the visual display during the experiment to ensure that they identified them as intended. This was done by asking them to describe the thirty depicted objects and their colour (pink or blue) aloud. The objects were shown in the middle of the computer screen. Participants could view each object at their own pace by clicking on a button, and they were corrected when an object was described incorrectly. This object was viewed again until it was described correctly.

Subsequently, the instructions of the actual experiment were read to the participants, and the eye-tracking system was mounted and calibrated. Participants' eye movements were monitored using an SR Research EyeLink II eye-tracking system, sampling at 250 Hz. Only the right eye of the participant was tracked. The spoken instructions were presented to the participants binaurally through headphones. Next, the participants were presented with a practice session in which the procedure of the experiment was illustrated. This practice session consisted of six trials (3 speech conditions × 2 target object types). The structure of a trial was as follows. First, participants saw a white screen with in the middle a little black cross, and they pressed a button to continue. Next, a white screen appeared with in the middle a central fixation point, and the participants were instructed to look at this point. The experimenter then initiated an automatic drift correction to correct for shifts of the head-mounted tracking system. After the automatic drift correction, the visual display appeared. The first instruction was given after 50 milliseconds. The participants had to look at the object that was mentioned, after which they pushed a button. Subsequently, a little black cross appeared in the centre of the grid and the participants were instructed to look at this cross. After 2000 milliseconds, the cross disappeared and the second instruction was given. Again, the participants had to look at the object that was mentioned, after which they pushed a button. Subsequently, the white screen with in the middle a little black cross appeared again and the participants pressed on a button indicating the start of the next trial. After completing the practice session, the actual experiment started, proceeding in the same way as the practice session. During the experiment, there was no interaction between the participant and the experiment leader.

After the participants completed the experiment, they were asked to listen to an instruction (i.e., *Kijk nu naar de BLAUWE BLOEM,* Now look at the BLUE FLOWER) realised in diphone synthesis, unit selection synthesis, and human speech. Next, they were asked to fill out a questionnaire. This questionnaire consisted of four items about the intelligibility (i.e., audibility, comprehensibility, perceptibility, and distinctness) and four items about the naturalness (i.e., intonation, pleasantness to listen, speech rate, and naturalness) of the three speech conditions. Each question was accompanied with a 7-point Likert scale on which the participants could indicate how much they agreed or disagreed with the content of each item.

2.4. Coding procedure and data processing

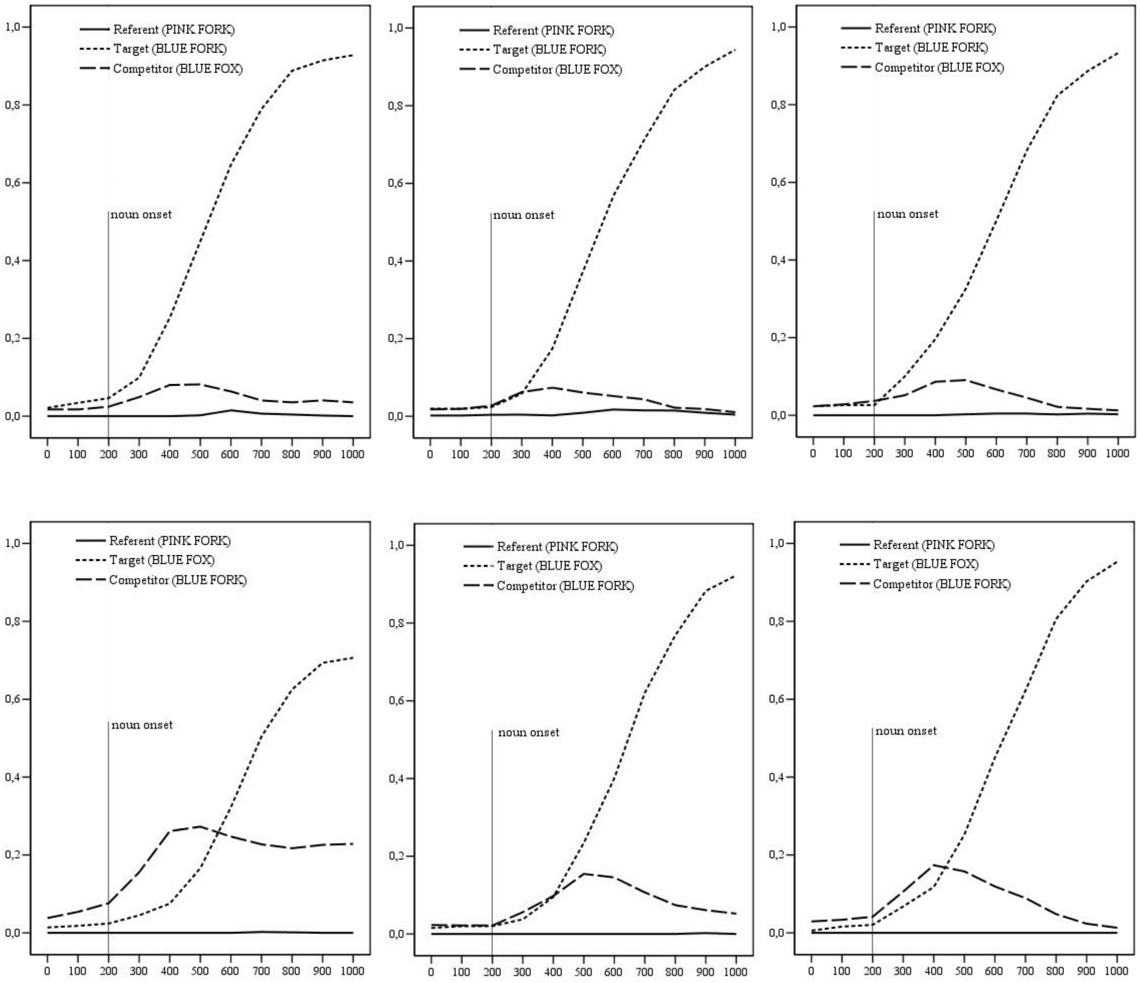
Eyelink software parsed the eye-movement data into fixations, saccades, and blinks. Fixations were automatically mapped (using the program Fixation[3]) on the objects presented in each trial, and this mapping was checked by hand. The fixations occurring in the first and second instruction of a trial were analysed. In the first instruction, trials in which less than 50% of the sample points after the onset of the referent noun belonged to fixations on the referent object were excluded from further analysis. In the second instruction, trials in which less than 50% of the sample points before the onset of the target noun belonged to fixations on the centre of the grid were excluded from further analysis. These trials were excluded because the instructions were not followed. The data of one participant was excluded, as she did not meet the above-mentioned criteria in any of the trials. The total amount of data that was excluded from further analysis was 7.7%, including the data discarded for the above-mentioned participant.

Fixation proportions were averaged over two time windows for each participant F_1 and item F_2 and analysed with a 3 (diphone synthesis, unit selection synthesis, human speech) × 2 (same target object type, different target object type) repeated measures analysis of variance (ANOVA)[4], with a significance threshold of .05. For post hoc tests, the Bonferroni method was used. The dependent variables were the mean proportions of fixations to the target and to the competitor. The first time window began 200 ms after the onset of the target noun, because this is the earliest point at which fixations driven by information from the target noun were expected [5, 14]. The time window extended over 400 ms, which roughly corresponded to the mean duration of the target noun. The second time window extended from 600 to 1000 ms after the target noun onset.

The results of the questionnaire were processed by mapping the items to which the participants disagreed to 1 and agreed to 7 and were analysed with a 3 (speech condition) × 4 (items) repeated measures analysis of variance (ANOVA), with a significance threshold of .05. For post hoc tests, the Bonferroni method was used.

[3] http://www.tilburguniversity.nl/faculties/humanities/people/cozijn/research

[4] Mauchly's test of sphericity was significant for some main effects and interactions. For these cases, we looked both at Greenhouse-Geisser and Huynh-Feldt corrections on the degrees of freedom, which gave similar results. For the sake of transparency, we report on the normal degrees of freedom.



DIHONE SYNTHESIS UNIT SELECTION SYNTHESIS HUMAN SPEECH

Figure 2: Proportions of fixations to the referent, the target, and the competitor for diphone synthesis, unit selection synthesis, and human speech for the second instruction mentioning a same target object type (top row) and different target object type (bottom row).

Table 2: Mean proportions of fixations to the target and competitor for the two time windows in relation to the speech condition

	Diphone synthesis		Unit selection synthesis		Human speech	
	Target	Competitor	Target	Competitor	Target	Competitor
Time window 200 – 600 ms	.22	.15	.20	.09	.21	.11
Time window 600 – 1000 ms	.72	.14	.78	.06	.78	.04

Table 3: Mean proportions of fixations to the target and competitor for the two time windows in relation to the target object types mentioned in the second instruction

	Target object of the same type		Target object of a different type	
	Target	Competitor	Target	Competitor
Time window 200 – 600 ms	.26	.07	.16	.16
Time window 600 – 1000 ms	.82	.03	.70	.12

3. Results

3.1. Eye movement data

Table 2 summarizes the mean proportions of fixations found within the time window 200 to 600 ms for the three speech conditions. The statistics showed that the mean proportions of fixations to the target did not differ significantly between the three speech conditions F_1 and $F_2 < 1$. In all three speech conditions, the mean proportions of fixations to the target were approximately 20%. However, there was a significant difference between the three speech conditions in the mean proportions of fixations to the competitor: F_1 [2,72] = 20.68, p< .001, partial eta squared = .37; F_2 [2,28] = 10.13, p< .001, partial eta squared = .42. The mean proportions of fixations to the competitor were the highest in the diphone synthesis condition and the lowest in the unit selection synthesis condition. The mean proportions of fixations to the competitor in the human speech condition fell between these two. Table 3 reveals that within the time window 200 to 600 ms, the mean proportions of fixations to the target were higher when the second instruction mentioned a target object of the same type than when it mentioned a target object of a different type: F_1 [1,36] = 48.82, p< .001, partial eta squared = .58; F_2 [1,14] = 34.08, p< .001, partial eta squared = .71. Conversely, the mean proportions of fixations to the competitor were higher when the second instruction mentioned a target object of a different type than when it mentioned a target object of the same type: F_1 [1,36] = 44.40, p< .001, partial eta squared = .55; F_2 [1,14] = 21.67, p< .001, partial eta squared = .61. Finally, within the time window 200 to 600 ms, an interaction was found between speech condition and target object type mentioned in the second instruction for both the mean proportions of fixations to the target: F_1 [2,72] = 18.93, p< .001, partial eta squared = .35; F_2 [2,28] = 11.18, p< .001, partial eta squared = .44, and to the competitor: F_1 [2,72] = 21.95, p< .001, partial eta squared = .38; F_2 [2,28] = 9.73, p< .005, partial eta squared = .41. For all three speech conditions, the mean proportions of fixations to the target were significantly higher when the second instruction mentioned a target object of the same type than when it mentioned a target object of a different type. Conversely, for all three speech conditions the mean proportions of fixations to the competitor were significantly higher when the second instruction mentioned a target object of a different type than when it mentioned a target object of the same type.

In the time window 600 to 1000 ms, an significant effect was found of speech condition in the mean proportions of fixations to the target, although not by items: F_1 [2,27] = 12.70, p< .001, partial eta squared = .26; F_2 [2,28] = 1.32, p = .28. The mean proportions of fixations to the target were the highest for unit selection synthesis and human speech and low for diphone synthesis. The results found for speech condition in the mean proportions of fixations to the competitor were similar to those found in the time window 200 to 600 ms, F_1 [2,72] = 57.16, p< .001, partial eta squared = .61; F_2 [2,28] = 5.28, p< .025, partial eta squared = .27. Also, similar results were found for the target object types mentioned in the mean proportions of fixations to the target: F_1 [1,36] = 72.92, p< .001, partial eta squared = .67; F_2 [1,14] = 19.93 p< .005, partial eta squared = .59, and to the competitor: F_1 [1,36] = 83.13, p< .001, partial eta squared = .7; F_2 [1,14] = 10.87, p< .01, partial eta squared = .44. Finally, a similar interaction was found between speech condition and target object type in the mean proportions of fixations to the competitor: F_1 [2,72] = 53.45, p< .001; partial eta squared = .60; F_2 [2,28] = 3.70, p< .05, partial eta squared = .21. The interaction found between speech condition and target object type in the mean proportions of fixations to the target was different from the results found in the previous time window, F_1 [2,72] = 57.20, p< .001; partial eta squared = .61; F_2 [2,28] = 5.96 p< .01, partial eta squared = .30. Only in the diphone synthesis condition: F_1 [1,36] = 129.89, p< .001; partial eta squared = .78; F_2 [1,14] = 13.18, p< .005, partial eta squared = .49, and in the unit selection synthesis condition: F_1 [1,36] = 17.12, p< .001; partial eta squared = .32; F_2 [1,14] = 7.26, p< .025; partial eta squared = .34, the mean proportions of fixations were higher when the second instruction mentioned a target object of the same type than when it mentioned a target object of a different type. For human speech, no difference between the speech conditions was found: F_1 [1,36] = 1.52, p = .27; $F_2 <$ 1.

3.2. Intelligibility and naturalness of the three speech conditions

Figure 3 illustrates the results found for the questionnaire on the intelligibility and the naturalness of the three speech conditions.

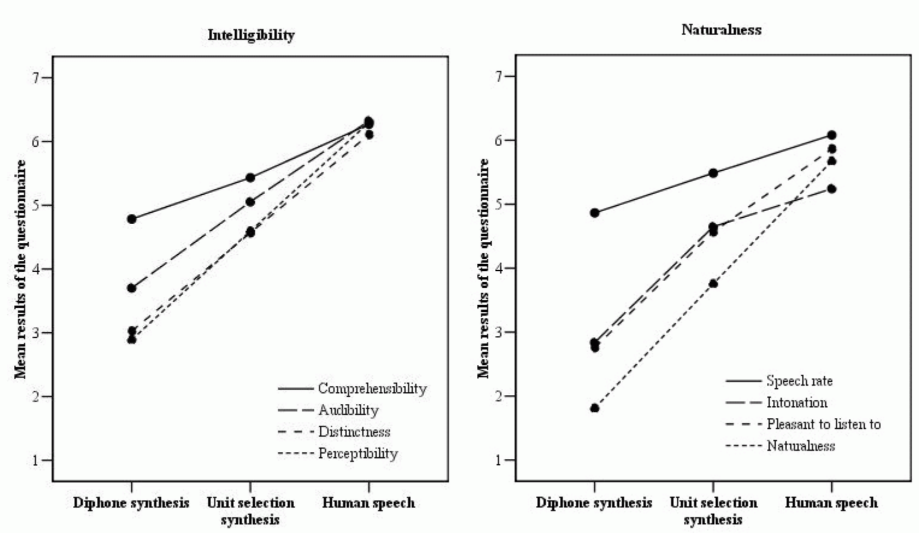


Figure 3: Mean results of the questionnaire on the intelligibility and naturalness of the three speech conditions

A significant effect was found of speech condition for both intelligibility: *F* [2,72] = 42.52, p< .001, η_p^2 = .54 and naturalness: *F* [2,72] = 49.83, p< .001, η_p^2 = .58. Post-hoc tests showed that all pairwise comparisons were significant at p< .001. For intelligibility and naturalness diphone synthesis was rated lowest followed by unit selection synthesis. Human speech was rated highest. Finally, Figure 3 shows that the participants were homogeneous in their ratings on the intelligibility and the naturalness of human speech, but they were heterogeneous synthesis.

4. Conclusion and discussion

In this paper, we described an experiment in which eye tracking was used to evaluate human speech, diphone synthesis, and unit selection synthesis having either contextually appropriate or inappropriate accent patterns. We found differences in the performance accuracy between the three speech conditions. In the time window 600 to 1000 ms, the mean proportions of fixations to the target were lowest for diphone synthesis and highest for unit selection synthesis and human speech. Also, in both time windows, significant differences between the three speech conditions were found in the mean proportions of fixations to the competitor. The mean proportions of fixations to the competitor were highest for diphone synthesis. An explanation for these results could be that the relatively poor segmental intelligibility of the diphone synthesis makes it harder for the participants to determine the disambiguation point of the acoustically confusable words. We also found that the participants anticipated the upcoming target. In both time windows, the mean proportions of fixations to the target were higher when the second instruction mentioned a target object of the same type. Moreover, interactions were found between speech condition and the target object type mentioned in the second instruction. In the time window 200 to 600 ms, the mean proportions of fixations to the target were significantly higher for all three speech conditions when the second instruction mentioned the same object type. However, in the time window 600 to 1000 ms, this interaction was only found for diphone synthesis and unit selection synthesis. These results indicate that not only the segmental intelligibility of synthetic speech plays an important role in speech processing, but also listeners' anticipations based on the accent patterns within the speech.

The results of the questionnaire showed that for both intelligibility and naturalness of the three speech conditions, diphone synthesis was rated lowest followed by unit selection synthesis. These subjective measures correspond with the results found in our eye-movement data. The combination of these offline subjective measures and online objective measures give a detailed insight in the perception and the processing of synthetic speech.

The experiment shows that eye tracking is a promising research method to evaluate synthetic speech. The results give an insight in how similar the processing of synthetic speech is compared to the processing of human speech on a segmental and a suprasegmental level. The complexity of the method could be reduced if a test bed environment would be created that enables an easy comparison of the processing of new speech synthesis systems. That way new speech synthesis methods could be tested in a standardised way.

5. References

[1] House, A.S., Williams, C.E., Hecker, M.H. and Kryter, K.D., "Articulation-testing methods: consonantal differentiation with a closed-response set." *JASA,* 37, 1965, pp 158-166.

[2] Pisoni, D.B., *Some measures of intelligibility and comprehension*. In J. Allen, M.S. Hunnicutt, and D.H. Klatt (eds.), 1987. From Text to Speech: the MITalk System. Cambridge University Press, Cambridge, pp.151-171.

[3] Schmidt-Nielsen, A., *Intelligibility and acceptability testing for speech technology*. In A. Syrdal, R. Bennett, and S. Greenspan (eds.), 1995. Applied Speech Technology. CRC: Boca Raton, pp. 194-231.

[4] Tanenhaus, M.K., Spivey-Knowlton, M.J., Eberhard, K.M., and Sedivy, J.E., "Integration of visual and linguistic information in spoken language comprehension". *Science*, 268, 1995, pp. 1632-1634.

[5] Altmann, G.T., and Kamide, Y., *Now you see it, now you don't: Mediating the mapping between language and the visual world.* In J. Henderson and F. Ferreira (eds.), 2004. The interface of language, vision, and action: Eye movements and the visual world. Psychology Press, New York, pp. 347-386.

[6] Swift, M.D., Campana, E., Allen, J.F., and Tanenhaus, M.K., "Monitoring eye movements as an evaluation of synthesized speech", *Proceedings of the IEEE 2002 Workshop on Speech Synthesis*, Santa Monica, CA.

[7] Sanderman, A.A., and Collier, R., "Prosodic phrasing and comprehension". *Language and Speech*, 40, 1997, pp 391-409.

[8] Weber, A., Braun, B., and Crocker, M. W., "Finding referents in time: eye-tracking evidence for the role of contrastive accents". *Language and Speech*, 49, 2006, pp. 367-392.

[9] Nooteboom, S.G., and Kruyt, J.G., "Accent, focus distribution, and perceived distribution of given and new information: An experiment". *JASA*, 82, 1987, pp. 1512–1524.

[10] Terken, J., and Nooteboom, S.G., "Opposite effects of accentuation and deaccentuation on verification latencies for Given and New information", *Language and Cognitive Processes*, 2, 1987, pp. 145-163.

[11] Black, A.W., Taylor, P., and Caley. R., The Festival Speech Synthesis System, System documentation. Centre for Speech Technology Research University of Edinburgh, Edition 1.4, for Festival Version 1.4.3, 2002.

[12] Gussenhoven, C., and Rietveld T., "A target-interpolation model for the intonation of Dutch". *Proceedings of the ICSLP*, Banff, Canada, pp. 1235-1238, 1992.

[13] Boersma, P. and Weenink, D., Praat, a system for doing phonetics by computer, version 3.4. Institute of Phonetic Sciences of the University of Amsterdam, Report 132., 1996.

[14] Matin, E., Shao, K. and Boff, K., "Saccadic overhead: information processing time with and without saccades", *Perceptual Psychophysics*, 53, 1993, pp. 372-380.

Perspectives for Articulatory Speech Synthesis

Bernd J. Kröger

Department of Phoniatrics, Pedaudiology, and Communication Disorders,
Medical Faculty of Aachen University, 52074 Aachen, Germany

bkroeger@ukaachen.de

Abstract. Articulatory speech synthesis currently has two perspectives. (i) Technical perspective: Due to progress in common computer hardware (general increase in computation rate) and software (usability of compilers and simulation software) it is now possible to develop comprehensive phonetic models of speech production reaching nearly real-time for the calculation of acoustic speech signals. Furthermore the phonetic knowledge increased to a degree that these production models now are capable of accomplishing a good up to high acoustic quality. Limitations are mainly the control modules. In this paper we argue for a self-learning input dependent gestural control model for articulatory speech synthesis. (ii) Theoretical perspective: A comprehensive articulatory speech synthesis system capable of producing high quality acoustic output necessarily incorporates a lot of knowledge on all phonetic aspects of speech production: articulatory sound targets, typical articulatory movement strategies for realizing sounds or syllables (e.g. coarticulation), a general concept for temporal coordination of speech relevant articulatory movements (i.e. speech gestures) etc. In this paper an example for such a system will be given and a suggestion for the still open question on strategies for control concepts for high-quality articulatory speech synthesis will be proposed.

The Blizzard Challenge: Evaluating Corpus-based Speech Synthesis Techniques

Alan W Black

Language Technologies Institute,
Carnegie Mellon University, Pittsburgh, PA.
awb@cs.cmu.edu

ABSTRACT

The Blizzard Challenge was started in 2005 as a way to evaluate different corpus speech synthesis techniques on a common data set. It has been noted that it is very hard to evaluate different speech synthesis techniques when different size and quality databases are used to build a voice. To remove the variable of database size and speaker quality, we proposed a common database that all participants would use.

The Challenge itself is for participants to take the given database (or databases) and build a voice using their voice building software. After a short time, a set of test sentences are released that are to be synthesized by each participants' system.

The synthesized utterances are collected together and a web-based listening test is set up. Two types of listening tests are carried out, a simple MOS based test, and a set of understandability tests where the listener is asked to type in what they hear.

Three sets of listeners are used: speech experts (provided from the participants' groups), volunteers (collect by web advertising), and paid undergraduate native speakers.

Each year the results have been presented at a workshop where participants present descriptions of their systems, and final results are given.

The challenge has brought together groups from academia and industry from around the world. Both established groups, and new groups have been represented. The results have been both interesting and unexpected.

But we see the Challenge as a long term evolving event. Modifications in the basic structure are being considered each year. For example: how to test if speaker identity is preserved in voice conversion based systems; how can we test multi-sentence synthesis; what about multi-lingual databases; and who is going to run it.

No individual results will be presented in this talk, but overall trends will be given as well as discussion of future directions for Blizzard.

A more detailed description of the motivation and details of the challenge is described in [Black and Tokuda 2005].

All the presentations including anonymized results are also available on line at http://festvox.org/blizzard/

REFERENCES

Black, A., and Tokuda, K., (2005) Blizzard Challenge -- 2005: Evaluating corpus-based speech synthesis on common datasets Interspeech 2005, Lisbon, Portugal.