FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Transformation and interpolation of language varieties for speech synthesis

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der technischen Wissenschaften

by

Dipl.-Ing. Markus Toman, BSc
Registration Number 0226101

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao. Univ. Prof. Dipl.-Ing. Dr.techn. Andreas Rauber

The dissertation has been reviewed by:

(Ao. Univ. Prof. Dipl.-Ing. Dr.techn. Andreas Rauber)	(Assoc. Prof. Junichi Yamagishi, PhD)

Wien, 11.1.2016

(Dipl.-Ing. Markus Toman, BSc)

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Markus Toman, BSc
Annie-Rosar-Weg 2/8/6, 1220 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasser)

Acknowledgements

The compressed results of four years of my work in the field of speech synthesis are laid out in this thesis. These years have been exciting and a enjoyable because of all the people who made the time worthwhile and whom I would like to thank here:
Dr. Michael Pucher, my supervisor at the FTW Telecommunications Research Center Vienna, who recruited and introduced me to the scientific community, making this work possible in the first place. His unbreakable optimism and ability to keep the spirits high even in the bleakest of moments were of invaluable help.
Prof. Andreas Rauber, my supervisor at the Vienna University of Technology, for his scientific guidance and detailed feedback at all stages of my work.
Prof. Junichi Yamagishi, for acting as reviewer for this thesis and as examiner at its defense, as well as for the pleasant cooperation in project SALB.
Dr. Dietmar Schabus, office mate and like-minded person, for the numerous productive and fun hours we shared when walking this way together.
All project partners from the various involved institutions, especially Cassia Valentini-Botinhao, Sylvia Moosmüller and Erich Schmid.
All members of FTW who contributed to such a pleasant and vibrant working environment.

On a personal note, I would like to thank my lovely wife, Sandra Winter-Toman, for being the essential part of my life that she is. I am grateful to my parents, Karin and Amand Toman for always supporting my ambitions. Cheers go out to my parents-in-law, Carmen and Josef Winter, and to all my dear friends, especially but not limited to: Johann Grabner, Eduard Margulies and Stefan Vollenhofer.

Dedicated to my unborn child, which will – for the first time – see the light of the day soon after the publication of this work.

This work was supported by the Austrian Science Fund (FWF) project *Acoustic modeling and transformation of varieties for speech synthesis* (AMTV, P23821-N23) and the Austrian Federal Ministry of Science and Research (BMWF) - Sparkling Science project *Sprachsynthese von Auditiven Lehrbüchern für Blinde SchülerInnen* (SALB).

Abstract

This thesis aims to advance the field of speech synthesis by investigating and developing new concepts for acoustic modeling, transformation and interpolation of language varieties (i.e. dialects, sociolects, foreign accents). The goal is to enable systems with speech output to adapt to individual needs and preferences of their users. Transformation of language varieties aims to convert a voice model from one variety to a model in another variety while retaining the voice characteristics. Between multiple voice models of different varieties, interpolation allows to generate intermediate varieties. Both approaches are used to widen the range of speaking styles available to speech output systems. Further, two specific applications are investigated in this thesis: foreign accent reduction and the generation of intelligible fast speech for visually impaired users

Statistical parametric speech synthesis using hidden Markov models is the general framework employed here, as its properties allow dynamic speech adaptation to different contexts and situations. All presented methods are evaluated through listening tests and objective measures where appropriate. To conduct these experiments, phone sets and recording scripts for three Austrian German dialects have been created and speech corpora from selected native dialect speakers have been recorded in studio quality. We present a method for unsupervised dialect interpolation and show that listeners are able to correctly perceive the changes in degree of dialect for different settings of the interpolation parameter. We show that transformation of dialects while retaining the original speaker characteristics is possible with the methods presented here. We also compare different approaches for generation of fast synthetic speech, including extrapolation of duration models of natural fast speech. Our experiments show that linearly compressed, natural speech signals are more intelligible than naturally produced fast speech produced by our professional speakers. Also, a comparison of different approaches to generating dialectal voice models is presented. Lastly, we describe a speech synthesis software framework that was developed during the course of this thesis and which we also released to the public.

Overall, this thesis shows how adaptive modeling can be applied to control and modify the language variety of a speech synthesis system.

Kurzfassung

Diese Dissertation im Bereich der Sprachsynthese behandelt neuentwickelte Konzepte zur akustischen Modellierung, Transformation und Interpolation von Sprachvarietäten (Dialekte, Soziolekte und Akzente). Das Ziel ist es, Systemen mit Sprachausgabe die flexible Anpassung an Bedürfnisse und Präferenzen der Nutzer zu ermöglichen. Transformation von Sprachvarietäten zielt darauf ab, ein Sprachmodell von einer Varietät in ein Modell einer anderen Varietät zu transformieren, wobei die Stimme des Sprechers bzw. der Sprecherin erhalten bleiben soll. Interpolation erlaubt die Erzeugung von Zwischenvarietäten aus Modellen mehrerer Varietäten. Beide Ansätze ermöglichen die Erweiterung der Vielfalt an Sprechstilen, welche Sprachausgabesystemen zur Verfügung stehen. Des weiteren werden in dieser Dissertation zwei spezielle Anwendungsfälle untersucht: automatische Reduktion von Fremdakzenten und die Produktion verständlicher, schneller Sprache für sehbehinderte und blinde Nutzer.

Statistisch-parametrische Sprachsynthese, insbesondere mittels Hidden-Markov-Modellen, erlaubt dynamische Adaptierung von Sprache an unterschiedliche Kontexte und Situationen, und ist daher das grundsätzliche Paradigma, welches in dieser Dissertation Anwendung findet. Alle präsentierten Methoden wurden durch Hörexperimente und, so angebracht, mittels objektiver Metriken untersucht und bewertet. Zur Durchführung dieser Experimente wurden Phone und Aufnahmeskripten für drei österreichische Sprachvarietäten definiert und Sprachkorpora von ausgewählten Sprechern und Sprecherinnen in Studioqualität aufgenommen. Wir präsentieren eine Methode zur automatischen Interpolation von Sprachvarietäten und zeigen, dass die dadurch erzeugten, graduellen Übergänge zwischen den Varietäten vom Menschen korrekt wahrnehmbar sind. Weiters zeigen wir, dass die vorgestellten Methoden Transformation von Sprachvarietäten ermöglichen, während dabei die Stimmcharakteristika erhalten bleiben. Ausserdem vergleichen wir verschiedene Ansätze zur Erzeugung schneller, synthetischer Sprache, unter anderem mittels Extrapolation von Dauermodellen natürlicher, schneller Sprache. Unsere Experimente zeigen, dass linear komprimierte, natürliche Sprachsignale verständlicher sind als natürlich produzierte, schnelle Sprache professioneller Sprecher und Sprecherinnen. Des weiteren präsentieren wir einen Vergleich verschiedener Methoden zur Erzeugung dialektaler Sprachmodelle. Ausserdem stellen wir ein Software-Framework zur Sprachsynthese vor, welches im Zuge dieser Dissertation entwickelt und veröffentlicht wurde.

Zusammengefasst zeigt diese Dissertation, wie adaptive Modellierung von Sprache verwendet werden kann, um Sprachvarietät und Sprechstil von Sprachausgabesystemen zu kontrollieren und zu modifizieren.

Contents

CHAPTER 1

Introduction

Speech synthesis is the artificial production of human speech. It enables a natural way for human-computer interaction and cognitive user interfaces [141]. While the synthesis of natural sounding, neutral style speech can be achieved using todays technology, adaptation of speech synthesis to different contexts and situations still poses a challenge [28]. This thesis aims to advance the field of speech synthesis by investigating and developing new concepts that enable systems employing speech output to adapt to user preferences and needs, particularly in context of social and regional language varieties and artificial fast speech for visually impaired users. The methods presented here are based on the statistical parametric paradigm of speech synthesis by employing Hidden Markov Models (HMMs).

1.1 Motivation

Nowadays speech synthesis is used in an increasing number of applications. Examples are speech output of personal digital assistants, announcements in public transport, screen readers for people with visual impairment, virtual avatars in entertainment and spoken dialog systems in customer support [141]. Producing natural sounding speech that can be adapted to different situations and user preferences can help to increase acceptability and usability of a speech output system. An example would be a dialog system using speech recognition and synthesis that adapts to the language variety, mood and preferences of the user.

Throughout this thesis we refer to regional language varieties as “dialects”, and to the language varieties spoken by second language learners as “foreign accents” or just “accents”. An example for a dialect is the Innervillgraten dialect, a variety of Austrian German spoken by people who grew up in Innervillgraten in Eastern Tyrol. A foreign accent on the other hand is for example the variety of Austrian German spoken by a person who grew up with French as their native language. We also use the term “sociolect” to refer to a language variety spoken by a specific social group. For example, in Vienna, language varieties are differentiated rather socially than regionally [79]. The language variety is an important part of the persona

of a voice-based user interface since "there is no such thing as a voice user interface with no personality" [14]. Cohen et al. define the term persona as the "standardized mental image of a personality or character that users infer from the applications voice and language choice", therefore speech synthesis is an essential part of a spoken dialog system's persona [14]. The language variety also influences our evaluation of speakers' attributes like competence, intelligence, and friendliness – imitation positively affects the perceived social attractiveness of the speaker [1].

Transforming a voice from one language variety to another allows to have a single (e.g. corporate) voice adapting to a multitude of users. Also, transforming from dialectal or accented speech to standard language becomes dialect or accent reduction, which has possible applications in language learning. It has been shown that it is beneficial for language learning when students are able to hear their own voice in standard language [7]. Interpolation allows even more fine-grained control over the voice characteristics by controlling the degree of the language variety in the output speech.

Another issue in adapting speech output to user preferences is the production of fast synthetic speech. Screen readers are an essential computer interface for blind users [77]. Therefore, blind individuals become increasingly capable of understanding speech reproduced at considerably high speaking rates [64, 120], allowing them to scan information more efficiently.

1.2 Problem statement

To improve versatile speech synthesis systems as motivated in the previous section, we identified the following research problems:

Synthesis of intermediate language varieties

Generating intermediate language varieties allows a more fine-grained control over the output of a synthesizer and the adaptation of the synthesis system to different users and contexts. For example, different degrees of dialectal or accented speech can be produced to match the preference of a specific user. Possible applications include entertainment, tourism, linguistic analysis or language learning. The difficulty is to produce intermediate voice models that are meaningful, intelligible and also linguistically correct. There are different levels of abstraction to be considered for this, such as the phonetic transcription, the statistical models, the feature vectors and the speech waveform.

- **Problem 1**: Given data of speaker A in two language varieties V_1 and V_2, a parameter α specifying the degree of interpolation and the phonetic transcription of the target utterance U, synthesize an output speech sample of an intermediate variety $I(V_1, V_2, \alpha)$.

This problem is treated in Chapter 4.

Acoustic modeling of language varieties

Modern speech technology has to be able to deal with language varieties which occur in natural human speech [28]. To build speech synthesis systems that are able to use a range of different

language varieties it is important that we have methods that allow for quick development of these voice models. Methods based on speaker adaptation (see Chapter 2) are therefore a natural choice. The lack of freely available, high-quality dialectal speech resources makes the creation of synthetic dialect voices expensive and time-consuming. The problem is now to augment a small set of dialectal speech resources with a larger data set of standard language or other dialects.

- **Problem 2**: Given speech data of multiple speakers $S_1 ... S_m$ in different varieties $V_1 ... V_n$ of the same language, generate synthetic speech for a specific speaker S_i with higher quality than when having only data in the variety of speaker S_i.

This problem is treated in Chapter 5.

Transformation of language varieties

Speech resources for a single speaker are usually not available for many language varieties. Transformation of language varieties means to transform a speaker's voice to another variety for which no recorded speech data exist. For example, using Standard Austrian German speech data of a speaker and building a Viennese speech model with the same voice characteristics.

This is useful for acoustic modeling with limited speech resources. It can be applied in language learning scenarios where users can listen to their own voice in a variety they would like to learn [7] (e.g. learning standard language for accent speakers), for entertainment purposes and to allow systems that dynamically adjust the language variety of their (e.g. corporate) voice to match the preference of the user. The difficulty is to isolate language variety information in voice data from speaker characteristic features.

- **Problem 3**: Given data of speaker A in language variety V_1, generate synthetic speech in language variety V_2 without having V_2 speech data from speaker A available.

This problem is treated in Chapter 6.

Synthesis of fast speech

Visually impaired users employ synthesis of fast speech to perceive more information in a given time span. Users can be trained to understand fast speech not intelligible for untrained users. The simplest approach to just linearly compress the speech signal does not resemble natural fast speech production and might not always result in intelligible speech signals. The problem is to incorporate background knowledge of natural fast speech production to produce highly intelligible fast speech.

- **Problem 4**: Given a set of voice models $M_1 ... M_n$ in different speaking rates, a phonetic transcription of an utterance U and a parameter α specifying the output speaking rate, synthesize a speech sample of U in the desired speaking rate α.

This problem is treated in Chapter 7.

1.3 Aim of this work

This thesis aims to advance the field of speech synthesis by investigating and developing new concepts for acoustic modeling, transformation and interpolation of language varieties (i.e. dialects, sociolects, accents). The goal is to enable systems with speech output to adapt to individual needs and preferences of their users. This not only aims to increase the acceptance of the system, but also to improve applications like computer-assisted language learning [27] or screen readers for visually impaired users [77].

1.4 Research process

For our research process, several phases were defined and applied to each problem. The phases were not aligned as a waterfall model [5], i.e. they were not necessarily disjunct temporally or in terms of content. Insights in later phases potentially triggered reevaluation and adaptation of strategies defined in previous phases.

The phases were defined as follows:

- **State of the art assessment**, with the goal to identify unsolved problems and open issues. Results are presented in Chapter 2.
- **Problem definition**, with the identified problems formally defined in Section 1.2.
- **Approach and hypothesis definition**, with the specified hypotheses for each problem given in Sections 4.1, 6.1, 7.1 and 5.1
- **Corpus building**, to create the dataset needed to test the evaluations. This process is described in Chapter 3.
- **Method development**, with the resulting methods described in the respective Chapters 5, 6, 4 and 7.
- **Experiment design, execution and analysis** to test the hypotheses. The general guidelines are described in Section 1.5, the actual experiments in the respective Chapters 5, 6, 4 and 7.
- **Reflection** to analyze and interpret results and implications of the conducted experiments. Specific discussions on the results for each problem can be found in the respective Chapters 5, 6, 4 and 7. A broader view on the implications of the results and on future work can be found in Chapter 8.

1.5 Experiment design, execution and analysis

To test the defined hypotheses, experiments were designed and conducted. This usually involved listening tests with users but also objective metrics when possible. Before each experiment, the target audience (for example experts, native speakers or visually impaired users), number of listeners, data set, test set and distribution of sound samples to listeners were defined. All trials

were performed double-blind. Most listening tests were conducted using manually designed web interfaces with 60–200 tasks for each listener. Depending on the experiment, these tasks involved listening to one or more sound samples, entering text or rating speech samples using radio buttons or sliders. Experiments involving visually impaired users were conducted at the "Bundes–Blindenerziehungsinstitut Wien", a school for blind located in Vienna. There we used accessible web interfaces and also physically assisted the listeners when necessary. Results from the experiments include Word-Error-Rates derived from text input, preference choices and ratings. They were visualized using boxplots, graph plots, (stacked) bar plots and presented in tables. Objective analyses, for example of distortion measures, were conducted when appropriate. Various statistical tests for significance were employed to aid in interpretation. Experiments were conducted for all problems and are described in the respective Chapters 5, 6, 4 and 7.

1.6 Technology

The technology used for our experiments is based on the EMIME [49] version of the HMM-based speech synthesis system HTS [41, 144], written in the C and Perl programming languages. The methods presented in this thesis have either been incorporated into HTS itself or implemented as stand-alone Python scripts operating on HTS data files. Listening tests were implemented as web interface in Python, HTML and JavaScript. Evaluation data was stored in SQLite databases. Statistical analyses were performed using Python, R, Matlab and the Speech Recognition Scoring Toolkit (SCTK). During the course of this work, we have also developed a speech synthesis framework in C++ which is presented in Chapter 3.

1.7 Additional material

Additional material to this thesis, including software, sound samples and utterances from the evaluation, can be found online at `http://mtoman.neuratec.com/thesis/`.

1.8 Publications

The content of this thesis has been partially published in the journal articles and conference proceedings listed below. All contributions were subject to a peer-review process with at least two reviewers assessing the scientific quality.

1. M. Toman, M. Pucher, S. Moosmüller, and D. Schabus. "Unsupervised and phonologically controlled interpolation of Austrian German language varieties for speech synthesis". In: *Speech Communication* 72 (2015), pp. 176–193

2. C. Valentini-Botinhao, M. Toman, M. Pucher, D. Schabus, and J. Yamagishi. "Intelligibility of time–compressed synthetic speech: compression method and speaking style". In: *Speech Communication* 74 (2015), pp. 52–64

3. M. Toman and M. Pucher. “Evaluation of state mapping based foreign accent conversion”. In: *Proceedings of the 16th Conference of the International Speech Communication Association (INTERSPEECH)*. Dresden, Germany, 2015, pp. 304–308

4. M. Toman and M. Pucher. “An Open Source Speech Synthesis Frontend for HTS”. in: *Proceedings of the 18th International Conference of Text, Speech and Dialogue (TSD)*. Lecture Notes in Artificial Intelligence. Plzeň, Czech Republic, 2015, pp. 291–298

5. C. Valentini-Botinhao, M. Toman, M. Pucher, D. Schabus, and J. Yamagishi. “Intelligibility Analysis of Fast Synthesized Speech”. In: *Proceedings of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Singapore, 2014, pp. 2922–2926

6. M. Toman, M. Pucher, and D. Schabus. “Multi-variety adaptive acoustic modeling in HSMM-based speech synthesis”. In: *Proceedings of the 8th ISCA Workshop on Speech Synthesis (SSW)*. Barcelona, Spain, 2013, pp. 83–87

7. M. Toman, M. Pucher, and D. Schabus. “Cross-variety speaker transformation in HSMM-based speech synthesis”. In: *Proceedings of the 8th ISCA Workshop on Speech Synthesis (SSW)*. Barcelona, Spain, 2013, pp. 77–81

8. M. Toman and M. Pucher. “Structural KLD for Cross-Variety Speaker Adaptation in HMM-based Speech Synthesis”. In: *Proceedings of the 10th IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA)*. Innsbruck, Austria, 2013, pp. 382–387

In addition, the the following publications are closely related to this thesis but are not treated herein.

9. M. Pucher, M. Toman, D. Schabus, B. Zillinger, C. Valentini-Botinhao, and J. Yamagishi. “Influence of speaker familiarity on blind and visually impaired childrens perception of synthetic voices in audio games”. In: *Proceedings of the 16th Conference of the International Speech Communication Association (INTERSPEECH)*. Dresden, Germany, 2015, pp. 1625–1629

10. M. Pucher, V. Xhafa, A. Dika, and M. Toman. “Adaptive speech synthesis of Albanian dialects”. In: *Proceedings of the 18th International Conference of Text, Speech and Dialogue (TSD)*. Lecture Notes in Artificial Intelligence. Plzeň, Czech Republic, 2015, pp. 158–164

1.9 Structure of this work

This thesis is structured as follows:

- Chapter 2 covers the state of the art in speech synthesis with focus on HMM-based speech synthesis and existing work related to the topics of this thesis.

- Chapter 3 treats the process of data gathering and preprocessing to build the speech corpora for the experiments in this thesis and also describes a speech synthesis software framework that we developed and released.
- Chapter 4 presents our approach to Problem 1, describing a method for unsupervised interpolation between language varieties. It compares different variations of the method, including an extended method that integrates phonological knowledge.
- Chapter 5 presents our approach to Problem 2, investigating different techniques for acoustic modeling of language varieties for speech synthesis.
- Chapter 6 presents our approach to Problem 3, describing different methods developed for cross-variety transformation and the experiments conducted for evaluation thereof. The focus lies on transformation between dialects but Section 6.5 also covers experiments on reduction of foreign accent.
- Chapter 7 presents our approach to Problem 4, comparing speeding up synthetic speech by means of linear compression with a novel method employing extrapolation of duration models in HMM-based speech synthesis.
- Chapter 8 summarizes the discussion sections from the previous chapters and reflects on the findings. The chapter also gives an outlook and concludes the thesis.

CHAPTER 2

State of the art

This chapter provides an overview on the state of the art of the methods on which this thesis is based upon. This encompasses a brief introduction to speech synthesis in general and to HMM-based speech synthesis in particular, as well as the state of the art in modeling, transformation and interpolation of language varieties. Lastly, one section is dedicated to the production of synthetic fast speech. A thorough introduction to speech synthesis in general can be found in [24, 108]. Overviews on HMM-based speech synthesis are given in [59, 111, 132, 146].

2.1 Text-to-Speech

Text-To-Speech (TTS) is the process of computationally generating speech from an input text [108]. TTS systems typically consist of two components: text analysis (frontend) and speech synthesis (backend). Text analysis aims to convert text written in standard orthography to a representation that incorporates phonetic and linguistic information. For example, a simple text analysis module might convert words of a language into phone sequences (a "letter-to-sound system"). Speech synthesis (or "waveform generation") is then responsible for generating a speech waveform from this representation [111]. Speech is generated with certain speaker characteristics like pitch, loudness and prosody. The sum of these characteristics will further be referred to as "voice". Over the years, different approaches to generate speech have emerged with formant synthesis as the first genuine synthesis technique. The authors of [24] define formant synthesis as follows:

> Formant synthesis is often called synthesis-by-rule; a term invented to make clear at the time that this was synthesis "from scratch". [...] Formant synthesis adopts a modular, model-based, acoustic-phonetic approach to the synthesis problem. The formant synthesiser makes use of the acoustic tube model, but does so in a particular way so that the control elements of the tube are easily related to acoustic-phonetic properties than can easily be observed.

Formants are spectral peaks of the sound spectrum and contain enough information for humans to distinguish between speech sounds [25]. Formant synthesizers simulate formant frequencies and amplitudes, typically depending on manually created rules [104]. They tend to reliably produce intelligible and consistent speech while retaining a small memory footprint. Naturalness on the other hand is difficult to achieve as the produced speech typically sounds very robotic [108].

With the advent of inexpensive and powerful computing capabilities, data-driven techniques became feasible and started to replace the rule-based approach. Especially concatenative synthesis became very popular and is still in widespread use in commercial applications at the time of writing. An example for an early concatenative method is diphone synthesis [44]. While the term "diphone" refers to an adjacent pair of phones, typically the second half of the first phone and the first half of the second phone are used to build a diphone synthesis system. The basic idea is to build a database of all diphones of a language and to concatenate the respective diphones when synthesizing a sentence. In unit selection based synthesis, databases of larger speech unit examples with varied prosodic and spectral characteristics [42, 94] are used. To select the best speech units for a given utterance, the Viterbi algorithm [123] is used to minimize target and transition costs between selected units. Target costs define the match of a speech unit from the database with the synthesis specification, transition costs the quality of concatenation of consecutive units. Unit selection based methods are able to produce high quality, natural sounding speech at the expense of a large memory footprint and little flexibility in modifying speech characteristics [11, 108]. The hight quality output of unit selection led to a widespread in commercial TTS systems [6, 15, 21].

2.2 HMM-based speech synthesis

With the trend of data-driven speech synthesis an approach termed "statistical parametric speech synthesis" emerged in the 1990s, giving rise to the popular approach of HMM-based speech synthesis [29, 109]. The authors of [111] define HMM-based speech synthesis as follows:

> "In this approach, several acoustic parameters are modeled using a time-series stochastic generative model. Statistical parametric speech synthesis which uses a hidden Markov model (HMM) as its generative model is typically called HMM-based speech synthesis."

A typical HMM-based speech synthesis system models each unit of speech (usually phones) of a speech corpus using a single N-state, left-to-right HMM. A series of acoustic feature vectors is extracted from the speech signal of each utterance. An acoustic feature vector is a concatenation of acoustic features such as mel-frequency cepstral coefficients [18, 30] and fundamental frequency, often also including first- and second-order time derivatives thereof. The feature vectors for each utterance are used to estimate the observation probability density functions (PDF) of each phone using embedded training [146], resulting in the N-state, left-to-right HMMs. Please note that all instances of a phone in the data corpus are used to train a single HMM. An example of a 3-state left-to-right HMM is shown in Figure 2.1.

The state transition probabilities a_{ij} define the duration of each state when generating the state sequence q. This approach has serious disadvantages that can be overcome by using a a

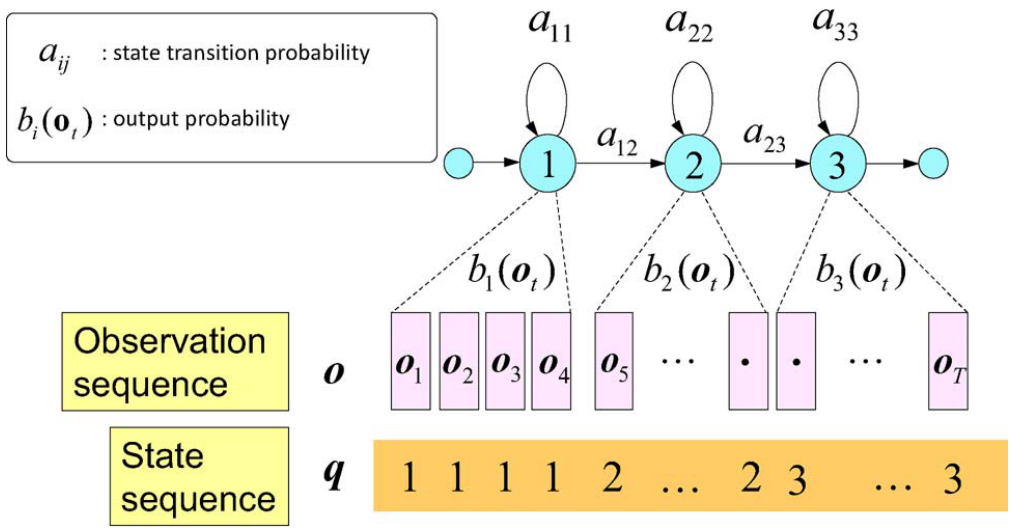

Figure 2.1: Example 3-state, left-to-right HMM (image from [111]).

method for explicit duration modeling, which will be discussed in a later section. The state output distributions b_i are usually represented as single multivariate Gaussian distributions or Gaussian mixture models [89], producing observation vectors o_t for the state sequence q.

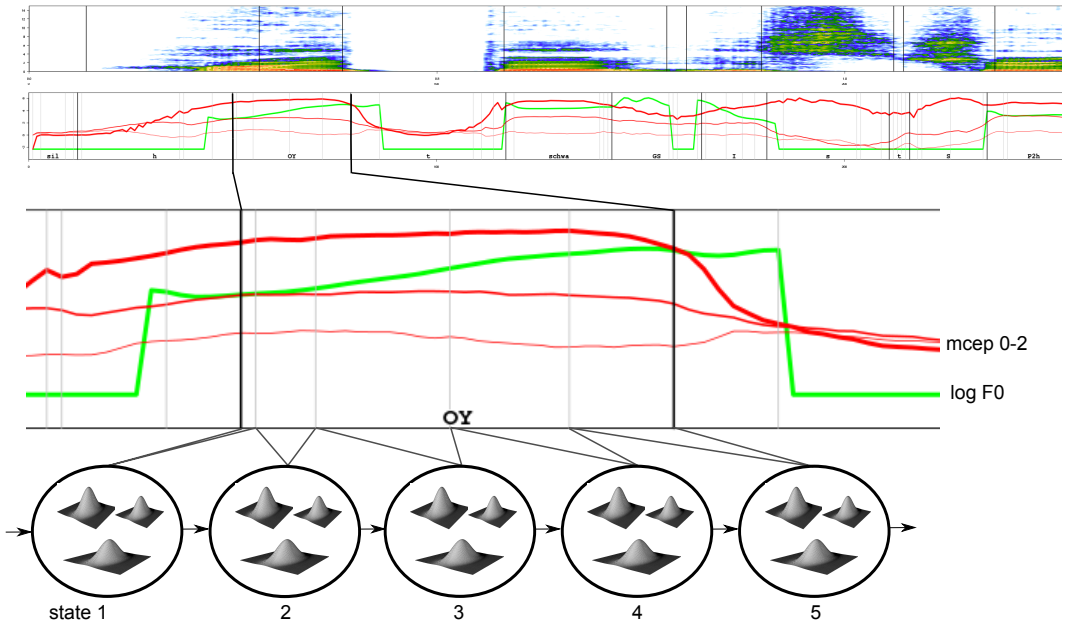

Figure 2.2: Relation of speech signal to HMM.

In the systems that are investigated in this thesis, the following acoustic features are used:

Mel-frequency cepstral coefficients [18, 30], the fundamental frequency F_0 (modeled as multi-space probability distribution [110]) and band-limited aperiodicity measures [48]. Each acoustic feature type used for modeling is commonly referred to as a stream [111]. The streams are modeled simultaneously using one PDF per feature stream. The PDFs themselves are usually represented by single multivariate Gaussian PDFs or Gaussian Mixture Models (GMMs). This relation is illustrated in Figure 2.2 for an example sentence. The top row shows the speech signal in time-frequency representation (i.e. the spectrogram) where the X axis represents time and the Y axis the frequencies. The second row presents the first three extracted mel-frequency cepstral coefficients and $logF_0$ of the speech signal for the sequence of phones. These phone borders can either be annotated manually or estimated using automatic alignment procedures [55] which align a known sequence of phones to the speech signal. The third row zooms in on the diphthong "OY" (IPA: oi), also showing the alignment of features to HMM states as vertical lines. The last row represents a 5-state HMM where each state output distribution is estimated from the acoustic features shown in row 3. This HMM can therefore be used to produce the diphthong 'OY".

Training

Given a sequence of acoustic feature vectors $\boldsymbol{o} = [\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, ..., \boldsymbol{o}_T^\top]^\top$, the training aims to find the HMM parameters λ_{max} that most likely produced the sequence $\boldsymbol{o}$. This is shown by Equation 2.1.

$$\lambda_{max} = \underset{\lambda}{\operatorname{argmax}}\, P(\boldsymbol{o}|\lambda) = \underset{\lambda}{\operatorname{argmax}} \sum_{\forall \boldsymbol{q}} P(\boldsymbol{o}, \boldsymbol{q}|\lambda). \tag{2.1}$$

Note that q is again an HMM state sequence. The number of possible sequences grows exponentially with the number of observations. The Baum-Welch algorithm [3, 86] is widely used to find the optimal HMM parameters λ_{max} in an efficient way. It converges to a local maximum and is a variant of the Expectation-Maximization algorithm [19] which can be used for maximum-likelihood estimation of latent variable models [23].

Synthesis

To synthesize a given utterance, the input text is converted to a sequence of phonetic units including contextual information that controls the selection of context dependent HMM state models (more details in Section 2.4). These HMMs (each consisting of N state models) are then concatenated to construct the utterance HMM λ. From this the most likely observation sequence $\boldsymbol{o}_{max}$, as shown by Equation 2.2, is generated.

$$\boldsymbol{o}_{max} = \underset{\boldsymbol{o}}{\operatorname{argmax}}\, P(\boldsymbol{o}, \boldsymbol{q}|\lambda). \tag{2.2}$$

The state sequence q is given by $q = \operatorname{argmax}_q P(q|\lambda)$.

Note that because of the Markov assumption, the most likely output $\boldsymbol{o}_{max}$ is just the sequence of the PDF means for all states. Therefore [60] introduced dynamic features, which are typically the first- and second-order time derivatives of the acoustic features. These derivatives are typically approximated by discrete differences of preceding and succeeding feature values.

To calculate the most likely sequence in this case, the algorithm of Tokuda et al. [109] can be used. A good overview on parameter generation can also be found in [147].

Example

A concrete example of a TTS system is given in Figure 2.3, showing a block diagram of the HMM-based speech synthesis system HTS used for training of a single speaker voice model.

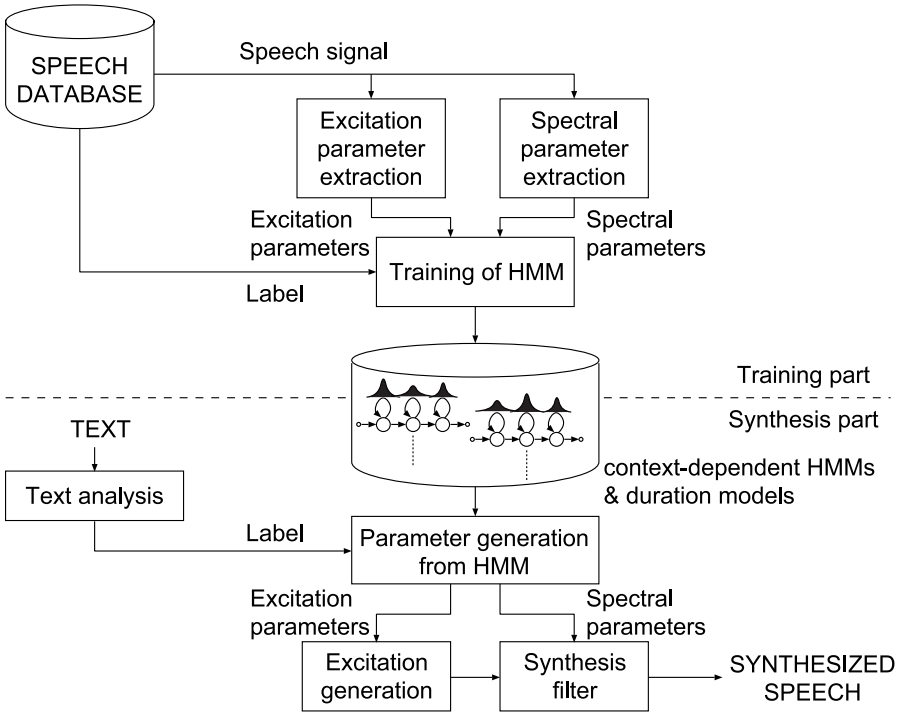

Figure 2.3: HMM-based speech synthesis system (HTS) used training and synthesis of a single speaker voice model, figure from [144].

The system input in the training phase consists of a training set of speech signal waveforms and corresponding labels from a speech database. Labels contain symbolic representations of the speech signal (i.e. phones) and contextual information including phonetic and linguistic features. In HTS, they are represented as a single string per phone. Labels and waveforms are used to train the voice model. In the synthesis phase, text analysis procedures are used to generate labels from natural text. These labels are then input to the generation process, generating acoustic feature vectors from the voice model. In Figure 2.3, "excitation parameters" are generated. In the context of the work of this thesis, these are referred to as the fundamental frequency F_0. Also, the voice model in Figure 2.3 contains context-dependent HMMs and duration models. These are explained in the following sections.

2.3 Duration modeling

In HMMs, the state duration is given by the state transition probabilities a_{ij}, which model an exponential decaying distribution as shown by Equation 2.3.

$$p_k(d) = (a_{kk})^{d-1}(1 - a_{kk}) \tag{2.3}$$

$p_k(d)$ is the probability of d consecutive observations in state k.
This is not an adequate representation of the temporal structure of speech as phone durations tend to be normal distributed [56]. Therefore, explicit duration modeling in the form of the semi-Markov structure was proposed [56, 138, 148]. They employ Hidden Semi-Markov Models (HSMMs) where the transition probability p_k of a certain state k is modeled by a Gaussian distribution $\mathcal{N}(\mu_k, \sigma_k^2)$. This violates the Markov property that the transition probability must only be dependent upon the current state, hence "semi-Markov". When using HSMMs, the probability of the next state also depends on the duration already spent in the current state. So in an HMM the transition probability is always constant, independently of the previous states. On the other hand when using HSMMs with an explicit Gaussian distribution, the probability of a transition to another state decreases once the number of previous self-transitions surpasses the mean μ_k of the distribution. HSMMs for speech recognition have already been proposed in [86].

Assuming the HSMM state transition goes from left-to-right without skipping states, it is possible to find the state sequence from the model by using the state duration probability, which in the HSMM paradigm is explicitly modeled by a distribution, as shown in Equation 2.4:

$$\boldsymbol{q}_{max} = \arg\max_{\boldsymbol{q}} P(\boldsymbol{q}|\lambda) = \operatorname*{argmax}_{\boldsymbol{q}} \prod_{k=1}^{K} p_k(d_k) \tag{2.4}$$

where $p_k(d_k)$ is the probability of duration d_k in state k, i.e. the probability that state k emits d_k observations; K is the number of states of the utterance HMM λ. [138] proposed the state durations that maximize Equation 2.4 as given by Equation 2.5:

$$d_k = \mu_k + \rho\, \sigma_k^2 \tag{2.5}$$

where μ_k and σ_k are the mean and the variance of the duration distribution of state k and ρ is a parameter to control the speaking rate. The values d_k then define the number of states and therefore the number of emitted acoustic feature vectors. More details on controlling the speaking rate can be found in Chapter 7, which deals with synthesis of fast speech.

2.4 Context-dependent modeling

As the acoustic realization of a phone varies greatly with its context, a possibility is to train HMM models not only for specific phones (monophone models) but for phones in specific contexts. For example, if two previous phones, the center phone and two subsequent phones are used to define the context, this is called a quinphone model. Other typically used contextual features are for example the position of the phone in the current word or sentence, the number of syllables in the current word etc. Also, prosodic information can be associated with the observation,

for example syllable stress or intonation changes when a sentence is uttered as a question. This type of meta-information associated with a acoustic feature vector is commonly called a "label" in the literature. If it contains contextual information, it is referred to as a "full-context label". All contexts used by default in the HTS system for the English language are listed in [146]. Ideally we would like to train a single HMM state model from multiple training instances for each possible combination of contextual features. This quickly becomes difficult due to the "curse of dimensionality" [4] – exponentially more training samples are required with each introduced contextual feature. So instead of training state models for all possible contextual feature combinations, a popular approach is to cluster the acoustic features and then train a single state model for each cluster. This leads to another issue: When synthesizing, only a full-context label but no acoustic features are available. If the label was not contained in the training data, the associated cluster can not be identified. To cope with this problem, decision tree based clustering has been proposed [75, 139] The acoustic features are clustered according to a set of "questions" on the contextual features.

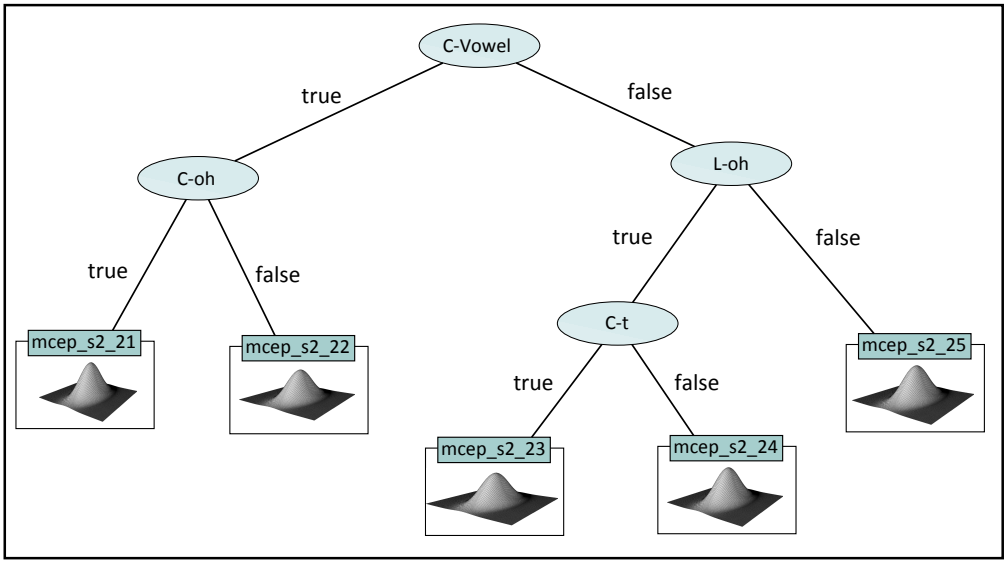

Figure 2.4: Example decision tree for context clustering.

An illustrative example for this can be seen in Figure 2.4. In this figure, "C-Vowel", for example, is a question that means "is the center phone a vowel?", "L-oh" stands for "is the previous phone oh?" where "oh" is a symbol for a specific phone. The leaf nodes then represent the clusters. A question is therefore a combination of a context feature and a specific split. For example, the "center phone is a vowel?" question defines a split by enumerating all phones which are vowels, which then form the "true" subtree. All phones not in this set form the "false" subtree.

Decision trees are constructed by iteratively selecting the question which separates the acoustic features best. This is illustrated in Figure 2.5 and can be quantified by impurity measures including variance, log likelihood or entropy [108]. It can be seen that the left question in

Figure 2.5 better separates the acoustic space than the right question and would therefore be selected when using acoustic within-class distance as impurity measure. A stopping criterion specifies when the splitting process ends. Examples for stopping criteria are a defined decrease in impurity or a minimum occupancy for each cluster (i.e. the number of data points in the cluster). Another popular criterion is Minimum Description Length (MDL) [91] which has also been incorporated into HMM-based speech synthesis [137].

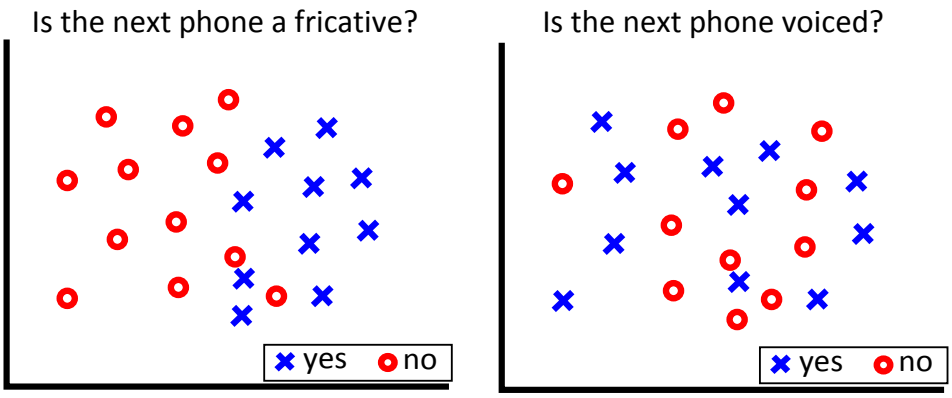

Figure 2.5: Impact of question selection on decision tree based clustering. The axes represent the acoustic feature space, the icon type represents the classification of the data point according to the given question.

For each acoustic feature stream and for each state, a decision tree is generated. A 5 state HMM with 3 streams would therefore result in 15 different decision trees and the number of HMM state models is given by the number of resulting clusters. Figure 2.4 illustrates a decision tree for the mel-cepstral feature stream and HMM state number 2. So for example, "mcep_s2_22" is the cluster with ID 22 and might contain a 40-dimensional PDF of mel-frequency cepstral coefficients. For example, a single mel-cepstral decision tree for our Austrian German average voice model had 13,808 leaf nodes.

A label to be synthesized is classified by all decision trees, yielding a PDF for each feature stream and state, representing the HMM output probabilities $b_i(o_i)$ as previously shown in Figure 2.1. Please note that durations are typically also treated as a feature stream and are also clustered accordingly. Classifying a label using a duration decision tree then yields a 1-dimensional Gaussian PDF representing the duration of the HSMM state model.

2.5 Average voice modeling

Average voice modeling is used to integrate data of multiple speakers into a single voice model, then to adapt models of a specific target speaker thereof. As many other concepts in HMM-based speech synthesis, this idea has originated in the field of speech recognition [20, 31, 54]. Adaptation allows deriving a high-quality voice model of a specific target speaker from an average voice model of multiple speakers. The advantage is that only a small amount of training data for the target speaker (also referred to as adaptation data) has to be available to produce

a voice model of higher quality [107]. A very popular approach for adaptation is to estimate linear transformations, applied to the average voice model PDFs, that maximize the likelihood of the adaptation data. One of the first methods used for this was Maximum Likelihood Linear Regression (MLLR) [106] but several other approaches emerged since then. A comparison of adaptation algorithms can be found in [134].

One issue of average voice models is the potentially dominant influence of individual speakers, for example when there is much more training data available for a specific speaker than for the others. Also in decision tree based clustering, some nodes or even subtrees might contain data of only a single speaker. This problem is treated by Speaker-Adaptive Training (SAT), proposed for speech synthesis by Yamagishi and Kobayashi [133]. SAT aims to reduce the influence of speaker dependence when training an average voice model. It assumes that the relation between average voice and speaker voices can be expressed as linear transformation and then simultaneously optimizes the transformations as well as the average voice model parameters so that the likelihood of the training data is maximized. They also employ "shared-decision-tree-based context clustering" to avoid speaker bias in single decision tree nodes by assuring that each node in the decision tree contains data from all speakers.

Average voice modeling and adaptation is an essential method for this thesis and is especially relevant for the dialect transformation techniques presented in Chapter 6 and the different approaches for acoustic modeling in Chapter 5.

To further improve average voice modeling, clustering of speakers was evaluated [17]. They assessed perceived similarities between 30 female speakers using a listening test, then applied multiple linear regression to identify speaker similarity automatically. Another listening test was used to assess the adaptation of a target speaker from a variety of average voice models. They found that using an average voice model comprising speakers similar to the target speaker was preferred to a global model encompassing all available speakers. However, average voice model using speakers selected automatically did not perform as well as the models resulting from manual speaker selection. Automatic selection of speakers for average voice models would also be helpful in training better dialectal average voice models but has not been investigated further in this thesis.

2.6 Modeling of language varieties

A current trend in the field of speech synthesis is to improve from neutral speech synthesis to dynamic adaptation of speech styles [28]. This enables the dynamic adaptation of voices to different contexts as in emotional speech [8, 105, 135] and also the transformation of a voice to different languages as needed for modern speech-to-speech translation systems. There has been much research on multi-lingual speech synthesis recently [49, 76, 84, 85, 128, 143].
Although emotional and multi-lingual speech as well as natural intonation are active areas of research, there is little work devoted to language varieties. Recent research efforts in the context of language varieties for speech synthesis are reviewed in [93]. While variety modeling is a basis for many application scenarios where a natural and realistic persona design is necessary, we have to cope with the problem that often no standardized orthographic form is available and existing speech resources for these varieties are rare. According to [93] we can distinguish

between fully-resourced and under-resourced modeling as well as different application fields. Fully-resourced modeling refers to modeling of languages and varieties with clearly defined phonetics and available speech resources. This is especially true for widely spoken languages like English. On the other hand, languages and language varieties spoken by few people are often under-resourced.

In fully-resourced modeling, Richmond et al. described how to generate pronunciation dictionaries based on morphological derivations of known words [90]. This useful for languages with a large set of affixes or which make heavy use of word-compounding (for example germ. "Apfelkuchen", engl."apple pie") as it is difficult to cover all derivations and combinations of words in a pronunciation dictionary. They reported that in preliminary experiments for 75% of tested words, their method produced the correct, fully-specified transcription. This can be used as replacement or extension to existing grapheme-to-phoneme rules to obtain contextual information on out-of-vocabulary words and could be beneficial for building an actual dialect synthesis system.

Nguyen et al. described the development of an HMM-based synthesizer for the modern Hanoi dialect of Northern Vietnamese, describing special challenges they encountered [73], comparable to our process of acquiring our dialect corpus described in Chapter 3. They employed an extensive text analysis module dealing with the phonetic particularities of Vietnamese to produce the full-context labels. For voice model training and synthesis they used standard state-of-the-art HMMs. They conducted subjective listening tests to assess different characteristics of the system. For evaluation of general quality, listeners rated 48 sentences produced by this system, recorded samples and samples produced by a unit selection system using the same data corpus. For ratings from $1-5$, with 5 being the best quality, they obtained mean ratings of 3.61, 4.82 and 2.80 respectively. This shows that the output of the HMM-based system was preferred to the unit selection system but still clearly distinguishable from natural speech. Wutiwiwatchai et al. described the process of building a speech corpus for the Thai language [130]. They recorded 60 to 100 sentences from 200 speakers so that not only all possible diphones of Thai are covered but also that the distribution of those units matches the distribution in the original text source. They also describe the difficulty that written Thai does not require spaces between words. An comprehensive overview on speech technology for Thai can be found in [131].

For developing synthesizers for under-resourced languages, different methods have been developed to aid the process of data acquisition and annotation. One study evaluated the combination of different dictionary learning techniques with a smaller dictionary available for bootstrapping [34]. Starting out with a small, hand-crafted dictionary, their approach is to first train a letter-to-sound system which is used to generate the remaining pronunciations of the training data. These are then used to train an acoustic voice model for speech recognition. The letter-to-sound system also provides a list of the N-best pronunciations for each word. The acoustic model is now used to find the most likely pronunciation from this list for each word. This process is then repeated to iteratively refine the pronunciation dictionary. They found that for Spanish, which has a strong correlation between orthography and phonetics, the original letter-to-sound models were very accurate but missed common alternate pronunciations. For English the letter-to-sound models trained from the initial dictionary were very inaccurate and benefited greatly from the proposed procedure. In their experiments, they used 1.000 words from a 5.000

word dictionary for bootstrapping. The presented method could increase the Word Recognition Accuracy from 41.38% for the small bootstrap lexicon to 43.25%, compared to 44.35% when using the full original dictionary. This method could be used to increase the size of dialectal pronunciation dictionaries but was not investigated further in the course of this thesis.

Watts et al. developed methods and tools for (semi-)automatic data selection and front-end construction for different languages, varieties and speaking styles e.g. from audio books [124]. Results from their work were used in [101] where these tools were applied on "found speech" to create a standardized multilingual corpus. For dialectal synthesis, such methods are useful for easy acquisition and annotation of dialect data, which is currently a time-consuming process. In further related work, a phoneme-to-phoneme conversion technique that uses decision trees to automatically convert pronunciations between American, British and South African English accents was developed [58]. As a transcription of the dialect utterance is required for most techniques presented here, this method could be used to automatically generate the phonetic transcription for less-resourced dialects from a fully-resourced variety.

2.7 Cross-lingual transformation

The aim of cross-lingual transformation is to transform data in language L_1 to language L_2 while retaining the original speaker characteristics. These methods can be applied to cross-variety transformation, which is treated in Chapter 6, and can roughly be categorized as three basic approaches.

Frame-level transformation

Qian et al. [84] developed a method that operates on the level of single acoustic feature frames. Using frequency warping to perform voice conversion, the voice of a speaker in L_2 is converted to the voice of another speaker in L_1. This method of transforming the voice of a speaker is also known as vocal tract length normalization (VTLN) [53]. The frequency warping function used in this method is a piecewise-linear function which is obtained by mapping formant frequencies from L_1 to L_2 speaker for the same, long vowels. This included manual vowel selection and checking of the extracted formants. To complete system uses frequency warping to change the voice characteristics of a native L_2 speaker to the voice characteristics of the L_1 speaker. The warped acoustic feature frames are then used to find the most similar frames in the recordings of the L_1 speaker. These frames are then concatenated to generate training data for a HMM-based synthesizer. For their experiments they used data from a bilingual speaker, consisting of 1.000 Mandarin and 500 English sentences. The data to be transformed consisted of 750 English sentences. For the subjective listening test, 50 sentences in general domain and 50 semantically unpredictable sentences were played to 8 native Mandarin speakers. They found that the output of this method was preferred to a comparable state mapping approach. This study relied on bilingual data, manual vowel and sentence selection and fine tuning of the warping function. While instead of using bilingual data, similar vowels in different languages might be selected and the selection itself could be automatized, it is unclear how this would affect the results of the evaluation.

State-level transformation

This category of methods operates on the level of HMM state models. As our work on cross-variety transformation is based on state mapping, this approach will be described in more detail in Chapter 6. Wu et al. proposed a state mapping based approach for cross-lingual speaker adaptation [128]. The method that finds mappings between the most similar HMM state output PDFs of two average voices, one in L_1, the other one in L_2. They described two variants for using this mapping to achieve cross-lingual transformation: data mapping and transform mapping. Given adaptation data in L_1 of a speaker A, in data mapping, the HMM state output PDFs of the adaptation data (in L_1) are mapped to the PDFs in L_2. Then regular speaker adaptation is used to adapt A from the average voice model of L_2. In transform mapping, linear transformations from the L_1 average voice to the adaptation data of speaker A are estimated using MLLR as proposed by Tamura et al. [106]. In regular speaker adaptation, these transformations would be applied to the average voice model of L_1 to generate a voice model of speaker A. For transform mapping, the transformations are applied to L_2 according to the state mapping. Their experiments were based on English and Japanese speech from 2 male and 2 female speakers each, about 1 hour recordings per speaker. 50 sentences of one male Japanese speaker were used to transform this speaker from Japanese to English. They compared the state mapping methods to a phone mapping approach in two subjective listening tests. 8 listeners were presented 15 synthesized samples, randomly selected from a pool of 40 samples, and had to rate quality and speaker similarity. They found that transform mapping was rated the highest for quality, data mapping for speaker similarity.

Liang and Dines [57] proposed an extension to this method. They used a decision tree to cluster the HMM state output PDFs into phonetic categories. Mapping is then restricted to only occur within these clusters. They reported a reduction of mel-cepstral distortion but only subtle improvements were noticed by listeners in a subjective listening test. This experiment involved 21 listeners rating naturalness and 17 listeners rating speaker similarity. Our work presented in Chapter 6 investigates the effects of alternative approaches to constrain the possible state mappings by phonetic information.

Phone-level transformation

Another study investigated cross-lingual speaker adaptation using phone mapping [127]. They manually mapped Chinese Initials and Finals (the components of Chinese syllables) to English phonemes using two different strategies: one-to-one mapping and one-to-sequence mapping. These differ in the sense that a Chinese Initial or Final is mapped to either a single English phoneme or a sequence of phonemes respectively. The mapping is applied to the labels of the Chinese speaker data and then adaptation from an English average voice model is performed. For their experiments they trained an English average voice model from about 1 hour speech from each of 4 male and 1 female speakers. They evaluated different configurations, including different amounts of adaptation data (10, 100 and 1000 Chinese utterances), adaptation of different feature streams and the one-to-one and one-to-sequence mapping variants. They found that 10 adaptation utterances were enough to produce speech of reasonable quality but using more data produced more stable and clearer speech, so they conducted the other evaluations using

100 utterances. Adaptation of the fundamental frequency F_0 (instead of using the original F_0 features from the speaker) improved the transformation but also introduced a slight degree of unnaturalness. One-to-sequence phone mappings were slightly preferred to one-to-one phone mappings by 8 listeners in a subjective listening test involving 40 sentences per listener. They concluded the work with an outlook to state mapping based methods, which performed better in subsequent studies [128].

2.8 Accent conversion

Accented speech is ubiquitous in everyday communication. It has been shown that adaptive speech synthesis [133] can be applied to create voices that retain the speaker accent. Studies have shown that adaptation from an average voice model using 105 sentences is able to overrule the influence of the accent of the average voice and produce accented voice models [47, 125]. Foreign accent conversion aims to convert from one speaker variety to another one. The main direction of conversion is from accented speech to some form of standard where foreign accent conversion becomes accent reduction.

In [7] it was shown that it is beneficial for language learning when students are able to hear their own voice producing native-accented utterances. In their work they use contours of the fundamental frequency $F0$, local speech rate, and intensity from the same utterance of a native reference speaker and copy them to the learners' speech signals. This is achieved by analysis, modification and synthesis. First they time-align the sequences by means of Dynamic Time Warping (DTW) [70, 88] and Pitch Synchronous Overlap and Add (PSOLA) [13]. The $F0$ contours were combined by multiplying the normalized $F0$ contour of the reference sample with the mean $F0$ of the learners' speech signal. They found the resynthesized, less accented stimuli to be more effective than natural stimuli of the non-accented reference speaker in a language learning experiment. This experiment involved two groups of Italian learners of German where one group was provided with training samples in their own, manipulated voice and the other group was provided with correctly pronounced training samples in their teacher's voice. After the training phase, both groups had to reproduce learned sentences and were judged by German native speakers. The samples produced by this method were especially helpful in providing feedback for learning correct stress positions. This method only allows for the modification of accented recordings for which reference samples by a native speaker exists. Therefore it does not provide speech synthesis of arbitrary text but the results show that accent conversion can be beneficial in language learning scenarios.

In another technique [26], accented portions of speech are replaced with alternative less accented segments from the same speaker corpus by finding the segment that is closest to the respective segment from a native reference speaker. This concatenative method relies on detection of the most accented speech segments within an utterance. This is achieved by creating a phonetic transcription of the accented speech, including the misspellings, which is then compared to the transcription of the native speaker data. To fully automatize this method, an automatic detection of accented speech segments would be needed. They report a 20% reduction in perceived (American English) accent compared to the natural accented speech. This experiment was conducted using crowd-sourcing where potential listeners were required to pass a screening test in

which American English accents had to be identified. In the actual experiment, 20 listeners had to rate the degree of foreign accent in 40 sample utterances each. They also observed a strong coupling between accent and speaker identity. We used a similar approach in our evaluations, as that capable native listeners had to rate the degree of accent or dialect in presented sound samples.

Another study investigated the role of consonantal segments in perception of foreign accent. They compared different approaches to generate English speech with specific Spanish accented consonants [52]. For the first approach, a highly competent bilingual speaker produced English speech with a set of specified consonants pronounced in a Spanish accent. The second approach involved semi-automatic replacement of consonants in English speech recordings with consonants from the Spanish recordings. For the third approach, a bilingual HMM voice model was trained where the accent identity was added as a feature to the full-context labels. They conducted a subjective listening test with one group of 9 native English speakers and another group of 21 native Spanish speakers. Each participant listened to 108 words produced by all 3 approaches, with and without accented consonants. They had to type each presented word and also rate the degree of foreign accent. For both groups and all approaches, intelligibility was significantly lower for the samples with accented consonants compared to the non-accented samples. Still, the group of native Spanish speakers achieved higher scores in understanding accented samples than the group of native English speakers. This also suggests that foreign accent reduction increases intelligibility of the produced speech. Both groups rated the degree of accent lower for the synthetic speech samples compared to the first two approaches involving recorded samples. We have also seen this effect in our experiments in Chapter 6. The authors considered the possibility of using extrapolation techniques to emphasize foreign accent as an application for foreign language learning. The methods presented in Chapter 4 could also be used to produce extrapolated dialect or foreign accent, which we did not investigate.

Accent conversion can be considered a special case of cross-variety transformation when transforming from a language variety (the accent) to standard language, so investigate this case in Chapter 6.

2.9 Interpolation

Interpolation of voice models aims to generate speech that combines voice characteristics of the interpolated models. It was first applied in HSMM-based synthesis for speaker interpolation [140], generating "in-between speakers" with characteristics of the input speakers. Another application of interpolation is emotional speech synthesis. Tachibana et al. presented a system for controllable emotional speech synthesis by interpolating between voice models of different emotions and speaking styles (i.e. neutral, joyful, sad and rough) [105].

In language variety interpolation, Pucher et al. have shown how to interpolate between phonetically different dialects in a supervised way [81]. In this method, a manually defined phone-mapping between Standard Austrian German and the Viennese dialect was used. Linear interpolation was then performed between the manually mapped HMM state models. Evaluation tests showed that listeners actually perceive the intermediate varieties created by interpolation as such. In Chapter 4 we extend this method to operate completely unsupervised, we integrate

background knowledge on phonological processes and also investigate further Austrian dialects. In another study, Astrinaki et al. [2] have shown how to interpolate between clusters of accented English speech within a reactive HMM-based synthesis system. "Reactive" means that the interpolation parameter can be controlled while the system is outputting speech. In this method, phonetic differences between the accents were not considered (i.e. the same set of phones was used for all accents). This means that a given text is translated to the same sequence of phones and therefore HSMM states for each accent, so the state mapping is implicitly given. The method we present in Chapter 4 allows interpolation between utterances of different length and with different phonetic content.

Another study [129] applied the interpolation techniques from [140] to adjust the accent level for English words in Thai texts. Their approach is to interpolate between a pure English accent and English uttered using only phones from the Thai inventory. They described the problem of one-to-many mappings occurring when interpolating between HMM state sequences of different lengths. They solve this problem by duplicating states and recalculating state durations. Our dialect interpolation method in Chapter 4 uses a similar approach to deal with one-to-many mappings but the process of finding the mappings is very different. This is because the application in [129] involves only differences in pronunciation, where phones in one accent are represented by one or more phones in the other accent. In the case of the dialects that we investigated, the mapping between phones is not always evident. For overall preference, their experimental setup consisted of 22 native Thai listeners who were presented one sentence at a time with 5 different accent degrees. The listeners then had to select their preferred sample for 10 sentences. They found that the listeners preferred the interpolated samples to the pure English or pure Thai samples. Another interesting finding was that the listeners tended to prefer English words commonly used in Thai to be pronounced with a stronger Thai accent and uncommon words with a stronger English accent.

2.10 Fast speech

When human beings produce fast speech, they compress vowels more than consonants [32]. Also both word-level [46] and sentence-level [78] unstressed syllables are compressed more than stressed ones. Another important aspect of fast speech is the reduction of pauses. [35] claim that reducing pauses is possibly the strongest acoustic change when speaking faster, most probably due to the limitations of how much speakers can speed up their articulation rate [37]. These observed changes can be the result of an attempt to preserve the aspects of speech that carry more information, i.e. stressed segments. However, the presence of pauses has been shown to contribute positively to intelligibility [95], suggesting that a massive reduction of pauses should also be avoided.

Production of fast speech is more difficult for human beings, making the speech sound mumbled and less intelligible. It has been shown that natural fast speech (typically around 1.56 times faster than normal speech) is harder to process in terms of reaction time, and also that it is preferred less than linearly compressed speech [45, 46]. Linearity here refers to the same compression rate used for a full sentence, i.e. vowels, consonants, silence and speech are compressed at the same rate. Linearly compressed speech was found to be more intelligible and preferred

over a nonlinearly compressed version of speech where fast speech prosodic patterns were mimicked [46] at a high speaking rate (2.85 times). [45] claims that possibly the only nonlinear aspect of natural fast speech duration changes that can improve intelligibility at high speaking rates is the removal of pauses but only when rates are relatively high, as it was shown in results obtained by another nonlinear compression method, the MACH1 algorithm [16]. This method is based on the acoustics of fast speech with the addition of compressed pauses. At very fast speaking rates (2.5 to 4.1 times) MACH1 improves comprehension and is preferable to linearly compressed speech but no advantage was found at the fast speech speaking rate (1.4 times) [38]. Moers et al. [63] reported with a different fast speech corpus (2.0) that linearly compressed plain speech is more intelligible than fast speech but for an ultra fast speaking rate (rate was not reported) a linearly compressed version of their corpus of fast speech was more intelligible than the plain counterpart and chosen to be more natural. Although there is a clear relation between the speed of articulation and the benefit of fast speech over linearly compressed speech, there is also something to be said about the different strategies that each person adopts when speaking fast. Due to speaker variability, benefits seen at one rate for one speaker might not necessarily be applicable to recordings of another speaker.

[51] found that fast natural speech was more intelligible than fast speech generated by a formant-based synthesizer and that the intelligibility gap grows with the speaking rate. [62] reported that the use of units made of multiple phones which are prone to heavy coarticulation creates speech that is more preferable in terms of intelligibility and naturalness than using only units of single phones when building a unit selection synthesizer with fast data. More recently, [103] evaluated the intelligibility of a wider range of synthesizers: formant, diphone, unit selection and HMM-based. It was found that the unit selection systems were more intelligible across speech rates. However, in this evaluation, however, the evaluated synthesizers were based on different speakers and the compression methods adopted by each system were not reported. Literature on fast synthesized speech also focuses on the effect on blind listeners as expert users of this technology. Intelligibility of formant and concatenative synthesizers was compared in [102] for individuals with early onset blindness. It was found that recognition levels depend on the speaking rate and factors such as age, familiarity with the particular synthesizer and voice. In general, authors found that familiarity with the synthetic voice can alleviate the intelligibility drop that grows with rate, as a particular formant synthesizer voice outperformed other voices.

To improve natural fast speech synthesis for blind individuals, [80] evaluated a method that uses interpolation between a model trained with normal and a model trained with fast speech data. The most successful method applied interpolation only on the state durations and using the acoustic features of the speech in normal speaking rate. This approach for generating fast speech is the basis for the interpolation method used in the study which we present in Chapter 7.

CHAPTER 3

Speech synthesis data and framework

This chapter describes planning, gathering and post-processing of the speech corpora used throughout this thesis. It also describes the SALB framework, a software framework for HMM-based speech synthesis that was developed during the course of this thesis. It was released to the public[1] under an MIT-style license, making it freely available for personal and commercial use as long as the copyright notice is retained.

Parts of this chapter have been published previously in [112].

3.1 Data recording and processing

The Standard Austrian German and Viennese corpora were recorded in previous projects [79, 81] and reused here. The Bad Goisern and Innervillgraten dialect corpora as well as the corpus of Austrian German accents were created during the course of this work.

All recordings were conducted in studio setting at the Acoustics Research Institute of the Austrian Academy of Sciences. Sound samples were recorded at 44100 Hz, 16 bits/sample. The voice model training process was also performed using these specifications, if not stated otherwise. Cutting and selection of utterances was performed manually. Noise cancellation and volume normalization was applied to the recordings. Synthesized samples used in the experiments were also volume normalized. A 5ms frame shift was used for the extraction of 40-dimensional mel-cepstral features, fundamental frequency and 25-dimensional band-limited aperiodicity [48] measures. For each speaker, the data was split into a training and a test set. The criteria for the test set selection was that each phone had to occur at least twice in the set. Test sets for the data sets presented here were between 20 and 50 sentences per speaker.

For our phone set definitions (see Tables 3.1 and 3.3) we also assembled a table of phone classifications (e.g. vowel, consonant, fricative, stop) which was used to automatically generate the sets of questions used in the decision tree based clustering [75, 139]. So for example, every phone which was classified as vowel in this table would be included in the set of phones for

[1] SALB framework available at `http://m-toman.github.io/SALB/`

which question “IsVowel” yields true. Our question sets consisted of roughly 2,100 possible questions for each variety and were adapted from the English question coming with the HTS demonstration scripts (listed in [111]).

3.2 Standard Austrian German

In previous projects [81], 4387 sentences at a normal and 198 sentences at a fast speaking rate of a male, professional speaker of Standard Austrian German (SAG) were recorded over multiple sessions. Both sets were phonetically balanced and selected from linguistic standard corpora and from news papers. SAG refers to the variety spoken by the upper social classes of the big cultural centers located predominantly in the Middle Bavarian region [66, 67, 100], shown in Figure 3.1 as “Central Bavarian dialects” and spanning large parts of Northern and Eastern Austria. For the fast sentences, the speaker was asked to speak as fast as possible while staying intelligible. The set of phones used for this corpus is shown in Table 3.1. Detailed information on SAG phonetics for TTS can be found in [72].

Category	Phones (IPA)
Vowels (monoph.)	ɑ ɑː ɒ ɒː ɐ e ɛ eː i ɪ iː ɔ o oː øː æː ɐ œ œː ə u ʊ uː ʏ y yː
Vowels (monoph.) nasalized	ɒ̃ː ɔ̃ː ɑ̃ːɐ̯ œ̃ː
Diphthongs	aɪ̯ ɒːɐ̯ ɑːɐ̯ ɒɪ̯ ɑʊ̯ ɛɐ̯ ɛːɐ̯ ɪɐ̯ iɐ̯ iːɐ̯ ɔɐ̯ ɔːɐ̯ oːɐ̯ ɔʏ̯ øːɐ̯ œɐ̯ ʊɐ̯ uːɐ̯ ʊːɐ̯ ʏɐ̯ yːɐ̯
Plosives (stops)	b̥ d̥ g̊ k ʔ p t
Nasal stops	m n ŋ
Fricatives	ç x f h s ʃ v z ʒ
Approximants	j
Trill	r
Lateral approx.	l

Table 3.1: Standard Austrian German phone set.

3.3 Austrian German dialects

For the experiments presented in Chapters 5, 6 and 4, we used a corpus of three dialects: the dialect of Innervillgraten (IVG) in Eastern Tyrol, the dialect of Bad Goisern (GOI) in the South

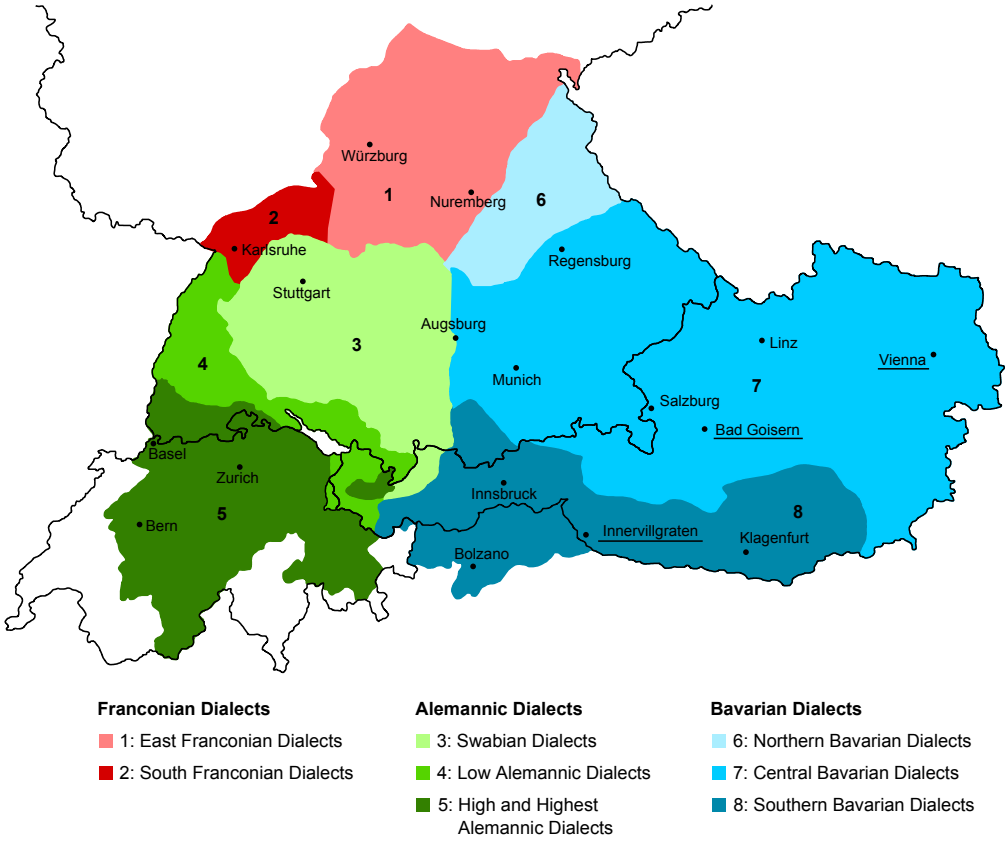

Figure 3.1: Upper German dialects, with the locations of the recorded dialects highlighted. Image from [96].

of Upper Austria, and the dialect of Vienna (VD). As can be seen in Figure 3.1, IVG belongs to the South Bavarian dialect group, GOI belongs to the (South)-Central Bavarian dialect group and VD to the Central Bavarian dialect group. Since the speakers were genuine dialect speakers, meaning they were raised in the respective dialect and learned SAG in school, SAG spoken by these speakers contained also regional features. Therefore, the SAG variety produced by the GOI and IVG speakers is referred to as Regional Standard Austrian German (RSAG).

SAG orth.	**Morgen kommt meine Schwester.**
SAG phon.	ˈmɔɐ̯gŋ̩ kɔmt maɐ̯nɛ ˈʃvɛstɐ
RSAG phon.	ˈmɔɐ̯gŋ̩ kɔmt maɐ̯nɛ ˈʃvᵋestɐ
GOI phon.	ˈmɔ ɾɪŋ̩ kimd̥ mãɐ̯̃ ˈʃvᵋesd̥a

Table 3.2: Example for differences between SAG, RSAG as spoken in Bad Goisern, and GOI.

To illustrate this, in Table 3.2, the difference between SAG, RSAG as spoken in Bad Goisern, and GOI is shown for an example sentence. In RSAG, /ɛ/ of **Schwester** is slightly diphthongized, a process which is not allowed in SAG. Also, the diphthong in **meine** differs between the two

varieties. These two features cue a regional variant of SAG which, in turn, is still completely different from the dialect version of the model sentence.

Ten dialect speakers, gender balanced, were recruited for the GOI and the IVG corpus, respectively. The recordings consisted of 18-20 hours of spontaneous speech, reading tasks, picture naming tasks, and translation tasks from SAG into the dialect. From these recordings, a phone set was created for each dialect (see Table 3.3). Taking into account the occurrences of each phone and the context-specific variations, 660 phonetically balanced sentences were selected. For the final recordings, two male and two female speakers for each dialect were selected according to the following linguistic criteria:

- "Native speaker", i.e., raised within the dialect
- Consistent application of characteristic phonological processes (e.g., assimilations, deletions)
- Lexical knowledge and morpho-syntactic competence

To record the 660 phonetically balanced dialect sentences, the speakers heard the dialect speech sample they were asked to utter and were also presented with an orthographic transcription close to the standard orthography. In addition, these speakers also read a corpus of 223 SAG sentences.

The VD corpus was recorded in previous projects [79, 81] and therefore the recording procedure differed slightly. Ten actors and actresses were invited for a casting in which they had to perform reading tasks in both SAG and VD. For the VD samples, they had to transform SAG sentences into the VD. Subsequently, the recordings were subjected to analysis and one male speaker who performed best was chosen for the final recording sessions. The final corpus consists of 513 phonetically balanced VD sentences and 223 SAG sentences.

Table 3.3 shows the phone sets for the different varieties. Affricates were split into two phones in (R)SAG and VD as these were already defined in a previous project. For alveolar stops, a further category had to be introduced in order to capture instances which can neither be assigned to [t] nor to [d̥]. Since consonant duration is the decisive feature in differentiating stops in Austrian dialects, we symbolize this additional category as [d̥ː] (see [68] for a discussion).

3.4 Austrian German accents

For the accent conversion experiment presented in Chapter 6), we recorded 5 female and 5 male speakers with (Austrian) German as second language (L2) and the following first languages (L1, with ISO 639-1 codes): Bulgarian (BG), UK-English (EN), Estonian (ET), French (FR), Greek (EL), Slovakian (SK), Spanish (ES), Hungarian (HU), Serbian (SR). 320 phonetically balanced sentences were recorded for each speaker from which 23 were held out as test set. This set contained the 223 SAG sentences as well as 97 additional news paper sentences. The phone set used in this corpora is shown in Table 3.3 in the (R)SAG column.

Category	(R)SAG	VD	IVG	GOI
Vowels (monoph.)	ɑ ɑː ɒ ɒː ɐ e ɛ eː i ɪ iː ɔ o oː øː æː ɐ œ œː ə u ʊ uː ʏ y yː	ɑ ɑː ɒ ɒː ɐ e ɛ eː i ɪ iː ɔ o oː œː øː æː ɐ œ ə u ʊ uː y yː	ɑ ɑː ɒ ɒː e ɛ eː ɛː i ɪ iː ɔ o ɔː oː ɐ œ ə u ʊ uː ʏ y yː	ɑ ɑː ɒ ɒː e ɛ eː ɛː i ɪ iː ɔ o ɔː oː ø øː ɐ œ u ʊ uː ʏ y yː
Vowels (monoph.) nasalized	ɒ̃ː ɔ̃ː æ̃ː œ̃ː	ɒ̃ ɒ̃ː ɑ̃ː ɛ̃ː ĩː õː æ̃ː		ɒ̃ ɒ̃ː ɑ̃ ɑ̃ː ɛ̃ ɛ̃ː ĩː ĩ ɔ̃ ɐ̃ ũ
Diphthongs	aɪ̯ ɒːɐ̯ ɑːɐ̯ ɒɪ̯ ɑʊ̯ ɛɐ̯ ɛːɐ̯ ɪɐ̯ iːɐ̯ ɔɐ̯ œɐ̯ øːɐ̯ ɔʏ̯ oːɐ̯ ʊɐ̯ uːɐ̯ ɒn̯ ʏɐ̯ yːɐ̯	ɒːɐ̯ ɪɐ̯ ɛɐ̯ ɛːɐ̯ iɐ̯ iːɐ̯ ɔɐ̯ øːɐ̯ œɐ̯ ʊɐ̯ uːɐ̯ ʊɪ̯ ʏɐ̯ yːɐ̯	aɐ̯ aɛ̯ ɑɔ̯ ɑɔɐ̯ ɑoɐ̯ eɑ̯ ɛɑ̯ eɛ̯ eɛ̯ ɛɐ̯ ɛi̯ ɛɐ̯ iɛ̯ ɪi̯ iɐ̯ ɔɑ̯ oɑ̯ ɔɛ̯ oɐ̯ ɔɔ̯ ɔɐ̯ oi̯ ɔo̯ ɔɐ̯ ɔʊ̯ œɐ̯ œø̯ uɑ̯ uɛ̯ ʊɐ̯ uɐ̯ ʊu̯ ʏɐ̯ ɔʊ̯ ɛa̯ ɛʊ̯ ui̯ ʊi̯ ʊu̯	ɑɒ̯ aɐ̯ ɑo̯ ɑɔ̯ eɑ̯ ɛɑ̯ ɛɐ̯ eɐ̯ iɛ̯ iɐ̯ ɔɐ̯ oi̯ ɔo̯ ɔɐ̯ ɔʊ̯ œɐ̯ œø̯ uɑ̯ uɐ̯ ʊɐ̯ uɐ̯ ʊu̯ yɐ̯
Diphthongs nasalized				ɑɒ̃ aɐ̃ ɑo̯ ɛɐ̃ ɔɑ̃ oɑ̃
Plosives (stops)	b̥ d̥ g̥ k ʔ p t	b̥ d̥ g̥ k ʔ p t	b̥ b d̥ d d̥ː g̥ g k kʰ ʔ t tʰ	b̥ d̥ g̥ k kʰ ʔ t tʰ pʰ
Nasal stops	m n ŋ	m n ŋ	m n ŋ	m n ŋ
Fricatives	ç x f h s ʃ v z ʒ	ç x f h s ʃ v	β ð ç x f ɣ h s ʃ v	β ç x f ɣ h s ʃ v
Affricates			b̥͡f kχ͡ ks̠ tʃ ts	b̥͡f ks̠ ts
Approximants	j	j	j	j
Trill	r	r	ɾ ɾχ	ɾ
Lateral approx.	l	l	l l	l

Table 3.3: Phone sets of the language varieties used throughout this thesis, phones represented by IPA symbols.

3.5 Software framework

The toolkit used throughout this thesis for creating HMM-based voice models is called HTS [144]. It produces HTS voice models that can be used to actually synthesize speech waveforms, using separate software toolkits. A popular, freely available framework is hts_engine [40]. The inputs to hts_engine are a HTS voice model and the full-context labels to be synthesized. Speech synthesis frontends on the other hand provide means for analyzing and processing natural language text, producing the full-context labels. In HTS, each full-context label is represented as a single string containing one phone to be synthesized as well as surrounding phones and other contextual information, including position in utterance, prosodic information etc. (see also Chapter 2). While not exclusively being frontends and not specifically targeted for HTS, popular choices are Festival [119] or Flite [12]. Both were originally built for concatenative speech synthesis but have been extended to incorporate hts_engine to synthesize from HTS voice models. Festival is a complex software framework for building speech synthesis systems for Unix-based operating systems, written in C++ and Scheme. A Festival version of our SAG voice is available online[2]. Flite was built as a lightweight alternative to Festival, written in C and targeting low-memory devices like mobile phones and embedded systems.

The main goal when creating the SALB framework was to easily allow HTS voices to be used with the Speech Application Programming Interface 5 (SAPI5). SAPI5 allows the framework to be installed on the Microsoft Windows operation system as TTS engine, making HTS voice models available as system voices to applications like screen readers, e-book creators etc. The second goal was flexible integration of new languages and phone sets. The third goal was portability to mobile devices.

Flite has been adapted for HTS in the Flite+hts_engine software [40] and due to its small and portable nature it seemed like a good fit to our requirements. Compiling Flite on Microsoft Windows is possible, allowing the implementation of the SAPI5 interface. It also compiles with minor changes on most current mobile operating systems (Android, iOS, Blackberry were tested at the time of writing). The main issue with Flite was the integration of new languages. Pronunciation dictionaries, letter-to-sound trees and other language-specifics are directly written in C and compiled into the binary.[3]

Therefore our framework integrates Flite for text analysis of English while additionally providing an internal text analysis module that can utilize pronunciation dictionaries and letter to sound trees in Festival format. For waveform synthesis, we integrate hts_engine and allow the extension by other synthesizers.

The framework also includes a free voice model of our SAG speaker corpus presented in Section 3.2. It also encompasses a pronunciation dictionary with 14,000 entries, letter to sound rules and procedures for number conversion. The framework has been used for evaluations with blind school children in [82] and to create a speech synthesis system for Albanian dialects in [83]. The following descriptions refer to the current version of the SALB framework at the time of writing: v0.8.3.

[2] SAG voice "Leopold" for Festival: http://sourceforge.net/projects/at-festival/

[3] We have published a guide on adding a new language to Flite online: http://sourceforge.net/p/at-flite/wiki/AddingNewLanguage/

3.6 Framework architecture

We abstract the components and interfaces of the TTS system, allowing components to be replaced by alternative implementations (for example, using the internal text analysis module instead of Flite or using another synthesizer than hts_engine). The general architecture of the SALB framework is shown in Figure 3.2. Frontend modules provide means to communicate with the user or other applications. For example, the user can trigger speech synthesis tasks using the Command-Line-Interface (CLI). Microsoft Windows applications can use the SAPI5 interface to use the framework in a uniformly manner together with other TTS engines. JNI allows interfacing with Java applications, e.g. an Android app. The frontend modules use the C++ Application Programming Interface (API) of the core module `manager`, which in turn coordinates the backend modules for text processing and waveform synthesis. The C++ API can also be used directly to embed the framework in other applications.

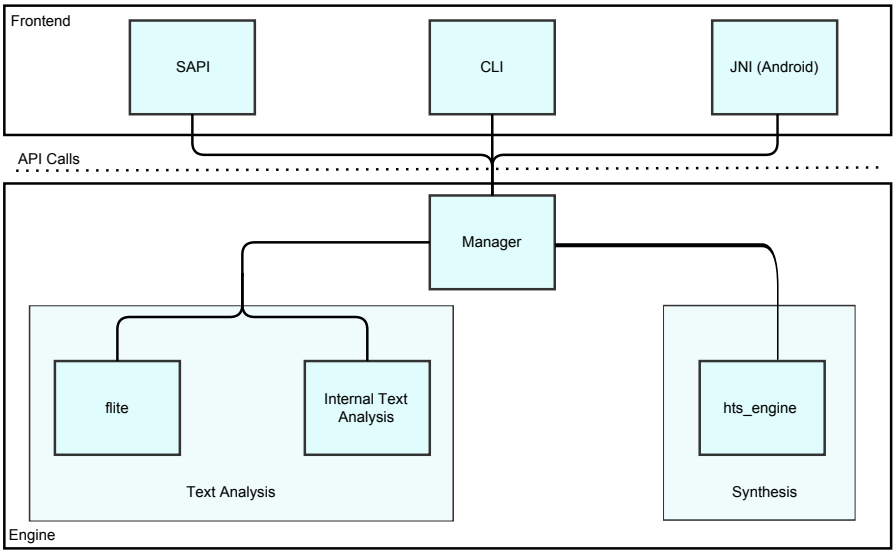

Figure 3.2: General framework architecture

Manager module and API

The core of the framework is the manager module which provides a uniform API for frontend modules or other applications. It provides abstractions for the essential elements of the speech synthesis process, shown in Figure 3.3. This API is provided by the `TTSManager` class. A `TextFragment` represents a piece of text in a given language. Each `TextFragment` has `FragmentProperties` associated with it, controlling the synthesis parameters (e.g. voice, speaking rate) for this text fragment. Multiple `TextFragment` objects can form a `Text` object. This can be used to synthesize a text consisting of fragments with different synthesis parameters (e.g. a text read by different voices). A `Text` or `TextFragment` object with associated `FragmentProperties` can be passed to a `TTSManager` object which executes the speech

synthesis process and returns a `TTSResult` object. `TextAnalyzer` and `Synthesizer` are abstract base classes for text analysis- and synthesis modules respectively.

The `TTSManager` object first selects an adequate `TextAnalyzer` object based on the value in `FragmentProperties`, specifying the text analyzer to use or a value specifying the language of the text. For example, `FragmentProperties` might explicitly enforce the use of Flite as text analysis module (i.e. an instance of `FliteTextAnalyzer`, which is a subclass of `TextAnalyzer`). Or it might provide only a language identification and the system then chooses an appropriate text analysis module by itself. The `TextFragment` is then passed to the chosen `TextAnalyzer` object, which processes the text input and returns a series of `Label` objects in a container object called `Labels`. A `Label` represents the basic unit for synthesis. For example, if the `Label` is being passed to hts_engine, it is represented as the full-context label string expected by hts_engine. In the next step, the `TTSManager` selects an adequate instance of a subclass of `Synthesizer`, again based on the information in `FragmentProperties`. At the time of writing, the only available `Synthesizer` is `HTSEngineSynthesizer`. The `Label` is then passed to the `Synthesizer` with the `Label` class being responsible for providing the desired format. Currently the only available format is the HTS label format. In the last step, the `Synthesizer` returns a `TTSResult` object, containing the synthesized waveform as `vector` of discrete samples as well as meta information.

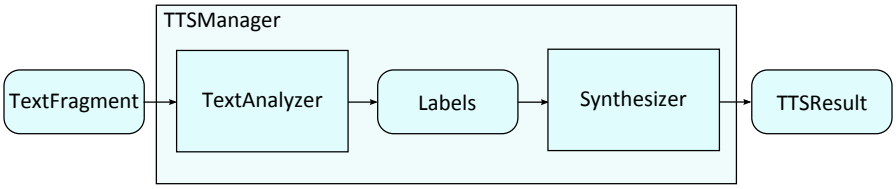

Figure 3.3: Data flow in synthesis process

Frontend modules

The following sections describe the frontend modules included in the framework at the time of writing.

Speech Application Programming Interface (SAPI)

The framework provides a SAPI5 frontend module. This allows the registration of the framework as speech synthesis engine on Microsoft Windows platforms, therefore enabling HTS voices to be registered as system voices. We provide a Microsoft Visual Studio [61] project to compile a SAPI-enabled dynamic link library (DLL) for 32-bit and 64-bit systems that can be registered with the operating system. We provide an extra tool, `register_voice`, that enters the voices in the Microsoft Windows registry. Lastly a script, that allows to create installer packages that install or uninstall the TTS engine and associated voice models, is bundled with the framework.

Command Line Interface

The distribution also contains a simple command line tool which, produces sound files from text input. It requires at minimum the text input, the language of the text, a HTS voice model if Flite is used for text analysis. If the internal text analysis module is used, it additionally requires a text processing rules file, containing pronunciation dictionary and letter-to-sound rules in Festival format.

Android

Integration in Android apps is possible through the Java Native Interface (JNI) and the Android Native Development Kit (NDK) [36]. The framework comes with Android `make` files, a JNI wrapper and a Java class, which plays the synthesized speech when given text input.

Text analysis modules

The following sections describe the text analysis modules included in the framework.

Flite

For converting English text to a series of labels for synthesis, we integrated Flite into the framework. The class `FliteTextAnalyzer` (derived from base class `TextAnalyzer`) is a wrapper converting input and output data for and from Flite.

Internal text analyzer

The distribution also comes with an internal text analysis module `InternalTextAnalyzer` (derived from base class `TextAnalyzer`). This module reads a specific rules file consisting of a pronunciation dictionary and letter to sound rules in Festival format. The pronunciation dictionary is read completely on startup but parsing happens lazily, only when specific entries are needed. Preprocessing of text (e.g. for numbers and dates) can be added by extending the `Normalizer` class which delegates to different implementations based on the chosen language. We provide a simple `Normalizer` for Austrian German (`AustrianGermanNormalizer`), handling numbers and some special characters, as well as a comprehensive rules file, consisting of a pronunciation dictionary and letter-to-sound rules, which can serve as an example on how to integrate new languages. This module internally uses a hierarchical utterance structure consisting of the classes `Phrase`, `Word`, `Syllable` and `Phone`. The `PhraseIterator` class is used to navigate this structure to build the resulting `Label` object (for example, to calculate the number of syllables in the succeeding word).

Synthesis modules

The following section describes the synthesis module in the framework.

hts_engine

This module provides a wrapper around the hts_engine library and is implemented by the class `HTSEngineSynthesizer` (derived from the abstract base class `Synthesizer`). The `Label` objects provide strings in HTS label format which are the input to hts_engine. The resulting waveform is then converted and encapsulated in a `TTSResult` object and returned to the `TTSManager` and subsequently to the caller. We have changed the algorithm for changing the speaking rate to linear scaling due to the results presentend in Chapter 7.

C++ API Example

In this section we present a minimal example on synthesizing a single sentence using the C++ API of the framework.

```
#include "TTSManager.h"

using namespace htstts;

int main() {
   std::string input = "Hello world.";
   TTSManager tts;

   FragmentPropertiesPtr properties = std::make_shared<FragmentProperties>();
   (*properties)[PROPERTY_KEY_SYNTHESIZER] = PROPERTY_VALUE_HTSENGINE;
   (*properties)[PROPERTY_KEY_TEXTANALYZER] = PROPERTY_VALUE_AUTOMATIC;
   (*properties)[PROPERTY_KEY_TEXTANALYZER_RULES] = "leo.rules";
   (*properties)[PROPERTY_KEY_VOICE_PATH]  = "leo.htsvoice";
   (*properties)[PROPERTY_KEY_LANGUAGE]    = "de-at";
   (*properties)[PROPERTY_KEY_VOICE_NAME]  = "Leo";

   // create a text fragment with given input text and properties
   TextFragmentPtr tf = std::make_shared<TextFragment>(input, properties);
   TTSResultPtr result = tts.SynthesizeTextFragment(tf);
   save_result_riff(result, "out.wav");

   return 0;
}
```

Listing 3.1: Sample for using the C++ API to synthesize a sentence

In Listing 3.1 we first define the string `input` to be synthesized, then construct a `TTSManager` object `tts` and a `FragmentProperties` object `properties`. A description of the available synthesis properties is given in the next section. Please note that we use smart pointers to the actual objects here. Next, a `TextFragment` object `tf` is constructed and the text `input` as well as the `TextFragmentProperties` `properties` are passed to it. The `TextFragment` can now either be synthesized directly or added to a `Text` object. Here we synthesize it directly by passing it to the `SynthesizeTextFragment` method of the `TTSManager` object `tts`. We retrieve a `TTSResult` object `result` containing the synthesized waveform. Using the helper function `save_result_riff`, we can save this result to

a file named `"out.wav"`. Error handling is done using exceptions, which is omitted here for simplicity.

Synthesis properties

The following properties are available for controlling the synthesis process:

- PROPERTY_KEY_SYNTHESIZER defines which synthesis module shall be used.
- PROPERTY_KEY_TEXTANALYZER defines which text analysis module shall be used.
- PROPERTY_KEY_TEXTANALYZER_RULES defines the path to the text analysis rules.
- PROPERTY_KEY_LANGUAGE defines the language for this text fragment.
- PROPERTY_KEY_VOICE_NAME defines the name of the voice to use.
- PROPERTY_KEY_VOICE_PATH defines the path to the voice model file.
- PROPERTY_KEY_VOLUME defines the volume of the synthesis (0-100).
- PROPERTY_KEY_RATE defines the speaking rate (0.5-4.0).

Please note that fragments of a longer text can also be synthesized from multiple voice models (i.e. different speakers for different parts of the text).

3.7 Extending the framework by new languages

One main goal when developing the framework was the possibility to easily integrate voice models of other languages than English. As literature aiding this process is scarce, we present some basic guidelines in the following sections.

Gathering data

Before creating a recording script, a defined phone set is needed. These already exist for many languages. If not, the inventory of all relevant phones of a language should be defined in cooperation with phoneticians. One possibility is to gather conversational speech data in the target language and then produce manual transcriptions. The granularity of the transcription is very important and has a direct impact on the quality of the final voice models. For example, diphthongs can be modeled as separate phone symbols or split into two symbols. When a phone set is defined, recording scripts can be generated either through manual transcription or by using orthographic text (e.g. from newspapers) and a letter to sound system. The recording script should contain all phones in as many triphone contexts, better quinphone contexts, as possible. In any case each phone should occur multiple times (preferably at least 10 times, considering that a set set will also be split off the training data). Given a data set of phonetically transcribed sentences, algorithms to solve the set cover problem can be employed to produce a final recording script.

From this data set, a test set can be selected by the same procedure (e.g. select the smallest set that contains each phone at least 2 times). Speakers should be recorded in studio setting and with neutral prosody. The output of this step is a phone set, recording scripts and a corpus of recorded utterances in waveforms and transcriptions.

Integration

The internal text analyzer reads rules for text processing from an input stream. These rules consist of a lexicon and a letter to sound tree. A word is first looked up in the lexicon. If it can not be found there, the letter to sound rules are used to create the phonetic transcription. A voice model and the text processing rules are sufficient for bassic speech synthesis and building SAPI5 voice packages using the framework. Extending the source code is necessary if more sophisticated text processing is required, the C++ class `AustrianGermanNormalizer` can be used as an example for this. The method `InternalTextAnalyzer::TextFragmentToPhrase` can be adapted to implement alternatives to the Festival lexica and letter to sound rules.

Lexicon

The lexicon (or pronunciation dictionary) part of the text rules is a set of mappings from orthography to phonetics. The framework uses Festival style lexica in Scheme syntax.[4] A lexicon can be derived from the data corpus used for the recording or from publicly available pronunciation dictionaries.

Letter-to-sound rules

The internal text analyzer is able to read Festival style letter to sound rules.[5] A method for building letter to sound rules from an existing lexicon can be found in [9]. For languages with an orthography very close to the phonetics, the letter to sound rules can also be hand-crafted.

Training voice models

Voice models to be used with the SALB framework can be trained using the HTS toolkit. Available demonstration scripts for speaker dependent voices can be adapted for this purpose. When using the demonstration scripts, it is necessary to replace raw sound files, labels and question files. Label files can be produced using the SALB framework once at least a lexicon containing all words from the training set is available. The question files contain questions used in the decision tree based clustering [139]. A minimal question file should at least contain all phone identity questions (e.g. for a phone "A" and quinphone models, at least the following questions should be defined: "LL-A", "L-A", "C-A", "R-A", "RR-A"). When all parameters (e.g. sampling rate) have been set, the training process can be started to produce an "htsvoice" model file that can be used with the SALB framework.

[4] Festival lexicon definition: http://www.cstr.ed.ac.uk/projects/festival/manual/festival_13.html
[5] Festival LTS rules: http://www.cstr.ed.ac.uk/projects/festival/manual/festival_13.html#SEC43

3.8 Discussion

This chapter presented the process of data gathering and preprocessing to build the speech corpora for the experiments with Austrian German language varieties. It also presented the SALB open source speech synthesis framework, a software that bridges existing tools for HTS-based synthesis like hts_engine and Flite with SAPI5 to enable HTS voices to be used as Windows system voices. It allows to load and use Festival style lexica and letter to sound rules for easy integration of new languages. The C++ API allows embedding in other applications as well as Android and iOS apps. It also includes voice model, lexicon and letter to sound rules of a male, professional speaker built from the SAG corpus presented in Section 3.2. We have also given a brief tutorial on how to train voice models for further languages and add use them with the SALB framework.

CHAPTER 4

Synthesis of intermediate language varieties

This chapter presents a method for unsupervised, gradual interpolation between language varieties. We apply this method to Austrian German dialects, allowing a speech synthesis system to control the degree of dialect in the produced speech. This enables a larger variety of speaking styles available for a given voice model (for example, a corporate voice) and allows adaptation to user preferences, increasing the acceptability of a voice. In Chapter 6 we presented a method for cross-variety speaker transformation based on HSMM state mapping. Transforming the voice of a speaker from one variety to another can be used as a basis for interpolation. For example, a single voice model could be transformed to multiple other varieties and then interpolation can be used to synthesize samples for intermediate stages, enabling a large spectrum of speaking styles with recordings available in only a single language variety. Furthermore, interpolation methods could also be used to extend existing multi-variety speech databases or speech databases with similar languages by augmenting them with interpolated data. In general, the method presented here can be applied to any interpolation of state sequences of HMM models, which makes it also applicable for facial animation [97]. In [81] it was already shown for one standard/variety pair that supervised interpolation between language varieties produces intermediate language variants that are meaningful and intelligible. For this type of interpolation a manually defined phone mapping was necessary (i.e. between which phones and HSMM states should interpolation be performed). Here we introduce an unsupervised method where the interpolation is defined on a pair of HSMM state sequences that are not necessarily of equal length or ordering. This automatic derivation of the HSMM state mapping employs DTW [70, 88] and is subsequently described in Section 4.3. Compared to [81], this method introduces one-to-many mappings between states, requiring a more sophisticated duration modeling procedure, which will be described in Section 4.4. Also, in this work we perform interpolation between Regional Standard Austrian German (RSAG) and the three dialects from Innervillgraten (IVG), Bad Goisern (GOI) and Vienna (VD), as presented in Section 3.3. The difficulty of dialect interpolation lies in lex-

ical, phonological, and phonetic differences between the varieties [93]. In this contribution we focus on interpolation of phonetic differences.
Parts of this chapter have been published previously in [115]. This work was conducted in collaboration with the Acoustics Research Institute (ARI) Vienna of the Austrian Academy of Sciences.

4.1 Problem definition

This chapter treats Problem 1, defined as:

- **Problem 1**: Given data of speaker A in two language varieties V_1 and V_2, a parameter α specifying the degree of interpolation and the phonetic transcription of the target utterance U, synthesize an output speech sample of an intermediate variety $I(V_1, V_2, \alpha)$.

Therefore a system to solve this problem has the following **input**:

- A voice model of speaker A in language variety V_1 and in V_2.
- Interpolation parameter α.
- Phonetic transcription of utterance U.

The system should produce the following **output**:

- Synthesized speech sample for utterance U in the intermediate language variety $I(V_1, V_2)$.

In this chapter we test Hypothesis 1:

- **Hypothesis 1**: Unsupervised interpolation on HSMM-based voice models can produce intermediate language variants that are meaningful and intelligible.

4.2 Intermediate language varieties

In this work, interpolation is performed between RSAG and the three dialects IVG, GOI and VD. When observing the RSAG - dialect interaction in natural speech, we can find the following characterizing alternations[1]:

1. **Phonological process**: Socio-phonological studies on Austrian varieties demonstrate that certain alternations between two varieties, usually a standard variety and a dialect, are phonetically well motivated and thus can be described as phonological processes, e.g., spirantization of intervocalic lenis stops [66] like
    - [ɑːb̥ɐ] to [ɑːbɐ] to [ɑːβɐ]
      aber (engl. “but”) or

[1] // denotes the phonological representation, [] the phonetic realization.

- [laɛd̥ɐ] to [laɛdɐ] to [laɛðɐ]
  leider (engl. “unfortunately”).

Spirantization is the process of a plosive becoming a fricative. So in the examples above, the [b̥] in [aːb̥ɐ] becomes [β] in [aːβɐ] and the [d̥] in [laɛd̥ɐ] becomes [ð] in [laɛðɐ]. These gradual transitions can be modeled by interpolation.

2. **Input-switch rules**: Other alternations lack such phonetic motivations because of a different historical development. These alternations are therefore described as input-switch rules, e.g.

   - /g̊uːt/ ↔ /guad̥/ or
   - /g̊uːt/ ↔ /gui̯d̥/
     gut (engl. “good”).

   No gradual transitions from e.g., /g̊uːt/ to /guad̥/ can be observed [22, 66]. Because of their phonetic saliency, input-switch rules are sociolinguistic markers as defined by [50], meaning they are subjected to stereotyping and social evaluation (positive or negative). Therefore, interpolation is not feasible in these cases.

3. **Pseudo-phonological process**: Many input-switch rules involve diphthongs vs. monophthongs; i.e. the standard form is a diphthong, the dialect form is a monophthong. Standard Austrian German features a vast variety of phonetic diphthongal realizations [69], so that any (slight) movement in formant frequencies is interpreted as a diphthong [65]. Sociolinguistically, the input-switch rule persists; the diphtong is the standard form, the monophthong is the dialect form, such as in the following examples:

   - /haɛs/ ↔ /haːs/
     heiß (engl. “hot”) or
   - /kɑɔ̯fsd̥/ ↔ /kɑːfsd̥/
     kaufst (engl. “(you) buy”)

   However, the gradual decrease in formant frequency movement can be elegantly captured by interpolation, without attracting negative evaluation from the listener’s part. Consequently, modeling this case using HSMM interpolation is feasible although the alternation is actually an input-switch rule.

Because of the existence of input-switch rules, it is not phonetically feasible to interpolate whole utterances. Therefore, we introduce “region-based interpolation”. This introduces another level of mappings on regions spanning multiple phones. Each region can then be defined as either (pseudo-)phonological process or input-switch rule and will therefore be treated differently. For example, the words **Ziege** (RSAG) vs. **Goaß** (dialect; engl. “goat”) might form mapped regions that should not be interpolated. This procedure is described in detail in Section 4.7.

4.3 Unsupervised interpolation

This section describes the interpolation methods used to generate intermediate language varieties.

In [81] we implemented and evaluated a *supervised interpolation* method that allows for gradual interpolation between language varieties when a phoneme mapping is given. Here we extend this method by an *unsupervised interpolation* that is based on DTW [70, 88] of HSMMs. This method implements gradual interpolation between varieties by automatically finding phone mappings.

Given the phonetic transcription (full-context labels) of utterance U in variety V, we can obtain a sequence of HSMM state models by classifying the labels using the decision tree of the V voice model (see Section 2.4). If we do this for both V_1 and V_2 we obtain two different state model sequences, called "sentence HSMMs". As we perform interpolation [140] between HSMM state models, we need a mapping between the states of the two sequences. So in a next step we calculate this mapping by applying DTW on the two state sequences, using KLD between the mel-cepstral streams as distance metric. Since the mel-cepstral parameters are modeled by k-dimensional multivariate Gaussian distributions and not Gaussian mixture models, we can use the analytic form of KLD [39]. By using a symmetric version of KLD, we ensure that the whole interpolation is also symmetric [70].

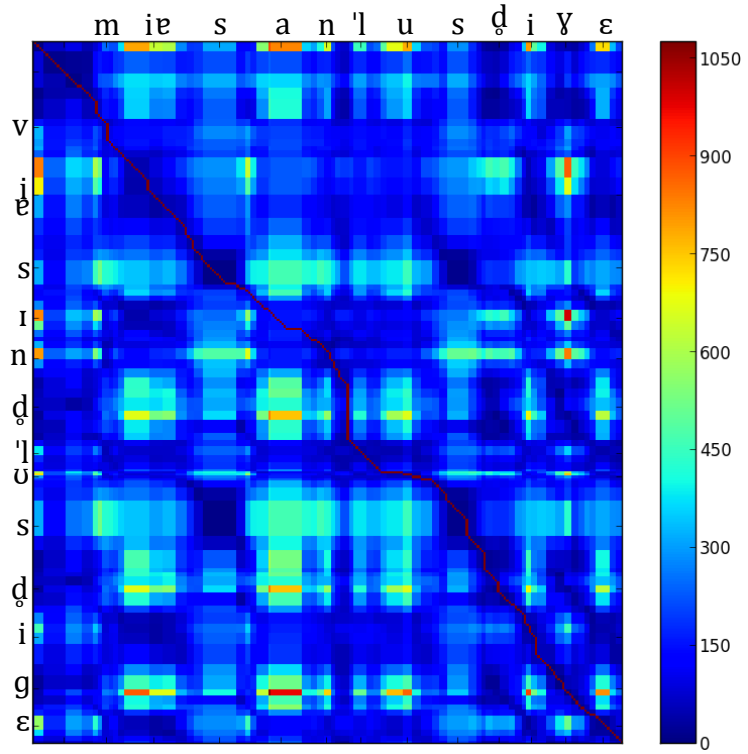

Figure 4.1: DTW result for RSAG - VD interpolation between expanded state sequences of the sequence "Wir sind lustige" (engl. "We are funny").

Figure 4.1 shows an actual DTW alignment for the sentence beginning "Wir sind lustige..." (engl. "We are funny..."). The horizontal axis represents the VD version of the sentence, the vertical axis the RSAG version. Each unit along the axes then represents a single HSMM state model and the intensity values the degree of similarity between them. We can see the optimal warping path closely following the diagonal, mapping mostly states of similar phones. For each of these mappings, a new HSMM state is generated by interpolating the mapped HSMM states. In the last step, all newly generated states are concatenated and form the sentence HSMM, which is then used to synthesize the output utterance.

It should be noted that the sentence HSMM constructed along the DTW warping path can consist of more states than the original sentence HSMMs. The interpolation endpoints ($\alpha = 0.0$ and $\alpha = 1.0$) are therefore not necessarily exactly the same as when not using the presented method.

The acoustic features models, in our case mel-cepstral, $F0$ and aperiodicity PDFs, are linearly interpolated [140] according to an interpolation parameter α. The interpolated mean μ and covariance matrix U are given by Equations 4.1 and 4.2.

$$\mu = \alpha\mu_1 + (1.0 - \alpha)\mu_2 \tag{4.1}$$

$$U = \alpha^2 U_1 + (1.0 - \alpha)^2 U_2 \tag{4.2}$$

$\mathcal{N}(\mu, U)$ is then used as output PDF of the newly generated HSMM state. The interpolation parameter α therefore controls the impact of the input PDFs in V_1 and V_2 on the resulting PDF.

In principle the same equations are also used for interpolating state durations except for one issue: Horizontal and vertical parts of the warping path indicate one-to-many mappings between states. These type of mappings have to be treated carefully to assure that the total duration of the segment is correct. Our approach to interpolate state durations in these cases is described in detail in Section 4.4.

4.4 Duration interpolation

This section describes how the durations for the final HSMM, which is constructed from the DTW warping path, are calculated.

As described in Section 4.3, the optimal warping path between two HSMM state sequences might contain one-to-many mappings. This is always true when the number of states is different for the two sequences. We have to handle these cases so that the total duration of the final HSMM state sequence is also the result of a linear interpolation between the total durations of the individual sequences with respect to α.

In the system used here [146], the duration of an HSMM state k is modeled by a Gaussian distribution $\mathcal{N}(\mu_k, \sigma_k^2)$ (see Section 2.3). The actual state duration d_k is then calculated as given by Equation 4.3.

$$d_k = \mu_k + \rho\, \sigma_k^2 \tag{4.3}$$

The parameter ρ controls the speaking rate and is set to 0.0 for our experiments, so in this case the state duration d_k equals the mean value μ_k. But as we perform interpolation on the state

durations d_k, changing the speaking rate using ρ would also be possible without modifications to the method.

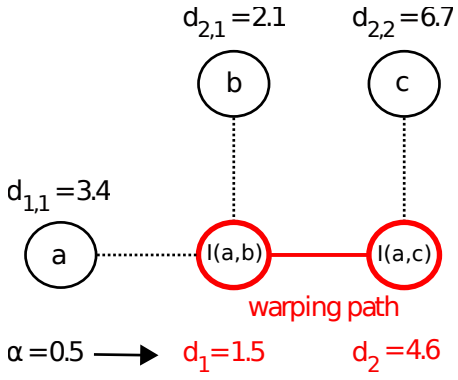

Figure 4.2: Interpolation of state durations in a one-to-two mapping.

Figure 4.2 shows an example of a one-to-many mapping. In a DTW matrix (for example shown in Figure 4.1), this would be a horizontal part of the warping path with a length of 2 units. State a with duration $d_{1,1}$ is mapped to state b with duration $d_{2,1}$ and state c with duration $d_{2,2}$. This results in the two interpolated states $I(a, b)$ and $I(a, c)$ with durations d_1 and d_2 respectively. While the acoustic features for the new states can be interpolated straightforward as described in Section 4.3, the durations have to be calculated differently. The interpolated total duration d for both states is given by Equation 4.4. With an example interpolation parameter $\alpha = 0.5$, for this example we obtain $d = 6.1$.

$$d = (d_1 + d_2) = d_{1,1}\alpha + (d_{2,1} + d_{2,2})(1 - \alpha) \tag{4.4}$$

Now we distribute the total duration d to $I(a, b)$ and $I(a, c)$ according to the relation between b and c as shown in Equations 4.5 and 4.6, resulting in $d_1 = 1.5$ and $d_2 = 4.6$.

$$d_1 = d\frac{d_{2,1}}{d_{2,1} + d_{2,2}} \tag{4.5}$$

$$d_2 = d\frac{d_{2,2}}{d_{2,1} + d_{2,2}} \tag{4.6}$$

We can generalize this approach as shown in Equation 4.7.

$$d_i = \left(\sum_{j=1}^{m} d_{1,j}\alpha + \sum_{k=1}^{n} d_{2,k}(1 - \alpha) \right) \frac{d_{1,i}}{\sum_j d_{1,j}} \frac{d_{2,i}}{\sum_k d_{2,k}}. \tag{4.7}$$

Here $\langle d_{1,1}, .., d_{1,m} \rangle$ and $\langle d_{2,1}, .., d_{2,n} \rangle$ are the two duration sequences involved in the interpolation. For a one-to-one mapping this results in $m = n = 1$, reducing the formula to standard interpolation. For one-to-many interpolation, we get either $m = 1, n > 1$ or $m > 1, n = 1$.

The final duration of a state determines the number of feature vectors produced from it. The interpolation might produce states with durations smaller than 1.0. To cope with this, we

accumulate the durations and skip states according to Algorithm 1. The algorithm loops the final HSMM state sequence. The duration of each state (as calculated before) is accumulated in $accdur$. If $accdur < 1$ (i.e. less than 1 feature vector would be created by this state) then the current state is skipped. Else the state is added to the final model and its duration subtracted from $accdur$.

Algorithm 1 Algorithm for skipping states.

```
accdur ← 0
for all duration, state in HSMM state sequence do
    accdur ← accdur + duration
    if accdur ≥ 1 then
        accdur ← accdur − duration
        add(state)
    end if
end for
```

4.5 State expansion

We investigated two different approaches how sentence HSMMs can be input to DTW:

- *Unexpanded state method* is the regular approach where the sentence HSMMs are the input to DTW. In this case, each state has an associated duration which determines the number of feature vectors that will be the output of this state.
- *Expanded state method* means that we duplicate a given HSMM state with a mean duration d to d states with duration 1. Each of these states then represents a single feature vector. This is done for all states of both sentence HSMMs.

An illustration for these two approaches can be seen in Figure 4.3 where a 5-state HSMM is shown in unexpanded and expanded form.

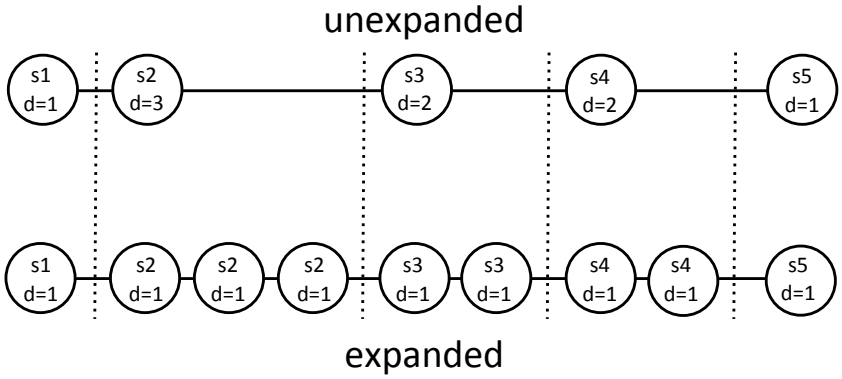

Figure 4.3: Illustration of the same HSMM in unexpanded and expanded form. d represents the duration (in feature frames) for each state.

An artificial, general example for the effect of this on the DTW result can be seen in Tables 4.1 and 4.2. In the unexpanded case in Table 4.1, the input to DTW are the two sequences $X = \langle a, b \rangle$ and $Y = \langle c, d \rangle$. In the expanded case in Table 4.2, the sequences are expanded to $X = \langle a, a, b, b \rangle$ and $Y = \langle c, c, c, d, d \rangle$. Both use the cost (or dissimilarity) function $cost(b, d) = 1 < cost(b, c) = 2 < cost(a, c) = 3 < cost(a, d) = 4$. In our case, X, Y would be the two sentence HSMMs, a, b, c, d the state output PDFs and $cost(x, y) = KLD(x, y)$. The tables represent the DTW result matrices as given by Equation 4.8.

$$\begin{aligned} DTW[i,j] := cost(X[i], Y[j]) + \min(&DTW[i-1,j], \\ &DTW[i,j-1], \\ &DTW[i-1,j-1]) \end{aligned} \tag{4.8}$$

The optimal warping path is represented by red, underlined numbers, resulting in mappings $\{\text{a} \leftrightarrow \text{c}, \text{b} \leftrightarrow \text{d}\}$ for the unexpanded method and $\{\text{a} \leftrightarrow \text{c}, \text{b} \leftrightarrow \text{c}, \text{b} \leftrightarrow \text{d}\}$ for the expanded method. This shows the greater flexibility of the expanded method as parts of a single HSMM state can be mapped to multiple other HSMM states.

b	∞	5	**4**
a	∞	**3**	7
	0	∞	∞
		c	d

Table 4.1: Dynamic-time-warping between state sequences “ab” and “cd”. State “a” is aligned with “c” and state “b” is aligned with “d”.

b	∞	10	10	10	9	**10**
b	∞	8	8	**8**	**9**	10
a	∞	6	**6**	9	13	17
a	∞	**3**	6	9	13	17
	0	∞	∞	∞	∞	∞
		c	c	c	d	d

Table 4.2: Dynamic-time-warping between expanded state sequences “aabb” and “cccdd”. States “a” are aligned with “c”, states “d” are aligned with “b” but states “c” are aligned with “a” and “b”.

In order to evaluate whether the behavior shown in Tables 4.1 and 4.2 is actually present in the interpolations of empirical speech samples, we analyzed the alignments of all interpolations of the data set used for our evaluation which we present in Section 4.8. We found that in the mapping of 12666 HSMM states in V_1 with 14902 HSMM states in V_2 there were 3656 mappings between expanded states which did not exist when using the unexpanded method. This shows that the two methods actually produce different results for the same input sequences.

The final HSMM state sequence is used as input for the parameter and waveform generation [146].

4.6 Phonetic analysis of interpolation errors

Unsupervised interpolation allows us to generate intermediate variants of utterances for any given utterance pair. In addition, it is also possible to deal with missing words. In terms of

linguistic correctness we might produce utterance variants that are wrong in the sense that such intermediate variants do not exist or that co-occurrence[2] requirements are not met.

We have conducted a phonetic analysis of the sample utterances used for the experiments described here. We identified the input-switch rules and the phonological processes involved in the interpolation from the (R)SAG input to the dialect output for the sample utterances. These results were then used in the extended method presented in Section 4.7. The analysis has been conducted by the Acoustics Research Institute Vienna and can be found in [115] but is not necessary to understand the following sections.

4.7 Phonological model

Based on the results of the phonetic analysis described in Section 4.6, we extended our interpolation algorithm to handle input-switch rules for those parts of the utterances for which interpolation is not phonetically feasible.

Region definition

To incorporate input-switch rules, we add meta information to each pair of utterances (A, B) to be interpolated. For utterance A and B, we define a set of regions $R(A)$ and $R(B)$ on a phone level. Every region $a \in R(A)$ and $b \in R(B)$ has to consist of at least one phone and can, at maximum, span the whole utterance. Also, a region must not necessarily consist only of consecutive phones (i.e. the region can be split up across the region) but the ordering of the phones must be preserved. The regions have to be selected so that a bijection $M : R(A) \leftrightarrow R(B)$ can be defined. This means, every utterance is split into regions and these regions are then mapped between the utterances. For every region mapping $m = (a, M(a)) \forall a \in R(A)$, a procedure to be applied during the interpolation process can be defined. For our experiments, we set the procedure for every mapping in the evaluation data to either "feature interpolation" or "feature switch". Feature interpolation is used in case of a phonological process or a pseudo-phonological process as described in Section 4.1. We use the feature switch procedure in case of an input-switch rule. Both procedures are described in Section 4.3. If each utterance forms a single region and the mapping between these two regions is associated with the feature interpolation procedure, we get the basic interpolation method as described previously. To summarize, regions define which procedure from Section 4.1 should be used for this part of the utterance.

For our experiments, we defined the mappings M and the associated procedures according to the results of our phonetic analysis as follows: From the evaluation data, extract a list of word mappings with an input-switch rule that cannot be modeled by interpolation. If such a word mapping occurs in a sentence, compare the prefix and suffix of the words on a phonological level. If the phone symbols are the same, a region for feature interpolation can be formed. This is useful as the acoustic realization of the phonetic symbols will still differ slightly for all dialects. The remaining, differing phones then form a region which will use the feature switch procedure. Finally, merge regions next to each other that have the same procedure assigned to

[2] Co-occurence requirements refer to the fact that within an utterance, it is not allowed to arbitrarily mix standard and dialect forms [98].

them. A more sophisticated algorithm could involve DTW on the phonological level, combined with a list of phone level mappings that should be realized using a feature switch procedure. For mapping on the word level, machine translation methods [71] could be employed.

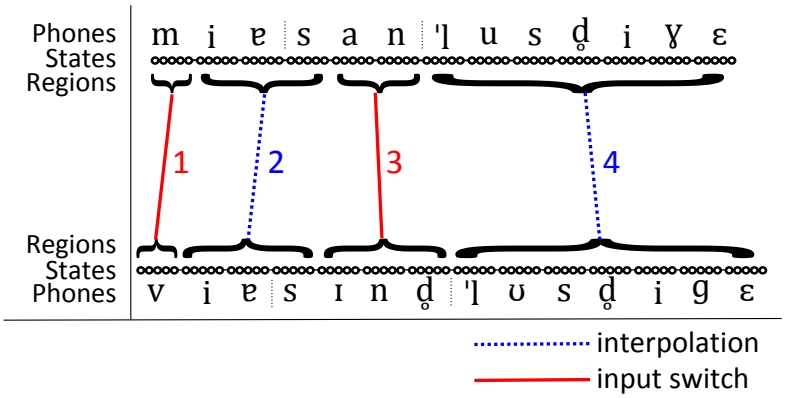

Figure 4.4: Example region definition of the sequence “Wir sind lustige" (engl. “We are funny”).

An example for a defined region mapping and procedures can be seen in Figure 4.4. As described previously, /viɐ̯/ and /miɐ̯/ are connected by an input-switch rule. As the suffix /iɐ/ is the same in both utterances, it can form the feature interpolation region 2. /v/ and /m/, on the other hand, form the feature switch region 1. It can also be seen that region 2 is merged with the /s/ from the following word, because this is again a word beginning with the same phones and also uses feature interpolation.

Region procedures

The function I_{linear} applies DTW on the HSMM states and linearly interpolates the associated features as well as the durations as described in Section 4.3 and returns the newly generated states.

The function I_{switch} does a feature switch for given HMM states and is shown by Equation 4.9.

$$I_{switch}(a, b, \alpha) = \begin{cases} a & \alpha \leq 0.5 \\ b & \alpha > 0.5 \end{cases} \tag{4.9}$$

Region-based interpolation

The inputs for the extended, region-based interpolation algorithm are: phonetic transcriptions of two utterances, the two associated voice models, the interpolation parameter α and the region information including region mapping and region procedure for each mapping. Algorithm 2 presents the subsequent steps. First, for each region, the indices of the HSMM states representing each phone in this region are retrieved using the voice model decision trees. $value(index)$

is then used to access the actual HSMM state model for the given index. is_switch and $is\_interpolation$ are functions that return true if the supplied region has to use the feature switch or feature interpolation procedure respectively. In case of feature switch, the function I_{switch} is used to retrieve the resulting indices for the current region and is then, together with the associated values, appended to the $results$ list. If the region has to be interpolated, DTW is applied, which returns a list of tuples of indices, representing the optimal warping path (as described in section 4.3). All HSMM states along this warping path are then interpolated using I_{linear} and the resulting, new HSMM states are returned. These are, together with the associated indices for each utterance, also appended to the $results$ list. Finally, $results$ is sorted according to a function $ordering$ (described in 4.7) which defines if the states should be in order of utterance A or utterance B.

Algorithm 2 Region-based interpolation algorithm.

1: $results \leftarrow list()$ {result list}
2: **for all** $r \in R(A)$ **do**
3: $idx_A \leftarrow$ indices of HMM states in r
4: $idx_B \leftarrow$ indices of HMM states in $M(r)$
5: **if** is_switch(r) **then**
6: $values = I_{switch}(value(idx_A), value(idx_B), \alpha)$
7: append $(idx_A, idx_B, values)$ to $results$
8: **else if** is_interpolation(r) **then**
9: $dtwpath \leftarrow DTW(idx_A, idx_B)$
10: $values = I_{linear}(dtwpath, \alpha)$ {dtwpath column 0 is indices for first utterance, column 1 for second utterance}
11: append $(dtwpath[0], dtwpath[1], values)$ to $results$
12: **end if**
13: sort $results$ by $results[I_{order}()]$
14: **end for**

Duration and order modeling

In the extended method, duration modeling for feature interpolation is as described in Section 4.4. For feature switching, this method is not necessary, the number of states and their durations can just be taken from one of the two involved utterances depending on α.

This extended interpolation method can also be used for utterances with a different syntactic structure. Consider the example of a translation from Standard German into dialect syntax that can be seen in Figure 4.5: “Ich ging weg” (engl. “I left”) $\leftrightarrow$ “Ich bin weg gegangen” (engl. “I have left”).

In this case, the region definition is a bit different because there are no neighbored regions that could be merged. The region-based interpolation algorithm would then again apply the associated procedures (feature interpolation or feature switch) on each mapped region. Feature interpolation creates states for each mapping along the DTW path. Feature switch uses states, durations and features from either utterance A or from B, depending on α. The states of the

regions are then concatenated in the order of A or B, also depending on α. So if $\alpha \leq 0.5$, the ordering of A is used, else ordering of B. For the example shown in Figure 4.5 this means that for $\alpha \leq 0.5$ the ordering of "Ich ging weg" is used, for $\alpha > 0.5$ the ordering of "Ich bin weg gegangen". The evaluations presented in this study did not include utterances which required reordering.

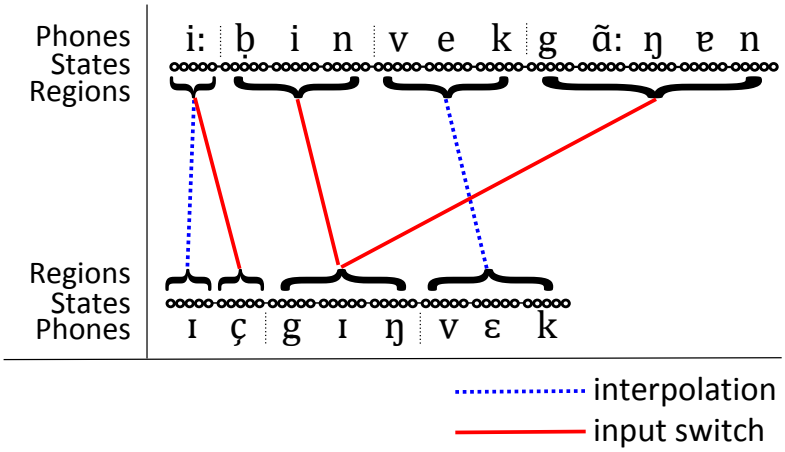

Figure 4.5: Example region definition of the translated sentence "Ich ging weg" ↔ "Ich bin weg gegangen".

4.8 Evaluation

We conducted two subjective listening tests to evaluate the presented methods. In the first test we compared the intelligibility and the applicability of interpolation for the expanded and the unexpanded method. Word-error-rate experiments were carried out in order to evaluate whether the interpolated samples have a higher word-error-rate than the uninterpolated samples, but were not meant to measure the inherent word-error-rates of dialects. The second experiment was used to assess the effect of the integration of input-switch rules in the interpolation process.

Table 4.3 shows sample utterances used in the evaluation. We interpolated between Regional Standard Austrian German (RSAG) and one dialect variety (IVG, GOI, VD) at a time. In total, we use 6 different utterances per variety, which were manually defined, transcribed and analyzed to eliminate potential errors of the text analysis module.

There are significant phonetic and lexical differences between RSAG and the respective dialect (IVG, GOI, VD). These differences lead to different numbers of phones and different numbers of lexical items between RSAG and dialects (e.g. RSAG "d̥iː ˈtaːgɛ - **die Tage**" vs. GOI "d̥ːɔːk - **die Tage**"[3] occurred in the evaluated samples).

[3] In GOI, the definite article "die" is reduced to [d̥] and subsequently merged with the initial [d̥] of "Tage".

SAG orth.	**Schnee liegt im Garten.**
RSAG phon.	ˈʃneː liːkt ɪm ˈgɑːd̥n
IVG phon.	ˈʃnea liːd̥ iŋ ˈgɔχd̥ɛ
SAG orth.	**Morgen kommt meine Schwester.**
RSAG phon.	ˈmɔɐ̯gŋ kɔmt maɐ̯nɛ ˈʃvɛ̯estɐ
GOI phon.	ˈmɔ rɪŋ kimd̥ mãẽ̯ ˈʃvɛ̯esd̥a
SAG orth.	**Wir sind lustige Leute.**
RSAG phon.	viɐ sɪnd̥ ˈlʊsd̥igɛ ˈlɒɐ̯tɛ
VD phon.	miɐ san ˈlusd̥iɣɛ ˈlæːtʰ

Table 4.3: Sample sentences which were interpolated between Regional Standard Austrian German (RSAG), Innervillgraten dialect (IVG), Bad Goisern dialect (GOI) and Viennese dialect (VD).

Expanded and unexpanded method

In this subjective evaluation we interpolated from RSAG to IVG, RSAG to GOI and RSAG to VD. We used 6 utterances and one speaker per dialect and interpolated each utterance using 6 different values for α (0.0, 0.2, 0.4, 0.6, 0.8, 1.0) using both expanded and unexpanded states. This setup produced 216 unique sound samples for the subjective listening test.

We had 12 native Austrian listeners who took part in the evaluation. While the listeners were mainly acquainted with VD, 2 listeners were IVG speakers and another 2 listeners grew up near regions where GOI is spoken. The evaluation was split into two parts. The first part was an intelligibility test where the listeners had to write the perceived content of audio samples into a text field. The samples were randomly assigned to the listeners under the constraint that each utterance was heard only once by each listener in order not to bias the evaluation. In the second part, the listeners had to score randomly assigned audio samples according to rate the dialect. The 6 possible score levels were: "Clearly Standard (1)", "Standard (2)", "Rather Standard (3)", "Rather dialect (4)", "Dialect (5)", "Clearly dialect (6)".

Figure 4.6 shows the scores for degree of dialect. It can also be seen that the subjective scores strongly changed from standard to dialect from $\alpha = 0.4$ to $\alpha = 0.6$. This seems like a natural boundary for a switch from "rather standard" to "rather dialect" and is actually reflected in the evaluation data. The scores are not significantly different although the unexpanded method shows a slightly higher variation in its mean score.

The word-error-rate results of the intelligibility test can be seen in Table 4.4. The unexpanded method has a lower error in all cases except for $\alpha = 0.6$. The word-error-rate of the intermediate variants was not significantly higher ($p > 0.4$) than the full dialectal case ($\alpha = 1.0$) in the Matched Pairs Sentence-Segment Word Error test [33]. Actually, it was significantly lower ($p < 0.02$) for $\alpha = 0.4$ than for $\alpha = 1.0$. This means that our interpolation methods do not produce a large number of errors, which would result in a higher word-error-rate for the intermediate variants. Although the dialects differ in their prevalence (i.e. VD is understood by many

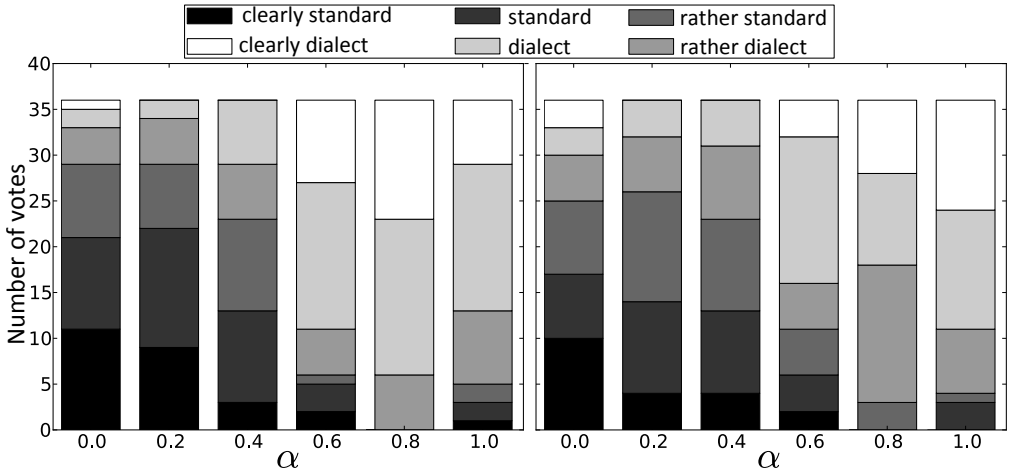

Figure 4.6: Scores for unexpanded (left) and expanded (right) method.

more Austrian inhabitants than IVG), the error rates were not significantly different (IVG: 11.2, GOI: 15.4, VD: 11.1).

	0.0	0.2	0.4	0.6	0.8	1.0	μ
unexp.	2.2	6.5	6.5	18.3	8.6	17.2	9.9
exp.	5.1	7.5	8.6	12.9	28.0	23.7	14.3

Table 4.4: Word-Error-Rates [%] per α and method.

While for $\alpha = 0.8$ we saw a large difference in word-error-rate, the overall difference between expanded and unexpanded was not significant. We chose the unexpanded method for our further experiments since it is computationally less expensive.

Interpolation with and without input-switch rules

For our second subjective evaluation we interpolated from RSAG to IVG, GOI and VD. For each of the 3 dialects, using 6 different values for α (0.0, 0.2, 0.4, 0.6, 0.8, 1.0), we interpolated 6 different utterances. Additionally, we generated the same set of samples again, this time including input-switch rules. This setup also produced 216 unique sound samples for the subjective listening test.[4]

34 native dialect listeners (born in the region where the dialect is spoken and raised in the respective dialect, 12 for VD, 11 for IVG, 11 for GOI) conducted the evaluation. The participants were carefully selected according to their dialect proficiency. Each listener had to score samples

[4]Samples at http://mtoman.neuratec.com/thesis/interpolation/

for her/his native dialect only. In this evaluation only native dialect listeners participated, since listeners had to answer questions that require a strong knowledge of the respective dialect.

The evaluation consisted of three parts: an intelligibility test, degree of dialect assignment, and a standard/dialect acceptance rating. For the intelligibility test, listeners had to write the perceived content of audio samples into a text field. The samples were randomly assigned to the listeners under the constraint that each utterance was heard only once by each listener. In the second part, each listener had to score all 72 audio samples of his/her dialect (in random order) according to degree of dialect. Since in this second evaluation only native dialect listeners participated, we allowed a more fine-grained control for the degree of dialect using a slider offered by the evaluation user interface. The two ends of the slider were named "standard" and "dialect" respectively, using the name of the actual dialect of the sample. In the third part, listeners had to answer if they accept the speaker of the sample as

- "speaking standard language and grown up with it",
- "speaking standard language which was acquired later in life",
- "speaking dialect and grown up with it",
- "speaking dialect which was acquired later in life",
- or "speaks neither standard nor dialect".

Again we used the name of the actual dialect in these questions, so e.g. "speaking Viennese dialect, grown up with it". In this way we wanted to evaluate the acceptability of our generated varieties.

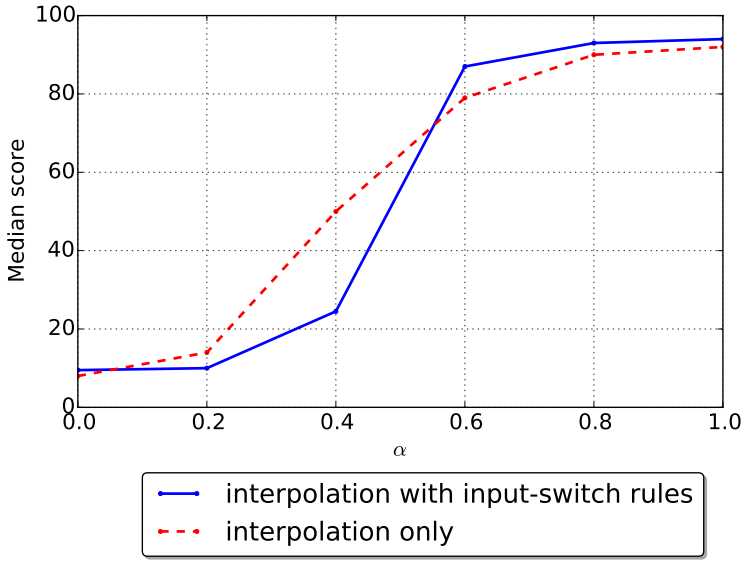

Figure 4.7: Median scores for degree of dialect.

Figure 4.7 shows the median scores for degree of dialect. The largest increase in subjective score again occurred from $\alpha = 0.4$ to $\alpha = 0.6$. As expected, the extended method that handles input-switch rules exhibits a steeper degree change here. For both methods, the subjective rating of the listeners roughly reflects the actual used value for α. This suggests that (linear) interpolation is a reasonable approach for generating in-between variants.

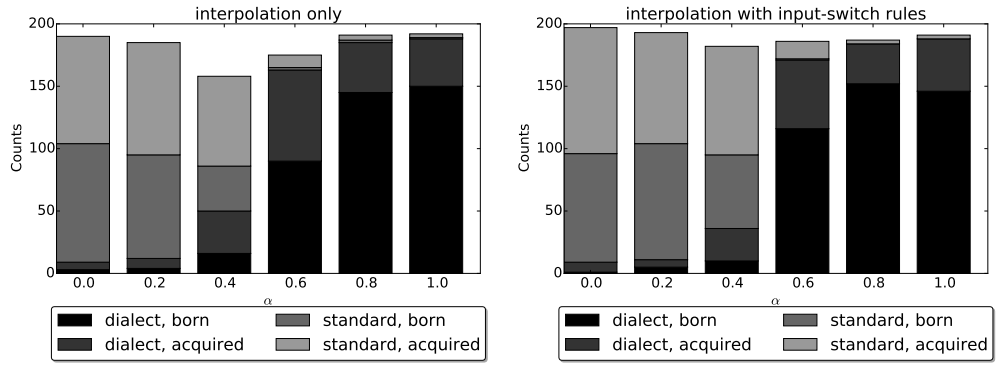

Figure 4.8: Speaker category choices.

Figure 4.8 shows the results of the standard/dialect acceptance test for interpolation without and interpolation with input-switch rules as stacked bar plots respectively. Again it can be seen that a higher α also results in the listeners perceiving the speaker of the samples as being an increasingly more authentic dialect speaker.

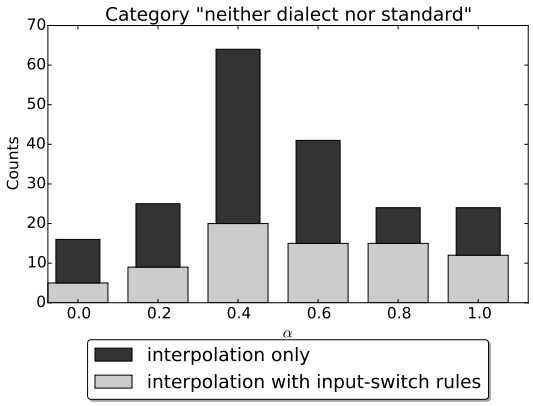

Figure 4.9: Category counts for “speaks neither standard nor dialect”.

Figure 4.9 shows the counts of the user choices of “speaks neither standard nor dialect”. Here it can be seen that using the method incorporating input-switch rules yields less votes for this category, especially for the strongly interpolated variants at $\alpha = 0.4$ and $\alpha = 0.6$.

Table 4.5 shows the Word-Error-Rates from the intelligibility part of the evaluation for interpolation only, interpolation with input-switch rules and for the different dialects. It can be seen that there are no significant differences for the different dialects and also between interpo-

	ipol.	ipol.+switch	IVG	GOI	VD
err.	14.6	13.1	13.9	14.2	13.2

Table 4.5: Word-Error-Rates [%] for "interpolation only" and "interpolation with switching rules" for the three dialects.

lation and interpolation with input-switch rules. This shows that interpolation by itself does not produce unintelligible intermediate variants and adding input-switch rules does not significantly increase intelligibility but increases the acceptance of the samples as an authentic dialect.

4.9 Discussion

We have presented an unsupervised interpolation method to generate in-between variants of language varieties. It employs DTW to find mappings of HSMM states for state-level interpolation. Two methods were introduced to handle state durations - either expand each state with duration N to N states with duration 1 (i.e. each state generates 1 feature frame), or continue with the unexpanded states. The results of our experiments suggest that linear interpolation of HSMM-states mapped by DTW is reasonable. Employing DTW on unexpanded or expanded state sequences showed no significant difference, so the computationally less expensive unexpanded method should be preferred. We also presented a method to interpolate state durations for one-to-many mappings.

Based on synthesized samples using these methods, we performed a phonetic analysis to identify utterance sections for which interpolation is not phonetically feasible. The extended method presented here introduces region definitions and region mappings for utterances. The features of the HSMM states of these regions are then subjected to either feature interpolation or feature switch as defined by the results from this analysis.

As expected, this introduces a sudden change in dialect degree perception from values below to above $\alpha = 0.5$, but at the same time decreases the generation of speech that the listeners rated as "neither standard nor dialect". Consequently, for producing correct in-between variants, including input-switch rules is beneficial. Still, even without treating input-switch rules, word error rate did not significantly decrease for highly interpolated samples (α between 0.4 and 0.6) as compared to the pure dialectal samples ($\alpha = 1.0$). Thus, even if no phonetic knowledge about these rules exist, the presented interpolation method still produces intelligible speech.

CHAPTER 5

Acoustic modeling of language varieties

Average voice modeling in combination with speaker adaptation allows to generate voice models of reasonable quality with a small training set [133], but it usually assumes the same language variety for all averaged speakers. The challenge is to adjust the model to cope with different language varieties and therefore different phone sets. Using such models, cross-variety speaker adaptation can be achieved by estimating transformations between average, variety average and speaker-specific models. The context of language varieties adds subtleties to the speaker adaptation approach that can be exploited by using overlapping phonetic information.

In [81] it was shown how adaptive dialect modeling methods can be applied to the modeling of two different varieties, namely standard Austrian German and Viennese dialect. Here we show advanced adaptive modeling methods for varieties and evaluate these methods with three Austrian German varieties: SAG, IVG and GOI.

Parts of this chapter have been previously published in [117].

5.1 Problem definition

This chapter treats Problem 2, defined as:

- **Problem 2**: Given speech data of multiple speakers $S_1...S_m$ in different varieties $V_1...V_n$ of the same language, generate synthetic speech for a specific speaker S_i with higher quality than when having only data in the variety of speaker S_i.

Therefore a system to solve this problem has the following **input**:

- Recorded speech of multiple speakers $S_1...S_m$ in different varieties $V_1...V_n$.
- Time-aligned phonetic transcription for all recordings.
- Set of decision tree questions for each variety $V_1...V_n$.
- Identification i of target speaker S_i for whom a voice model should be generated.

The system should produce the following **output**:

- Voice model of target speaker S_i that can be used to synthesize arbitrary utterances.
- Optional: Average voice model for each language variety $V_1 ... V_n$.

In this chapter we investigate Hypothesis 2:

- **Hypothesis 2**: Adaptive modeling techniques for dialectal data produce speech output of better quality than speaker dependent voice modeling techniques.

5.2 Modeling approaches

			Data Dependency	
Name	Target	# utt.	Speaker	Dialect
SD-DD (AT)	AT	198	✓	✓
SD-DD (IVG)	IVG	618	✓	✓
SD-DD (GOI)	GOI	622	✓	✓
SI-DD (AT)	AT	1790	×	✓
SI-DD (IVG)	IVG	1236	×	✓
SI-DD (GOI)	GOI	1244	×	✓
SI-SN	AT/IVG/GOI	4270	×	×
SI-SDN	AT/IVG/GOI	4270	×	×
SI-SDNC	AT/IVG/GOI	4270	×	×
DHN	AT/IVG/GOI	4270	×	×

Table 5.1: Dialect modeling approaches.

Table 5.1 defines multiple modeling approaches that we investigated and the number of utterances from our corpus (see Chapter 3) that are used for each approach.

Speaker-dependent, dialect-dependent (SD-DD)

A voice model is trained using only the data of the target speaker in her dialect.

Speaker-Independent, dialect-dependent (SI-DD)

An average voice model of multiple speakers in the dialect of the target speaker is trained. The target speaker is then adapted from this average model. So for each dialect, a single average voice model is trained and the target speaker is then adapted from the average model with of the according dialect.

Speaker-Normalization (SI-SN)

A single average model of all dialects and speakers is trained. Every target speaker is then adapted from this model.

Dialect-Hierarchical-Normalization (SI-DHN)

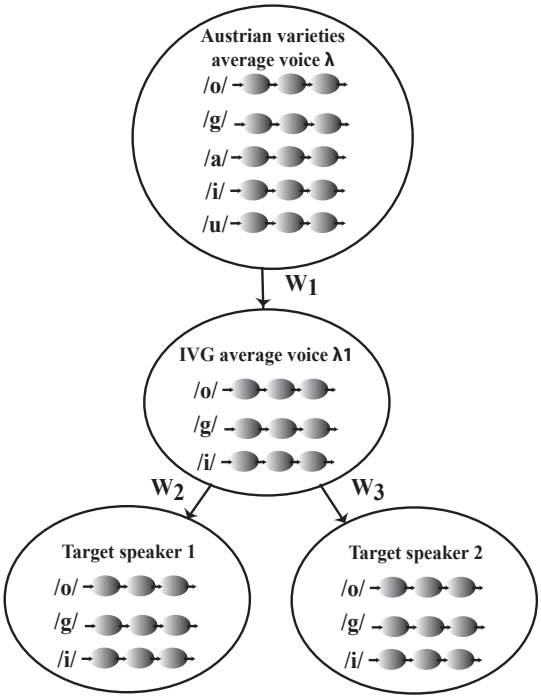

Figure 5.1: DHN: Dialect-hierarchical normalization.

A general dialect-independent voice model is trained first using data from all dialects and speakers as in SI-SN. From this average model, dialect-dependent voice models for each dialect are adapted. Finally, speaker-specific voice models are adapted from these. In the example in Figure 5.1 a dialect-dependent model for the IVG dialect is adapted from the full average voice model. IVG speaker-dependent models are then adapted from it.

Speaker-Dialect-Normalization (SI-SDN)

Speakers with data from multiple dialects are split into two subsets of speech data uttered by two different pseudo-speakers. These pseudo-speakers are treated as completely different speakers by the average voice training. In the example in Figure 5.2, AT and IVG recordings of speaker 1 exist. Speaker 1 will then be treated as two different speakers, one AT and one IVG speaker. Every target speaker is then adapted from this model.

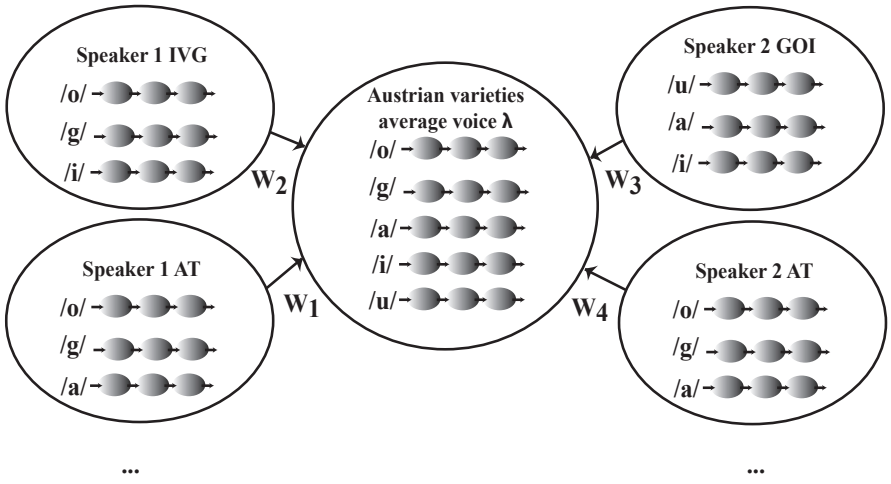

Figure 5.2: SDN: Speaker-dialect-normalization.

Speaker-Dialect-Normalization with dialect Clustering (SI-SDNC)

In addition to SI-SDN, an identification tag for each dialect is added to all training data labels and included in the questions for the decision-tree-based clustering. This enables branching by dialect in the generation of the decision tree.

5.3 Dialect clustering

In the clustering of dialects, new questions that identify the dialect of an utterance (Is_ivg, Is_goi, Is_at) are added to the set of questions used for the decision-tree-based clustering with Minimum Description Length (MDL) based node-splitting [99]. Dialect is treated as a feature in the context labels together with the other phonetic and linguistic features and it is therefore also included in the resulting acoustic model. Note that a decision tree was constructed independently for each combination of state index and acoustic parameter (mel-cepstrum,$\log F_0$, band-limited aperiodicity) and duration. The same idea has been reported for multi-accented English average voice models [136].

In the resulting decision trees we observed that the first dialect questions are used close to the roots of the decision trees. Figure 5.3 shows this part of the constructed decision tree for the mel-cepstral parameters of the third (middle) state and Figure 5.4 the corresponding duration parameter clustering tree. These are the top-most occurrences of variety questions in the trees and they appear on level 4 in the state 3 mel-cepstral tree and on level 5 in the duration tree.

In this example, “Is_ivg” means “Is the current utterance in Innervillgraten dialect?” and “Is_goi” means “Is the current utterance in Bad Goisern dialect?”. This means that after these questions, separate PDFs are produced for the different dialects. We also observed the labels which have been used to train each PDF: In SDN only 928 PDFs where estimated using data from a single variety and 1620 PDFs using data from more than one variety. For SDNC, 2431

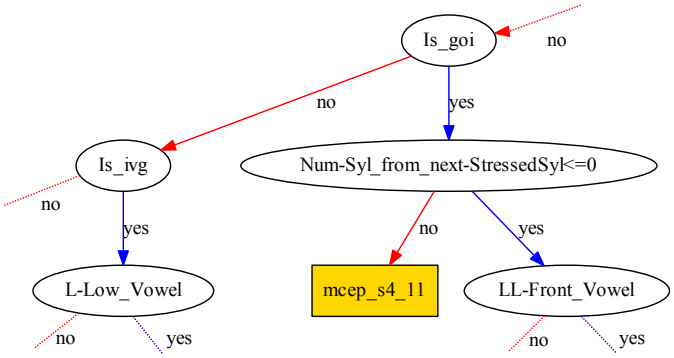

Figure 5.3: First occurrence of variety identity question in state 3 mel-cepstral decision tree.

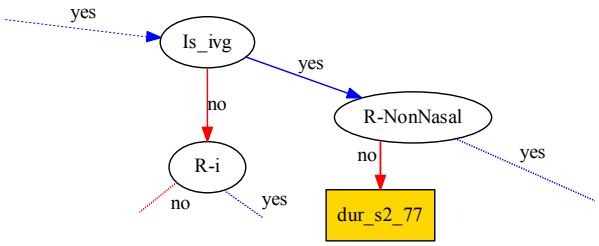

Figure 5.4: First occurrence of variety identity question in duration decision tree.

			# of occurrences		
Feature	State 1	State 2	State 3	State 4	State 5
mel-cepstral	76	139	53	54	67
log F0	28	62	70	44	33
bndap	23	24	37	28	29
duration			70		

Table 5.2: Occurrences of variety identity questions in decision trees.

PDFs were estimated using single variety data and 322 PDFs using data from multiple varieties. This also shows the strong effect of the variety questions on the clustering process. Overall occurrences of variety questions in mel-cepstral, logF0, bndap and duration decision trees can be seen in Table 5.2. It can be seen that variety questions are used in all states and therefore have a considerable effect on the produced voice model. If this effect increases the quality of the produced speech is evaluated in the next section.

Compared methods			wins	ties	sig.
DHN	:	recorded	1 : 49	0	$*$
DHN	:	SD-DD	6 : 26	18	$*$
DHN	:	SI-SDN	7 : 26	17	$*$
DHN	:	SI-SDNC	5 : 34	11	$*$
DHN	:	SI-DD	5 : 34	11	$*$
DHN	:	SI-SN	7 : 27	16	$*$
recorded	:	SD-DD	50 : 0	0	$*$
recorded	:	SI-SDN	49 : 0	1	$*$
recorded	:	SI-SDNC	48 : 0	2	$*$
recorded	:	SI-DD	46 : 3	1	$*$
recorded	:	SI-SN	49 : 0	1	$*$
SD-DD	:	SI-SDN	10 : 15	25	
SD-DD	:	SI-SDNC	10 : 15	25	
SD-DD	:	SI-DD	10 : 19	21	
SD-DD	:	SI-SN	19 : 13	18	
SI-SDN	:	SI-SDNC	14 : 8	28	
SI-SDN	:	SI-DD	12 : 15	23	
SI-SDN	:	SI-SN	16 : 4	30	$(*)$
SI-SDNC	:	SI-DD	13 : 15	22	
SI-SDNC	:	SI-SN	18 : 15	17	
SI-DD	:	SI-SN	10 : 14	26	

Table 5.3: Scores for subjective pair-wise comparison to reference sample.

5.4 Evaluation

To assess the quality of the synthetic voices resulting from the different modeling approaches described in Section 5.2, we carried out a subjective evaluation with 21 test listeners [1]. For this evaluation we only trained models with the data of the male speakers, to not mix gender in average voices. So we used four dialect speakers (two IVG and two GOI speakers) from who we have dialect and standard data available and 1 standard-only speaker. We applied model adaptation with AT, IVG, and GOI data to all models, therefore we had 30 voices in total, where 25 were adapted voices and five were speaker- and dialect-dependent voices. For each training set we held out 10 test utterances to allow comparison with the recordings and synthesized each of them using all presented methods for each of speaker. Comparing any two models for each (speaker, utterance)-combination gives rise to 1050 comparisons in total, which we distributed among our 21 listeners such that each listener heard each (speaker, utterance)-combination once and each method-pair two to three times.

For each of their 50 comparisons, the listeners heard a recorded reference sample and two samples from two different methods, where all three samples contained the same utterance from the same speaker. After listening to each of the three sound files as many times as they liked,

[1] Synthesis samples on http://mtoman.neuratec.com/thesis/modeling/

they were asked to tell which of the two samples they felt to be more similar to the reference sample. *Recorded* was also added as a method, i.e., in some comparisons the reference sample and one of the samples in question actually contained the same (recorded) signal. There was also a "tied" option (both samples equally similar to the reference).

The results are given in Table 5.3, where we have counted the number of "won" comparisons and the number of "ties" for each method pair. In the last column, the symbol "$*$" indicates statistical significance of the preference scores according to Bonferroni-corrected Pearson's χ^2-tests of independence with $p < 0.001$. Even with a relaxed significance threshold of $p < 0.05$, only one additional significance occurred (indicated by "$(*)$", but due to the large number of ties in this case (30), we do not consider this a meaningful difference between the two methods SI-SDN and SI-SN.

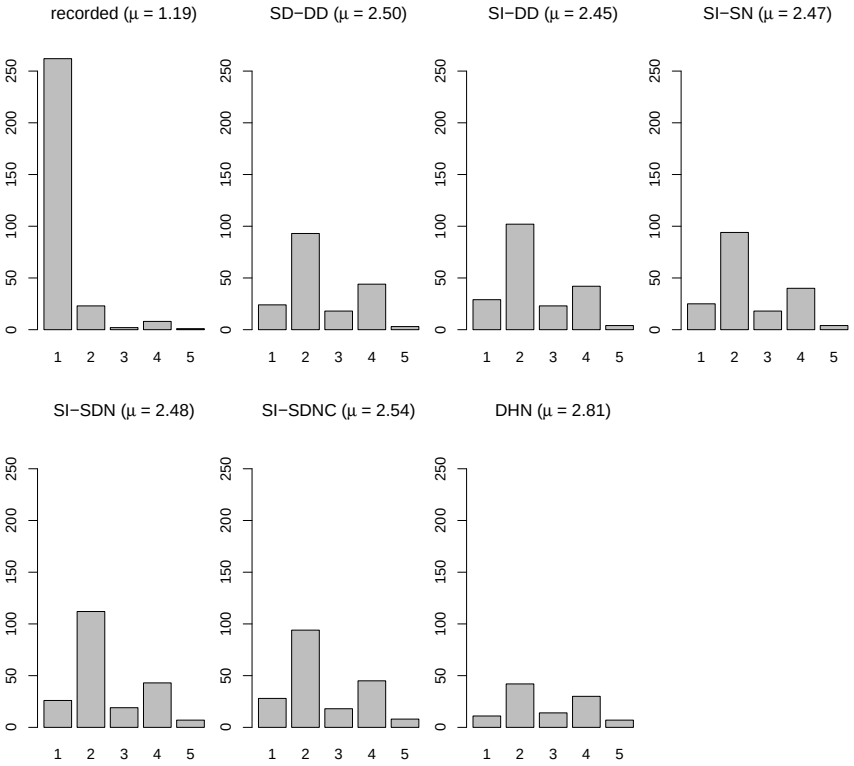

Figure 5.5: Frequencies of similarity votes for each of the seven methods to the recorded reference sample. 1 means very similar, 5 means very different.

Additionally, we asked the listeners to specify the degree of similarity for the "winning" method (or both methods, in case of a tie), by choosing one of the five options "very similar", "similar", "no opinion", "different" and "very different". The results are given in Figure 5.5 as frequency bar plots, where 1 means "very similar" and 5 means "very different". It can again be seen that there are no significant differences between the methods except for DHN giving worse results. The degree of similarity most selected was "similar" for all methods which suggests a satisfying overall quality.

5.5 Analysis

In the subjective listening test we found no significant increase in quality for any of the evaluated methods (with a significant decrease in quality for DHN). One reason for this might be the small number of speakers that was available, although for SI-SDN the average voice model encompassed 9 pseudo-speakers. Also, improvements with adaptive modeling for Austrian German and Viennese were reported [81] with similar-sized data sets. In the data set of [81], the phone set overlap between the varieties was larger, as between Austrian German and Viennese there was 77% overlap (i.e. 77% of the phones exist in both language varieties). In the data set used here, the phone set overlap was only 26% for AT $\cap$ IVG $\cap$ GOI (38 phones) in the full average voice spanning all varieties. Considering the language variety pairs, it was 33% for AT $\cap$ IVG and AT $\cap$ GOI (39 phones) and 59% for GOI $\cap$ IVG (66 phones).

5.6 Discussion

In this chapter we have investigated different adaptive modeling approaches for multi-variety modeling. We have shown that both speaker-dependent and adaptive approaches are able to achieve reasonable similarity between synthetic and recorded speech, with the exception of dialect-hierarchical training (DHN), which performed worse than all other methods (Table 5.3).

We identified one possible reason for the discrepancy of our findings with previous studies [81] as the small phone set overlap in our experiment. This suggests that clustering speech data prior to training and adaptation could be beneficial. For example, average voice models could be trained from more similar language varieties. This would of course require a much larger data set of language varieties and speakers. The small phonetic overlap of 33% shows that the distances between dialects and standard can be very high. So another possible investigation track would be to train average voice models from only dialectal data and omitting the standard variety data in training. It remains an open question how much data is necessary for a given degree of overlap to benefit from the adaptive approach. Another issue to be considered is that the phonetic overlap is not only dependent on the varieties themselves but on the granularity of the annotation. Because there are no standard phone sets for many language varieties, phone sets derived by multiple annotators might be significantly different. Concerning the overall similarity of synthesized samples to original ones we saw that we can achieve a satisfying modeling of overall similarity with all modeling methods except DHN (Figure 5.5), as most samples were rated as "similar" by the listeners.

Even if we saw no significant differences between adaptive and speaker-dependent modeling with this data set, we still recommend an adaptive approach, since it has shown its advantage under other conditions [81], possibly due to the large phone set overlap in that study. The adaptive methods never decreased the quality, except for DHN, which we would not recommend using in the current form. Adaptive methods can be especially beneficial if the data sets for single speakers are very small (for example, less than 100 sentences for a single speaker). Furthermore the adaptive approach opens up additional possibilities due to the common decision tree structure. For example, this can be useful for modeling of fast speech [80] or dialect interpolation [81].

CHAPTER 6

Cross-variety transformation

Speech resources for a single speaker are usually not available for a large number of language varieties. Transformation of language varieties aims to transform the voice of a speaker to another variety. So for example, using Standard Austrian German speech data of a speaker to generate a Viennese voice model with the same voice characteristics. This is useful for acoustic modeling with limited speech resources. It can be applied in language learning scenarios, where a user can listen to her own voice in the variety that she wants to learn. It also enables systems that dynamically adjust the language variety of their (e.g. corporate) voice to match the preference of the user. The difficulty is to isolate language variety characteristics from speaker characteristic in speech data.

Parts of this chapter have previously been published in [113, 114, 116].

6.1 Problem definition

This chapter treats Problem 3, defined as:

- **Problem 3**: Given data of speaker S in language variety V_1, generate synthetic speech in language variety V_2 without having V_2 speech data from speaker S available.

A system to solve this problem requires the following **input**:

- Speech recordings of speaker S in V_1.
- Time-aligned phonetic transcription recordings of speaker S.

For learning phonetic relations between language varieties V_1 and V_2, the system requires the following additional **input**:

- Recorded speech of multiple speakers in language varieties V_1 and V_2.

- Time-aligned phonetic transcription for the recorded speech.
- Set of decision tree questions for varieties V_1 and V_2.

The system should produce the following **output**:

- Voice model of speaker S that can be used to synthesize utterances in V_2.

In this chapter we investigate Hypothesis 3:

- **Hypothesis 3**: Transformation of HSMM voice models allows to generate intelligible speech in another, identifiable language variety than the original model.

So, cross-variety transformation aims to transform the voice of a speaker in a language variety V_1 to language variety V_2, without having training data in V_2 from that speaker. The basic approach used here is to have average voice models in V_1 and V_2 and then find a mapping between the most similar HSMM state PDFs. In our framework, a single acoustic, full-context phone model consists of 5 HSMM states with 3 feature streams, resulting in 15 PDFs. Average voice models are used to eliminate or at least reduce the influence of speaker-dependent characteristics. Using the mapping, phones which do not exist in a variety are constructed from the acoustically most similar phone components from the other variety. This mapping is then applied to the data of the speaker to be transformed. The complete approach is described in detail in the following sections.

6.2 Cross-Variety adaptation

Based on the state-level transformation method by Wu et al. [128], we developed a cross-variety transformation system employing a HSMM state mapping approach (see also Chapter 2). We extend the method in two different ways - by integrating structural information and by constraining the space of possible mappings, which are both motivated by our type of data (i.e. we are transforming language varieties instead of languages). We also evaluate the applicability of this approach to accent conversion.

Given data from multiple speakers in varieties V_1 and V_2, we build average voice models [134], denoted as AVG_1 and AVG_2 respectively. The corresponding set of decision trees for each average voice model will be further denoted as DT_1 and DT_2 respectively. DT_1 and DT_2 consist of multiple trees, for each feature stream (mel-frequency cepstral coefficients, $F0$, aperiodicity and duration) and for each of the 5 HSMM states. Given speech data of a V_1-speaker S, we aim to generate a voice model in V_2 with the characteristics of speaker S.

Mapping function

As a first step, for each PDF $A \in AVG_1$, a PDF $B \in AVG_2$ which minimizes $KLD(A, B)$ is determined. We define a mapping function M from AVG_1 to AVG_2, shown in Equation 6.1.

$$M(A) = \underset{B}{\operatorname{argmin}}\ \mathrm{KLD}(A, B) \tag{6.1}$$

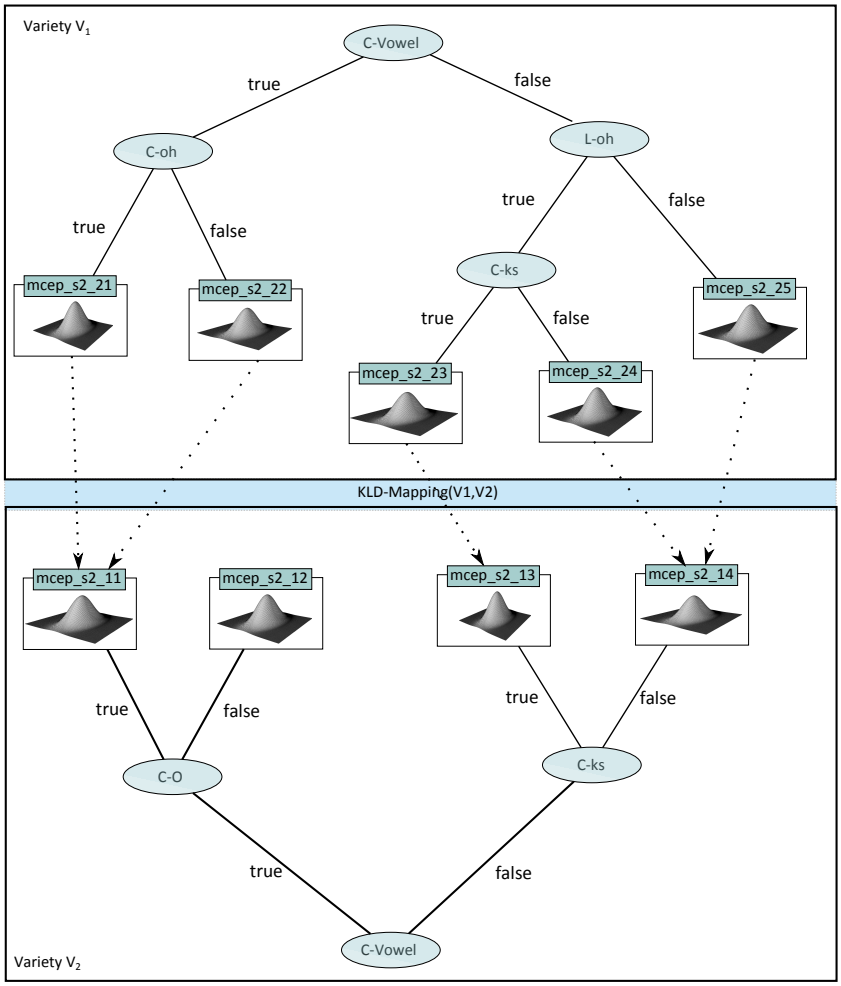

Figure 6.1: KLD mapping of PDFs clustered by decision tree.

Figure 6.1 illustrates the PDFs of AVG_1 and AVG_2, clustered by the decision trees DT_1, DT_2, and the mapping function M. For example, "mcep_s2_12" refers to the 40-dimensional PDFs with id 12 for the mel-frequency cepstral coefficients in HMM state 2. The decision tree questions used in this illustration consist of two parts. The second part is a phonetic symbol from the mapping of our phone set to ASCII symbols, for example "ks" represents x as in "wax". The first part of the question can for example be "C" for center, "L" for left or "R" for right, referring to the position of the phone in question. The mappings defined by M are represented by dotted lines. So this example shows a decision tree for the mel-cepstral features of HMM state 2. The actual number of mappings is much larger. For example, our experiments for transforming an Austrian German average voice to an IVG average voice resulted in 13,808 mappings.

Data mapping

As next step, we apply mapping function M to speaker S. To achieve this, it is necessary to classify the adaptation data labels of speaker S using the decision trees DT_1. Given a label L, $DT_1(L)$ yields one PDF for each combination of HMM state and feature stream. These PDFs are then subjected to the mapping function M: $M(DT_1(L))$. Considering the example in Figure 6.1: Label L from the adaptation data representing the center phone "oh" (IPA o:) would be classified as $mcep\_s2\_21 \in AVG_1$ for HMM state 2 and for the mel-cepstral feature stream. Mapping function M would then yield $M(mcep\_s2\_21 \in AVG_1) = mcep\_s2\_11 \in AVG_2$. Using this mapping function, each adaptation label L from speaker S is then associated with a number of PDFs in AVG_2, making the adaptation data compatible to AVG_2. This is comparable to the data mapping method described in [128].

The output of this step is a set of links from each full-context label in V_1 to multiple PDFs (for each state and stream) in AVG_2.

Regression tree generation

In the next step, all full-context labels of the adaptation data are associated with the leaf nodes of DT_2 according to $M(DT_1(L))$, as described in the previous section. Using Figure 6.1 as an example, a label L from the adaptation data (in V_1) that would be classified as "mcep_s2_22" by DT_1, will be associated with the leaf node "mcep_s2_11" in decision tree DT_2. The next step is to prune DT_2 by deleting leaf nodes which do not have at least a certain number of labels from the adaptation data associated with it. Associated labels of deleted nodes are moved to their parent node until all nodes in the tree have enough adaptation data. The resulting tree is then called a "regression tree" and the content of each node a "regression class". This is because this tree is then used to perform regular MLLR-based speaker adaptation by estimating a transformation (i.e. a linear regression) from each regression class to the actual speech data (see also the description on speaker adaptation in Chapter 2 for reference).

When implementing this algorithm, it should be noted that, by default, HTS again classifies the adaptation labels using DT_2 to find the regression classes. The decision trees DT_2 are not adequate to handle labels in variety V_1, especially if the phonetic structures of V_1 and V_2 are very different. In the worst case, all labels would be placed in the same regression class. For example, if the sets of phones of the two varieties are completely disjunct, all decision tree questions concerning phone symbols will yield false. Having all adaptation data in a single leaf node would on the other hand lead to a single, global transformation generated in the adaptation step, and the rest of the tree would be pruned. We modified HTS accordingly to deal with this issue.

6.3 Integrated structural information

We extend the previously described method to add weight to structural, phonetic information in the mapping process. The idea is to encourage mappings between full-context models with similar center phones. While full-context models can be very specific for a given context, so-called monophone models are trained from all data available for a given phone. We integrate

this general phone information into the mapping process as follows: the mapping function M (Equation 6.1) is replaced by M' as shown in Equation 6.2.

$$M'_\alpha(A) = \underset{B}{\operatorname{argmin}}(\alpha \ \text{KLD}(A, B) + (1 - \alpha) \ \text{KLD}_{mono}(A, B)) \tag{6.2}$$

KLD_{mono} describes the KLD between monophone PDFs, which are PDFs estimated from all available training data for a given phone. As opposed to "full-context labels", labels for monophone models are called "mono labels" and only consist of a phone symbol without any contextual information. With each PDF $A \in AVG_1$ we associate a set of monophone PDFs A', with A'_i being the i-th PDF in this set.

The relationship between A and A' is illustrated in Figure 6.2. Each PDF $A \in AVG_1$ has a number of associated full-context labels which were used to estimate A, i.e. the labels which are classified as A by the decision tree DT_1. In HTS, these labels are typically encoded as strings where the contextual features are separated by special characters. For example, the first 5 features are the left-left-, left-, center-, right-, right-right-phones separated by the special characters "^", "-", "+", "=". Differing separator characters are used so that simple pattern matching is possible, e.g. the center phone can always be found between "-" and "+". We retrieve these labels from the training data, extract the center phones (encircled symbols "ah" and "a" in Figure 6.2) and find the monophone PDFs for these phones. When implementing this in HTS with the EMIME average voice training scripts [49], monophone PDFs are usually estimated and stored during the average voice building process and therefore already available at that time, otherwise the monophone PDFs can be estimated from all data available for a given phone.

Given a set of monophone PDFs A' for $A \in AVG_1$ and B' for $B \in AVG_2$, we then calculate KLD_{mono} as the mean KLD of all combinations of PDFs from A' with all PDFs from B'. This is shown by Equation 6.3.

$$\text{KLD}_{mono}(A, B) = \frac{\sum_i \sum_j \text{KLD}(A'_i, B'_j)}{(|A'||B'|)} \tag{6.3}$$

As shown in Equation 6.2, the monophone KLD function $\text{KLD}_{mono}(A, B)$ is then linearly interpolated with the regular KLD function $\text{KLD}(A, B)$ using an interpolation parameter α. We then select the mapping with the lowest value $M'(A, B)$. As calculating the monophone KLD for all possible combinations of A and B is computationally expensive, we first calculate the n-best regular KLD values and calculate the monophone KLD for those combinations only. In our experiments, we set $n = 50$.

To find the optimal value of α, we trained voice models for α values from 0.5 to 1.0 in steps of 0.02. We then selected the α resulting in the highest model likelihood $P(\boldsymbol{o}|\lambda)$ (see Chapter 2). See Figure 6.3 for the search space for the data used in our evaluation. For the exact definition of the likelihood values used by HTS, see [142]. Experiments with other adaptation speakers showed that the behavior of likelihood when varying α is unpredictable with many local maxima, making efficient optimization difficult. But as average voice building and feature extraction of adaptation data has to be done only once, sampling the search space in even smaller steps for α than 0.02 is still reasonable.

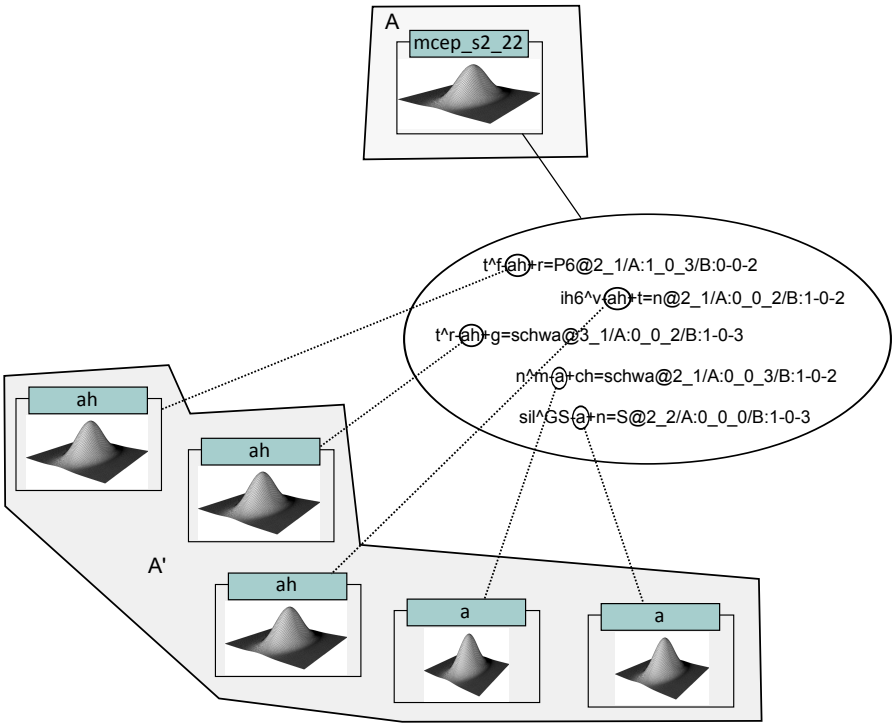

Figure 6.2: Relationship between full-context PDF A and monophone set of PDFs A'.

To assess the effect of integrating monophone KLD on the mapping function, we compared the resulting mappings for different values of α with the mappings that result from using regular KLD only (which is equivalent to using $\alpha = 1.0$). Using our SAG and IVG average voice models, we generated optimal mappings using M and M'_α for different values for α. Figure 6.4 shows the number of mappings that exist in both result sets plotted against α. As expected, the number of matching mappings increases with increasing values of α and therefore decreasing influence of monophone KLD on the mapping function. At the peak of model likelihood at $\alpha = 0.98$, only about 100 mappings are different from $\alpha = 1.0$. Further research is needed to analyze the influence of the PDFs involved in these mappings, as the importance of single PDFs to the final speech can differ tremendously. Also, PDFs that are rarely used during the synthesis process could still have an important impact when they amplify or produce a rare but highly perceptible error in the output speech.

Evaluation

This first evaluation was conducted to assess the overall feasibility of the presented cross-variety transformation method. At time of conduction, only IVG dialectal data was available. For the evaluation we used our male SAG average voice, our male IVG average voice and we used data from an SAG speaker to build an IVG voice model using the cross-variety transformation mech-

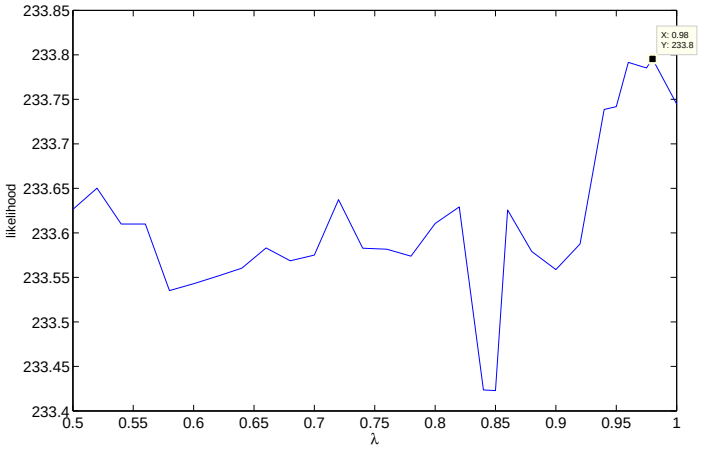

Figure 6.3: Effect of interpolation parameter α on model likelihood.

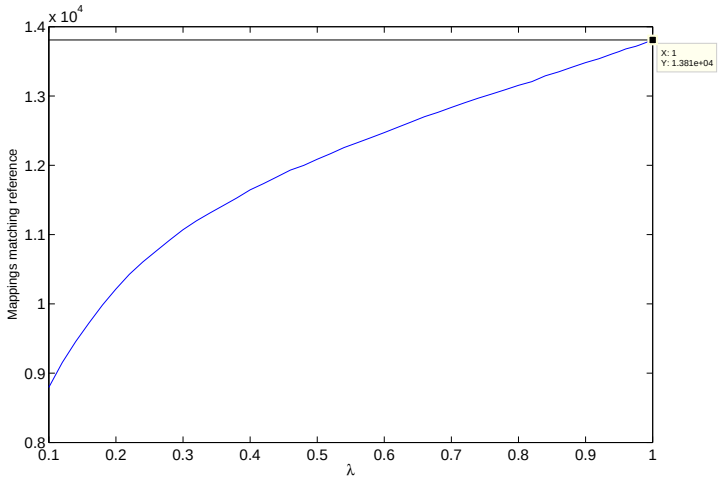

Figure 6.4: Number of mappings M that match M' when varying α.

anism described previously. From this model, we synthesized the test set of 21 utterances. As a baseline, we used the same speaker adaptation mechanism without the cross-variety extensions described previously.

We conducted a subjective listening test[1] with 5 expert listeners with a speech processing or linguistics background. Each expert listened to the results for all 21 utterances for both methods. The listeners were not given any information on the method used to synthesize each sample. Also, the positions of the samples on the evaluation interface were swapped randomly for each utterance. The listeners had to answer questions regarding language similarity, speaker similarity and overall quality. For language similarity, a sample of the same utterance from a

[1] Samples: http://mtoman.neuratec.com/thesis/transform-skld/

native IVG speaker was provided as reference. For speaker similarity, an unrelated utterance from the original recordings of the adaptation speaker was provided as reference.

The questions to be answered were:

- Which sample sounds more similar to the reference in terms of speaker identity?
- Rate the speaker similarity compared to the reference sample (1 - very different, 5 - very similar).
- Which sample sounds more similar to the reference in terms of language variety?
- Rate the language variety similarity compared to the reference sample (1 - very different, 5 - very similar).
- Which sample has the better overall speech quality?
- Rate the quality of each sample (1 - very bad quality, 5 - very good quality).

When the listener would not prefer one sample over the other, selecting "none" was allowed.

	mean			**median**			**std. deviation**	
	regular	cross-var.	Δ	regular	cross-var.	Δ	regular	cross-var.
speaker sim.	2.494	2.940	+0.446	2	3	+1	0.929	1.0517
language sim.	2.282	3.270	+0.988	2	3	+1	0.923	1.063
overall quality	2.186	3.212	+1.025	2	3	+1	0.931	0.962

Table 6.1: Sample scores for speaker and language similarity as well as overall quality.

Table 6.1 and Figure 6.5 show the results for the rating questions. It can be seen that the mean and median score was higher in all categories for cross-variety adaptation compared to regular speaker adaptation. Wilcoxon rank sum test and Welch two sample t-test both resulted in $p < 0.0001$ for both language similarity and overall quality as well as in $p < 0.005$ for speaker similarity.

The results for the questions where the listeners had to choose one sample over the other are shown in Table 6.2. It can be seen that the listeners were undecided on 25 samples when evaluating speaker similarity. This is consistent with the fact that the differences in speaker similarity scores were much lower than the other categories.

	regular	cross-var.	undecided
speaker sim.	15	65	25
language sim.	19	84	2
overall quality	15	87	3

Table 6.2: Evaluation of preferred samples.

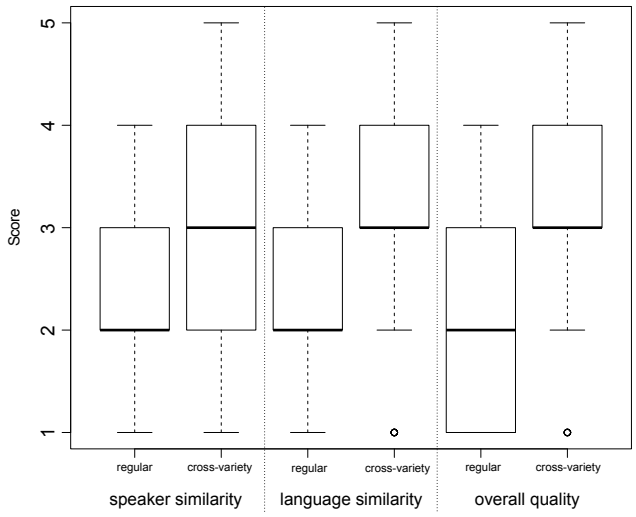

Figure 6.5: Boxplots of listening test scores.

6.4 Constrained mapping

Integration of structural, phonetic information as described above is a rather complex and computationally intensive method which is also very unpredictable and difficult to analyze. Therefore, we also investigated a constrained mapping approach. Instead of integrating acoustic information of monophone models, this approach operates on the symbolic level and regards phonetic identity when generating the optimal mappings. The basic idea is to map only between same phones, if existent in both varieties. We can, however, not fulfill this constraint directly, since the mapping itself is not defined on the phone level but only on the PDF level.

To implement the constraint, we apply the following procedure:
For each PDF $A \in AVG_1$ of variety V_1, we define $P(A)$ as the list of all center phones in all labels used to train A. We then define the representative phone $r(A)$ as the center phone that occurs most often in $P(A)$. Figure 6.6 illustrates how $r(A)$ and $P(A)$ are defined. First we find all full-context labels that have been used to train A. $P(A) = \{\text{a, a:, al}\}$ is then the set of all center phones from this list of labels and $r(A) = a$: is the center phone occurring most often (e.g. a: occurred 3 times in the example, a and al both occurred only once in the training data).

Next we constrain the mappings on the set of common phones. If the representative phone $r(A)$ occurs not only in phone set P_1 of variety V_1 (as it does per definition) but also in phone set P_2 of variety V_2, then we map from $A \in AVG_1$ to $B \in AVG_2$, if $r(A)$ is in $P(B)$.

So the Common Phone Data Mapping (CPDM) $M(A) = B$ from AVG_1 to AVG_2 has to

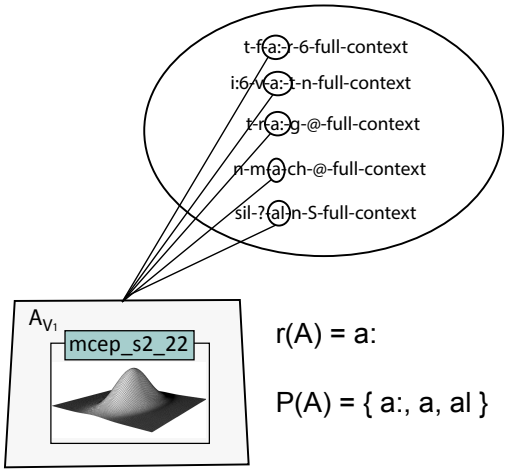

Figure 6.6: Definition of representative phone $r(A_{V_1})$ and list of all center phones $P(A_{V_1})$.

fulfill the conditional constraint given in Equation 6.4:

$$(r(A) \in P_2) \Rightarrow (r(A) \in P(B)) \tag{6.4}$$

In other words, if the representative phone $r(A)$ occurs in both varieties, we discard all potential mappings $M(A) = B$ for which $r(A)$ is not in the training data of B. For example, if the phone a: exists in variety V_1 and also in V_2, we only map from A to B if a: is also at least once in the training data of B (a: $\in P(B)$). If $r(A)$ does not occur in both varieties, we keep all potential mappings (A, B). Of all these remaining potential mappings, the mapping with the lowest KLD value is selected (as in Equation 6.1). A stronger constraint would be to require that $r(A)$ is also the most common phone in P_2, so $r(A) = r(B)$. However, we did not evaluate this approach because of cognitive limitations on the possible content for the listeners to score in the subjective listening tests. We found that few listeners were willing to participate in listening tests taking more than 30 minutes, also the quality of their assessment started to degrade at that point.

Figure 6.7 shows the percentage of (KLD-wise) 1-best and 200-best mappings between the different varieties that fulfill the constraint given by Equation 6.4. n-best means the n mappings with the lowest KLD score. To map from SAG to IVG, for example, there are 36% in the 1-best lists and 27% in the 200-best list where the mapping is between equal phones if there is an overlap.

Evaluation

We compared two methods for cross-variety transformation: *Data Mapping* (DM), which is the regular state mapping procedure, and *Common Phone Data Mapping* (CPDM), which is the constrained method presented previously. We conducted a subjective and an objective evaluation as well as an analysis of specific cases. These will be described in the following sections.

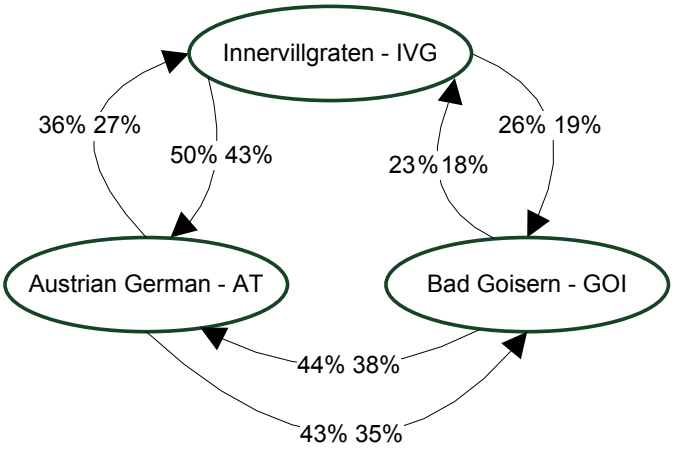

Figure 6.7: Transformations between varieties.

Compared methods	wins	ties
DM : CPDM	31 : 27	82

Table 6.3: Results of the variety similarity judgment part of the evaluation.

Subjective evaluation

We have carried out a subjective listening evaluation with 27 listeners participating, all native German speakers from different regions in Austria, including 9 listeners from our target regions (East Tyrol or Upper Austria). The subjective evaluation consisted of two parts. In the first part, we compared synthesized samples from the two methods with a reference sample. The listeners had to select the sample that they found to be more similar to the reference sample in terms of variety. The reference sample was a recording of the same sentence uttered by a different speaker of the target variety (V_2). Our assumption was that this experiment design allows that listeners who are not themselves speakers of the target variety can still judge the variety similarity. The results in Table 6.3 show that method DM was considered more similar 31 times, CPDM was considered more similar 27 times and 82 times they were considered equally similar to the reference. The difference in the number of "wins" (31 vs. 27) is not statistically significant according to a Bonferroni-corrected Pearson's χ^2-test of independence ($p > 0.58$). This and the large number of "ties" suggest that none of the two methods is superior to the other in terms of variety similarity.

Additionally, we asked the listeners to specify the degree of similarity concerning variety for the "winning" method (or both methods, in case of a tie), by choosing one of the five options "very similar", "similar", "no opinion", "different" and "very different". The results are given in Figure 6.8 as frequency bar plots, where 1 means "very similar" and 5 means "very different".

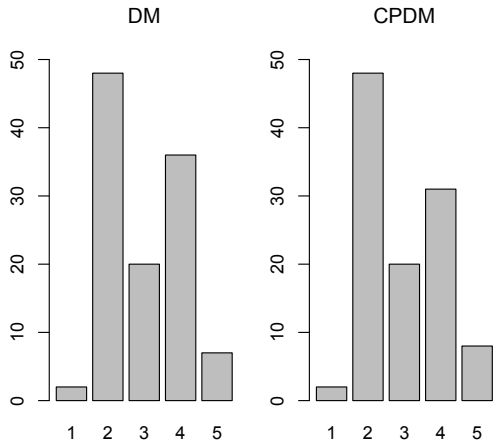

Figure 6.8: Frequencies of variety similarity votes for the two methods. 1 means "very similar" and 5 means "very different".

We can see that "similar" was the most frequently chosen option. The number of votes for "different" can be explained by the difficulty of the concept of variety similarity, which often includes a factor of authenticity. Authenticity can be affected negatively by the overall quality of synthetic speech, because errors in the synthesis can be perceived as if the speaker was a less authentic dialectal speaker.

In the second part of the evaluation, the goal was to assess speaker similarity. We presented the listeners one synthesized sample from one of the two methods and two recorded reference samples at a time. The two references were both the same utterance in a variety different from the target variety of the synthesized sample, one from the target speaker and one from a randomly selected different speaker of the same gender. The listeners were asked to decide to which of the two references the synthesized sample sounded more similar in terms of speaker identity. The results are given in Table 6.4, where for each of the two methods, the number of correct, wrong and undecided judgments are presented. For both methods, the number of correct speaker identifications is statistically significantly higher than the number of wrong speaker identifications (Bonferroni-corrected Pearson's χ^2-test of independence with $p < 0.001$).

Method	correct	wrong	undecided	sig.
DM	91	35	14	$*$
CPDM	91	28	21	$*$

Table 6.4: Results of the speaker identification part of the evaluation.

Again, the listeners were also asked to specify the degree of similarity by choosing one of the five options "very similar", "similar", "no opinion", "different" and "very different". The results are given in Figure 6.9 as frequency bar plots, where 1 means "very similar" and 5 means

"very different". It can be seen that while the number of votes for "similar" decreased for CPDM compared to DM, the number of votes for "very similar" increased.

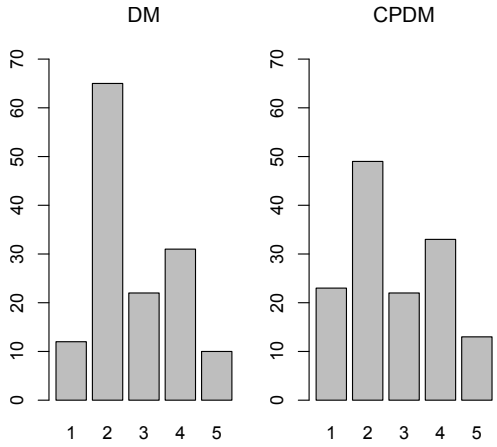

Figure 6.9: Frequencies of speaker similarity votes for the two methods. 1 means "very similar" and 5 means "very different".

Objective evaluation

We also conducted an objective evaluation by calculating mel-cepstral distortion between the trajectories resulting from the presented methods and trajectories extracted from original recordings. This was possible as we had recordings of standard as well as dialect the speakers. For the analysis we used our SAG test set consisting of 23 utterances. We transformed two IVG and two GOI speakers to SAG and calculated mel-cepstral distortion between the synthesized results and the SAG recordings of the same speaker for the same utterance. The samples were synthesized using the phone durations obtained by automatic alignment of the test recordings.

Table 6.5 shows the result of this analysis for the four speakers. It can be seen that the mean mel-cepstral distortion is lower for CPDM than for DM for the IVG speakers while it is higher for the GOI speakers. However, only the difference for speaker IVG 2 is significant according to a Bonferroni-corrected Paired t-test ($p < 4 \times 10^{-8}$). This shows that CPDM improves the model for one speaker and does not corrupt the model for the others.

We also trained speaker-dependent (SD) models (using 223 utterances) in SAG for every speaker for reference. The mean values and standard deviations for the speaker dependent models compared to the recordings can also be seen in Table 6.5. As expected, all speaker-dependent SAG models have significantly ($p < 2 \times 10^{-14}$) lower mel-cepstral distortion compared to the cross-variety transformation models. This shows that the mel-cepstral distortion metric covers aspects of speaker similarity.

	DM		CPDM		SD	
Speaker	μ	σ	μ	σ	μ	σ
IVG 1	6.33	0.43	6.29	0.36	5.42	0.31
IVG 2	7.01	0.72	6.68	0.66	5.46	0.29
GOI 1	6.85	0.43	6.86	0.44	5.90	0.47
GOI 2	7.03	0.74	7.12	0.73	6.10	0.81

Table 6.5: Objective evaluation results for all speakers and methods.

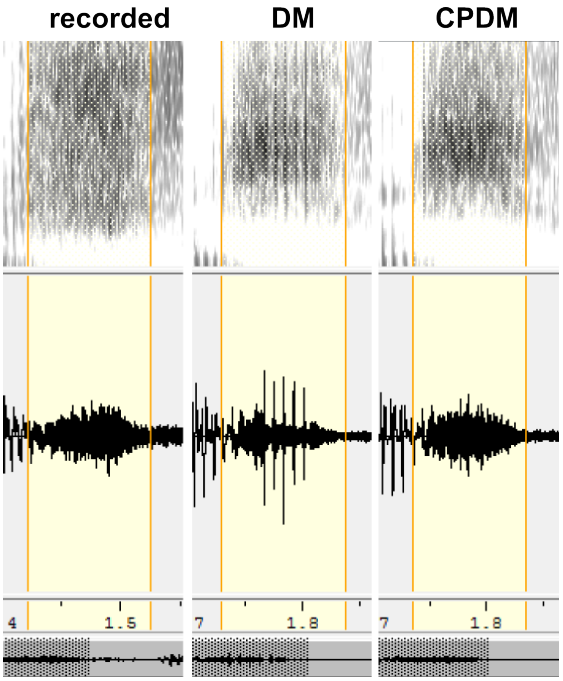

Figure 6.10: Waveforms for “s” sound as recorded and synthesized using DM and CPDM.

Analysis of specific cases

When manually inspecting the synthesized waveforms, we noticed structures that remarkably differed for the DM and CPDM methods. When listening to these parts, we could find glitches in DM that were absent in CPDM. While these glitches were quite distinct, they did not seem to be the dominant factor to influence scoring in the subjective listening test. As an example, consider Figure 6.10. The highlighted section of the waveform corresponds to the main part of the phone “s” and is presented for the recording of a GOI speaker and an SAG speaker transformed to GOI using DM and CPDM method. It can be seen that the waveform generated

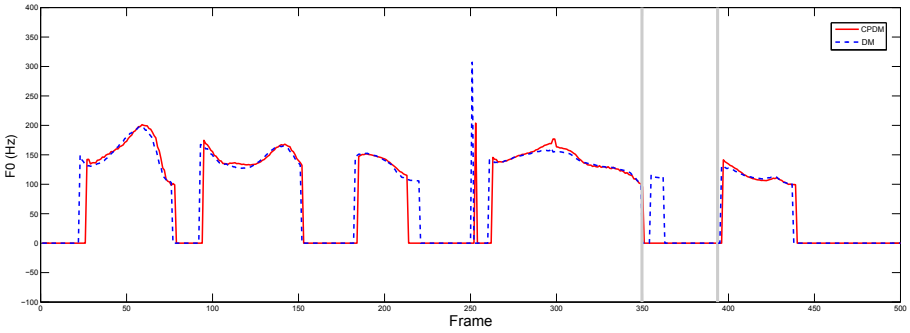

Figure 6.11: Different F0 trajectory for DM and CPDM.

by CPDM is smoother (less high-frequency spikes) and more closely resembles the waveform of the natural "s". In the DM version, the "s" has a crackling sound that is absent in the CPDM synthesis. Analyzing the different trajectories for this sample resulted in Figure 6.11. The "s"-sound is highlighted between frames 350 and 400 and it can be seen that the f0 values differ in this region. DM produces a voiced region compared to the correct, unvoiced region produced by CPDM. Reasons for this behavior remain subject of further investigation. Listening samples for some specific cases can be found on the demopage[2].

6.5 Accent conversion

We applied the methods described previously to the use case of foreign accent conversion. Here we investigate conversion from accented speech to SAG to achieve reduction of the foreign accent.

As average models of specific accents of a certain language are often not available, we apply HSMM state mapping between an accented, speaker specific voice model and a non-accented average voice model of SAG. We hypothesize that phone identity has a stronger influence on the KLD metric than speaker identity and that therefore this approach is feasible. In Section 6.4 we introduced constraints on the mappings for the overlapping phone sets of two language varieties to improve the quality of the transformed voice. As in this experiment we use the SAG phone set for both the accented and the non-accented models, these constraints are still likely to improve quality but also constrain the degrees of freedom for transforming the voice models (i.e. removing the accent) as the space of possible mappings is reduced. If the hypothesis is wrong and speaker identity has the stronger effect on the KLD measure, these constraints encourage mapping of same phones by incorporating the knowledge from the phonetic transcription instead of acoustic features. Therefore we included both the unconstrained and the constrained mapping methods in our experiments.

In [125] it was shown that listeners perform worse in speaker discrimination tasks when facing mixed conditions with natural and synthetic voices. Therefore we conclude that the baseline to evaluate accent conversion should not be natural but synthetic speech. [125] and [47] have

[2]Synthesis samples on http://mtoman.neuratec.com/thesis/transform-constrain/

shown that speaker adaptation with a set of 105 sentences is able to overrule the influence of the accent of the average voice and produce accented voice models. So in our experiments we use voice models of accent speakers adapted from a non-accented SAG average voice model (trained from 1790 utterances of 9 non-accented, male Austrian German speakers) as baseline. [52] showed that synthesized foreign accent received lower accent ratings by listeners than naturally produced accent. With our adaptive approach we assume to have similar findings concerning accent ratings. [126] also found that listeners perform worse in cross-lingual speaker discrimination tasks. This is very likely to also influence speaker discrimination tasks between accented and non-accented speech of the same speaker, as also suggested in [26]. We therefore also expect that accent conversion has an effect on the speaker discrimination performance.

Effect of mapping constraints

We investigated how many KLD mappings actually differed for the models used in this experiment when constraining the set of possible mappings. Figure 6.12 shows the number of matching mappings where the unconstrained and the constrained method selected the same mapping (i.e. the KLD-wise best mapping is not affected by the constraints), and the number of not matching mappings where both methods selected different mappings (i.e. due to the constraints, the KLD-wise best mapping was not used). In total 69% of the mappings were affected by the constraints, meaning that for the constrained method 69% of the selected mappings were not the KLD-wise best matches. This shows us that both methods are indeed very different and will produce different adapted acoustic models.

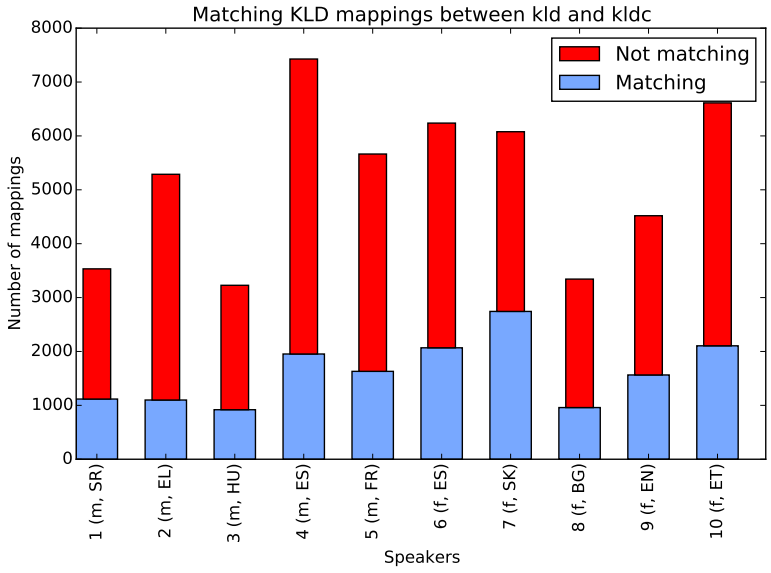

Figure 6.12: Number of mappings that differ when using constrained KLD for each speaker model.

Evaluation

For our evaluation we wanted to assess the accent degree, accent identity and speaker identity as perceived by the listeners for different methods. We used the data set of accented speakers as described in Section 3.4. For each of the 10 accent speakers all 297 utterances were used for adaptation from an average voice of 9 non-accented, male Austrian German speakers, trained from 1790 utterances.

The methods used in the evaluation were:

- accented natural speech ("rec")
- accented synthetic speech ("adapt")
- accent-reduced synthetic speech using KLD-based state mapping ("kld")
- accent-reduced synthetic speech using constrained KLD-based state mapping ("kldc")

20 listeners participated in the evaluation. They performed a listening test consisting of two parts with three different tasks. In the first part, the listeners had to identify and rate accents of 48 sound samples, in the second part they had to discriminate speakers in 120 sound sample pairs (i.e. 240 samples)[3].

Accent identity and rating

In this part of the evaluation each listener was presented 48 samples to rate. The first step for each sample was to rate the degree of the accent using a slider that ranged from "0 - no accent" to "100 - very strong accent". The second step was to try to identify the heard accent. The listeners could select an accent from a list showing the accents used in the evaluation. There was also the option to select none. If the listener moved the accent rating slider to "no accent", the accent selection disappeared. The listeners were asked to ignore the quality of the samples and focus on the accent only. The evaluation data consisted of 12 utterances spoken by 10 speakers using 4 methods, resulting in 480 unique samples. These were distributed equally to all listeners twice (i.e. 960 samples), resulting in 48 accented samples per listener. As a baseline we also included recordings and adapted synthetic speech from a non-accented speaker. These were 24 additional samples that were also distributed to all listeners twice, resulting in 2 to 3 non-accented samples for each listener.

Figure 6.13 shows the accent ratings and identity guesses for all accented speakers. It can be seen that the median accent rating for the recorded samples across all speakers was 60. The adapted voices were rated with a median rating of 49.5. While the authors found the synthetic voices to retain the characteristics of the accent quite well, the difference between recorded and adapted speech was still significant ($p < 0.007$, double-sided Mann-Whitney U test). This result agrees with the findings of [52]. "kld" and "kldc" achieved median ratings of 27.5 and 28 respectively with the differences to "adapt" and to "rec" being highly significant ($p \ll 0.001$, double-sided Mann-Whitney U test). The difference between "kld" and "kldc" was not significant.

[3] Examples at http://mtoman.neuratec.com/thesis/transform-accent/

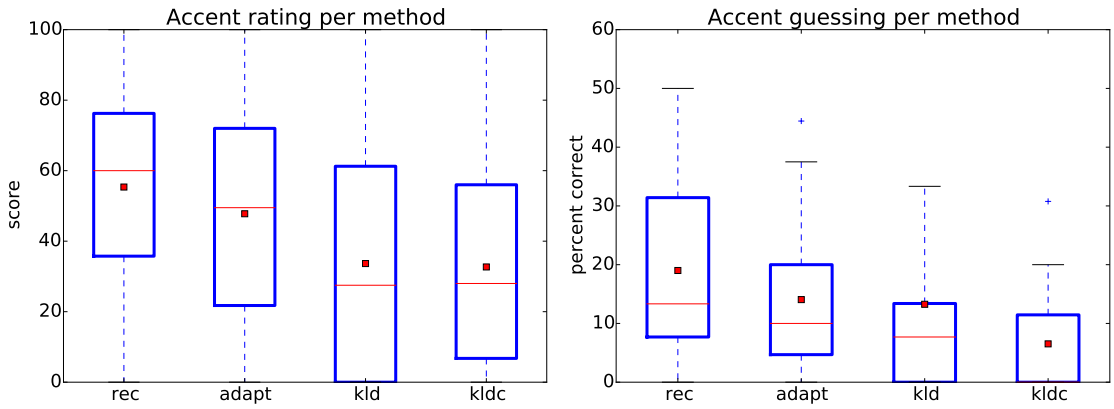

Figure 6.13: Accent ratings (left) and guesses (right) per method.

The results of the accent identity guessing task in Figure 6.13 are based on the samples for which the listeners selected an accent rating > 0 and therefore had the option to select an accent. Also, the data from the non-accented speaker was excluded. As expected, accents in "rec" samples were recognized correctly most often but still only with a median correctness of 13.3%. The speaker identified most often was the French male speaker, who was identified correctly with median 16% and mean 22%. Given that the listeners could select from 9 different accents, the results were actually not significantly higher than those which could have been expected from random guessing.

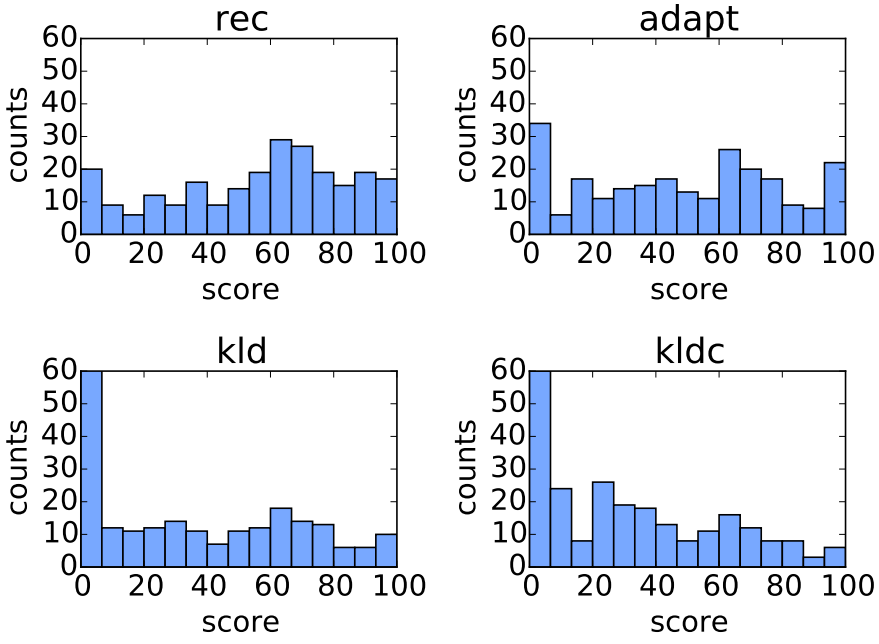

Figure 6.14: Accent rating histograms per method.

Figure 6.14 presents histograms of the accent ratings for all methods. It can be seen that the histograms for the accent reduced samples have high counts of ratings close to zero, which was the "no accent" setting for the rating slider in the evaluation interface.

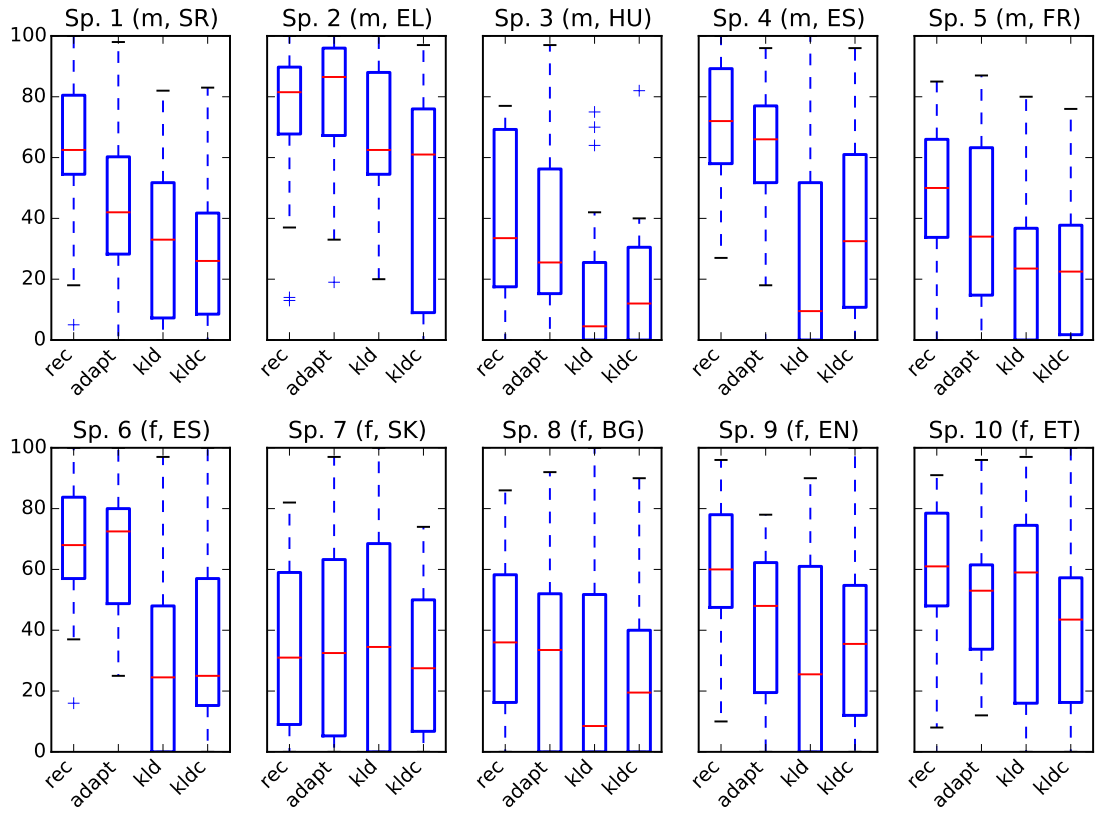

Figure 6.15: Accent ratings per speaker and method. For each speaker the gender (m/f) and L1 language is given in the title.

Figure 6.15 shows the accent ratings for all speakers and methods. It can be seen that the ratings between speakers varied a lot, with speaker 2 (male, EL – Greek) having the highest ratings (median 81.5 for "rec", 86.5 for "adapt"). The non-accented speaker is not shown as he was correctly identified as non-accented and received a median accent rating of 0. After the evaluation the listeners reported that they felt able to detect the presence of an accent but were unable to identify it most of the time. Many stated that they rated accents not too high as long as "they were able to understand the speaker." Conversely they tended to give higher accent ratings if the quality of the sample was degraded, despite being told to try to ignore quality issues. This was especially true for samples created by the KLD-based methods which often introduced errors in the synthetic speech.

Speaker identity

We assessed speaker identity by a speaker discrimination task. Our experiment setup was similar to the setup proposed in [125]. The listeners were presented with two samples at a time and had to vote if the same speaker could be heard in both samples. The listeners were told to try to ignore quality and accent of the samples and focus on the voice itself. The evaluation data consisted of 12 utterances uttered by 10 speakers using 4 methods. Each of these samples was paired with a sample of the same speaker and with a sample of another random speaker of the same gender for all other methods. Also the two utterances within each pair differed. The sample pairs were then equally distributed to all listeners, resulting in 120 pairs (i.e. 240 samples) for each listener.

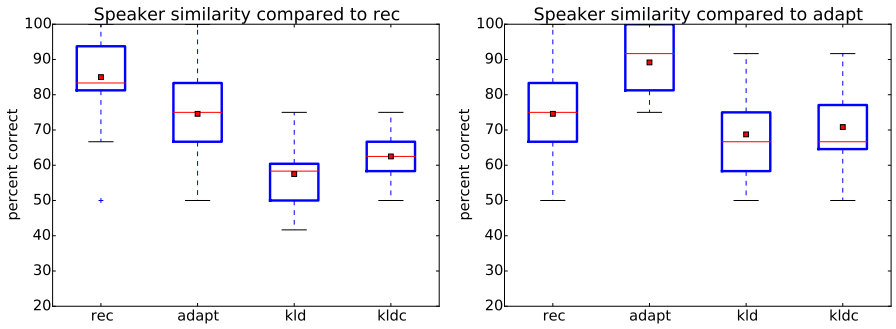

Figure 6.16: Speaker discrimination task results with reference method "rec" (left) and "adapt" (right).

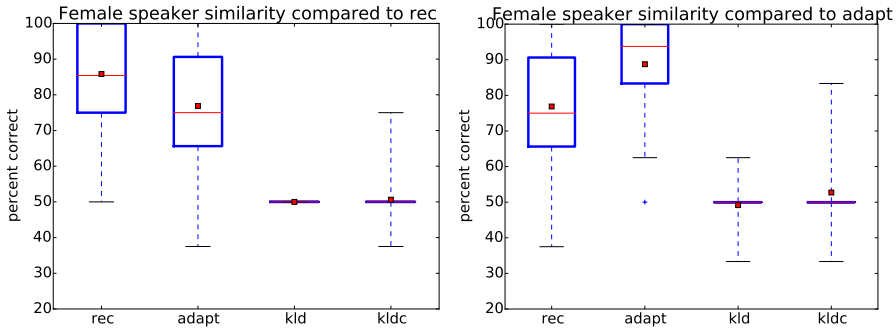

Figure 6.17: Speaker discrimination task results for female speakers with reference method "rec" (left) and "adapt" (right).

The overall results for the speaker discrimination task for "rec" and "adapt" samples can be seen in Figure 6.16. This shows how often listeners correctly identified the speaker in the presented sample when compared to the same speaker from either a sample of the recordings or from the synthesis of an adapted model. This shows that the KLD-based methods degrade speaker similarity. When listening to the synthesized samples, we noticed a bias in speaker

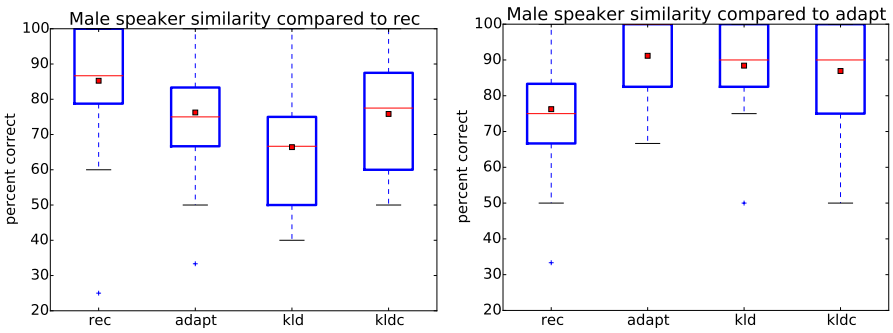

Figure 6.18: Speaker discrimination task results for male speakers with reference method "rec" (left) and "adapt" (right).

similarity between male and female speakers. We hypothesized that this is due to the fact that the average voice model was built from male speakers only.

In Figures 6.17 and 6.18 it can be seen that the speaker similarity between recorded and adapted synthetic speech was not significantly different for male and female speakers. This means that the adaptation of a female voice from a male average voice retained the speaker identity at a similar level as the adaptation of a male voice. In Figure 6.18 it can also be seen that the median percentage of correct discriminations of adapted male speakers from KLD-based speakers was at about 90% compared to 50% for the female speakers shown in Figure 6.17. The results show that the KLD-based methods are generally able to retain speaker similarity, but fail when the adapted voice is of a different gender than the average voice model. This suggests that a gender-matched or otherwise similar average voice model should be used for KLD-based accent conversion.

6.6 Discussion

In this chapter, a method for cross-variety transformation based on state mapping [128] has been presented. We described our approach to regression tree generation and the integration of structural information into the mapping process. During the subjective listening test it became evident that regular speaker adaptation is not sufficient for cross-variety adaptation. The method presented here greatly improves the quality of the generated voice. Unfortunately it is still very dependent on good quality average voices. An average voice biased to a very distinctive speaker will also introduce noticeable elements of this speaker into the adapted voice. Also, errors in the synthesized speech of the average voices will also be noticeable in synthesized speech of the adapted voice. This is especially relevant for varieties with few available speech data.

In the subjective evaluation, we saw that both data mapping methods can retain speaker similarity to a high degree and variety similarity to a smaller degree. In the pairwise comparison we did not see significant differences between the two methods. This conforms with prior work where different data mapping approaches also led only to subtle changes in the results [57]. We

also performed an objective evaluation for the bi-lingual data of speakers in two varieties. One of four speakers showed a significant improvement in mel-spectral distortion for CPDM over DM. We also noticed specific cases of reduction in synthesis errors with CPDM in a manual analysis. We also saw similar effects in the subsequent experiments on accent conversion (see Section 6.5), where constraining the mappings was highly beneficial for some speakers while making no difference for the other speakers. It remains an open question which factors cause this effect. But as CPDM did not lower the performance of the method and is also computationally inexpensive, we recommend including constraints for these special cases.

[125] showed that listeners perform significantly worse in speaker discrimination tasks when natural and synthetic voices are mixed. We could also observe this in our experiments as well as the fact that natural voices received higher accentedness ratings from our listeners than the synthetic voices. [52] attributes this effect to the over-smoothing nature of HSMM-based speech synthesis. In [126] it was also shown that listeners were significantly less accurate in cross-lingual speaker discrimination and [26] also revealed a strong coupling between accent and speaker identity. We hypothesize that the listeners of our experiment were also influenced by these effects, although we are not able to show the influence of accent on speaker discrimination with our experiments. To measure this effect we would need a different experiment design.

While the advantage of the method presented here is that it does not need an average accent voice we also noticed a higher quality degradation in the accent-reduced samples when compared with our previous experiments on dialect transformation [114] where we have used a dialect average voice. We hypothesized that the constrained mapping method is less effective for accent reduction; this could not be confirmed. As we know from previous experiments this method is able to alleviate the quality problem to a certain extent, so the constrained mapping method should be preferred. Future work would include a subjective listening test on the quality difference between the constrained and unconstrained KLD methods. It would also be interesting to quantify the difference in quality when using an accented average voice model instead of the accented speaker dependent voice model.

We found that speaker similarity of the synthetic voices was degraded when using an average model of different gender than the accented speaker. When using an average voice model of the same gender, speaker similarity was on a similar level as that of the adapted voices. This means that a more careful selection of the average voice model used is necessary than with regular adaptation. This also suggests that our hypothesis that phone identity overruled speaker identity in the KLD mapping might be wrong.

We also found that listeners rated the accentedness of synthetic speech lower than that of recorded speech. Listeners were able to detect the presence and rate the degree of an accent but were barely able to identify the (European) accents of Austrian German correctly.

CHAPTER 7

Synthesis of fast speech

Previous studies have shown that blind individuals are capable of understanding speech speaking rates up to a rate of 22 syllables per second, which is more than five times faster than typical speaking rates [64]. Therefore they often set reading devices to higher speaking rates, depending on how fast they would like to scan for information. To attend for such usage, speech synthesis systems should provide synthetic speech that remains intelligible at different speaking rates.

The statistical parametric paradigm for speech synthesis is beneficial for this task. These systems can generate intelligible speech from small databases, providing synthetic voices that match a certain speaker or accent without the need for long and high quality recordings. This can be particularly useful for blind children at a learning stage. It has been shown that familiarity with a speaker has a positive impact on the intelligibility of speech [74]. This technology could provide children with screen readers and tutoring tools that use voices built from familiar persons like their teacher's, friends or their own voice. We have shown that these beneficial effects on the learning experience can also be seen when using synthetic speech [82].

HSMM-based synthesizers model duration by using explicit state duration distributions, usually a Gaussian distribution (see also Section 2.3). To create speaking rates that are faster than the ones observed in the training data, the state-level duration can be increased by a factor proportional to the variance of the state duration, which we call "variance scaling" here. However, this method has not proven to be very effective at generating intelligible speech at higher speaking rates [80, 103]. Pucher et al. found interpolating between state durations of a normal rate and a fast rate voice model produced speech that was more intelligible than speech produced by variance scaling [80]. The fast rate voice model was trained from recordings where the speaker was asked to speak as fast as possible while still retaining intelligibility.

In this chapter, we analyze two aspects of producing fast synthesized speech. First we compare nonlinear manipulation of speech duration with linear compression. Second we investigate the influence of the training data, i.e. if recordings of fast speech can be helpful. We evaluate intelligibility of a fast and a normal female Scottish voice and an Austrian male voice, natural and synthetic, compressed using two nonlinear and one linear method and presented to listeners at different rates.

Parts of this chapter have been published previously in [120, 121]. This work has been a collaboration with the University of Edinburgh. The author of this thesis was responsible for the HSMM-based methods and the experiments with the Austrian German voice and the blind listeners. The partners at the University of Edinburgh conducted the same experiments in Scotland with the Scottish English voice and provided the methods based on signal processing as well as the data annotation.

7.1 Problem definition

This chapter treats Problem 4, defined as:

- **Problem 4**: Given a set of voice models $M_1 ... M_n$ in different speaking rates, a phonetic transcription of an utterance U and a parameter α specifying the output speaking rate, synthesize a speech sample of U in the desired speaking rate α.

Therefore a system to solve this problem has the following **input**:

- A set of voice models $M_1 ... M_n$ in different speaking rates.
- Output speaking rate α.
- Phonetic transcription of output utterance U.

The system should produce the following **output**:

- Synthesized speech sample for utterance U in the speaking rate α.

In this chapter we investigate Hypothesis 4:

- **Hypothesis 4**: HSMMs can be used to produce fast speech which is more intelligible than by using linear signal compression.

7.2 Time compression methods

In this section, we describe the methods for creating synthetic speech at fast speaking rates which we evaluate in this chapter. We also refer to them as time compression methods. The first two methods manipulate speech at the acoustic model level by modifying the HSMM state durations (mean and/or variance) while the third is applied to the synthesized speech waveform. The first two methods are considered to be nonlinear as each state is compressed at a different rate, as opposed to the third method, which is a linear method that compresses the waveform relatively uniformly across time and different speech units.

Variance scaling

Variance scaling is proposed as the standard method for duration control in HSMM-based synthesis [138], also described previously in Section 2.3. At generation time, the duration of state k is computed as shown in Equation 7.1, where μ_k and σ_k are the mean and the variance of the duration distribution of state k and ρ is a parameter to scale the duration variance. When $\rho = 0$ the duration is set to the mean state duration. Under this setting the synthesized voice should have the normal speaking rate of the training data. Turning $\rho > 0$ makes the synthetic speech slower and $\rho < 0$ makes it faster. ρ can be set according to a desired total duration as shown by Equation 7.2. From the values of d_k, we know how many frames are emitted by each state and therefore the state sequence. When synthesizing a sentence it is possible to set a desired total duration T. From the equations above, we are able to calculate the state duration d_k, however, rounding errors in the process of approximating a real value (time) to an integer value (number of states) means that the generated sentence will not necessarily have exactly duration T.

$$d_k = \mu_k + \rho\, \sigma_k^2 \tag{7.1}$$

$$\rho = \left(T - \sum_{k=1}^{K} \mu_k \right) \Big/ \sum_{k=1}^{K} \sigma_k^2 \tag{7.2}$$

The scaling factor is fixed across all states. State duration control is then proportional only to the variance: states whose duration model variance is higher will be compressed more. With this method we can potentially capture certain non-linearities between normal and fast speech durations seen in the training data. In the same way as in fast speech, vowels are more compressed than consonants, this method should also compress such units more as the duration model variance of states referring to vowels is higher [80].

Model interpolation and extrapolation

In previous work with fast synthetic speech, [80] showed that model interpolation [105, 140] can outperform the variance scaling method in terms of intelligibility and listener preference. Given two voice models of the same speaker trained with speech recorded at normal and fast speaking rates, the most successful method in that study was one that applied interpolation between duration models only, using the normal speaking rate models of cepstral, fundamental frequency and aperiodicity features. The interpolated duration d_i for state i is calculated as in Equation 7.3:

$$d_i = (1 - \alpha)\, \mu_i^n + \alpha\, \mu_i^f \tag{7.3}$$

where μ_i^n and μ_i^f denote the mean duration of state i in the normal and fast duration model and α is the interpolation ratio to control the speaking rate. We can generate speaking rates beyond the rate of the fast model by extrapolating ($\alpha > 1$). Interpolation itself is a linear method at a state level as the duration of each state is modified by a linear term. However because the duration change is different for different states, i.e. different speech segments, we classify it here as a non linear time compression method.

For the experiments in this chapter, we have implemented an additional constraint in this method. It is possible that for a given state of a given phone, the mean duration μ_i^f from the fast model is actually longer than the mean duration μ_i^n of the normal model, causing the speech segments generated for this state to become *slower* with growing α. If this is the case, we do not interpolate or extrapolate, but apply a linear factor β to μ_i^n, where β reflects the overall mean speaking rate difference between the normal and the fast voice models ($\beta = 1/1.55$ in our experiments).

WSOLA

The Waveform Similarity OverLap and Add (WSOLA) method was chosen here to achieve linear compression [122]. The method provides high enough quality while being computationally efficient and robust [122]. In WSOLA, speech frames to be overlapped are first cross-correlated to provide an appropriate time shift that ensures frames are added coherently, inspired by the idea that modified speech should maintain maximum local similarity to the original signal. The compression is relatively linear across time as the method allows a tolerance region within which a window can be placed in order to allow for a maximum similarity to be reached. The similarity measured can for instance be the cross correlation or the cross average magnitude difference function (AMDF) [92] between the downsampled sequence underlying a particular segment and that segment input sequence.

7.3 Speech databases and voices

In this section, we present the German and English corpora used in our experiments as well as details on how we trained the synthetic voices. The synthetic voice described as the fast voice was built by adapting the duration model only to the fast speech data as [80] reported it results in more intelligible voices.

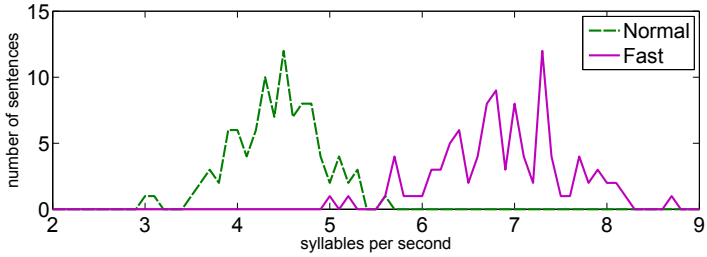

Figure 7.1: German synthetic voices: syllables per second distribution calculated from labels. Average values for Normal and Fast: 4.5 and 7.0.

German – corpus and voices

For the Austrian German part of the experiments, we used the SAG corpus as described in Section 3.2 with 4387 sentences at a normal and 198 sentences at a fast speaking rate. Differences

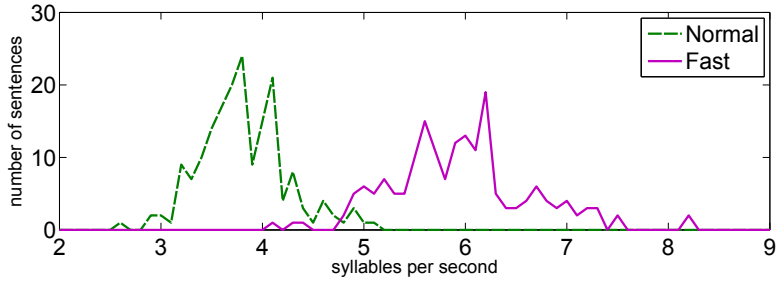

Figure 7.2: English synthetic voices: syllables per second distribution calculated from labels. Average values for Normal and Fast: 3.8 and 6.0.

to the English corpus are that the recordings were sampled at 44.1 kHz and that 40 Mel-cepstral coefficients were extracted. Otherwise the procedure and parameters were the same as for English.

The average SPS values for the normal and fast German synthetic voices are 4.5 and 7.0, and the WPM values are 152.7 and 237.1, respectively. Therefore, the German voice is considerably faster than the English voice at both speaking rates when considering SPS. The WPM values are difficult to compare as the German language makes use of compound words, so German words tend to be longer on average (for example consider English "apple pie" vs. German "Apfelkuchen"). Interestingly, the fast model is also about 1.55 times faster than the normal model, i.e., the speed-up factor between the two English models and between the two German models is the same. Figure 7.1 shows the SPS distribution for the two German models.

English – corpus and voices

We recorded a Scottish female voice talent reading 4600 sentences at a normal speed and 800 sentences (contained in the normal read prompts) at a fast speed. The instruction was to speak as fast as possible while maintaining intelligibility.

To train the acoustic models, we extracted the following features from the natural speech sampled at 48 kHz: 59 Mel cepstral coefficients [30], Mel-scale fundamental frequency ($F0$) and 25 aperiodicity features [48].

We trained two voices. What we refer to as the model N, is a voice trained only with speech produced at the normal speaking rate. The duration model was adapted (see speaker adaptation [134] in Chapter 2) using the 800 sentences of fast speech to create what is referred to as the voice F.

To measure the speaking rate of each synthetic voice we calculated the rate of syllables per second (SPS) and words per minute (WPM) for each sentence used in the evaluation. On average the SPS values of the normal and the fast voice are 3.8 and 6.0 while the values for WPM are 206.7 and 320.9, respectively. Speech synthesized using the fast model is around 1.55 times

faster, which agrees with the literature [46] on naturally produced fast speech. Figure 7.2 shows the histogram of SPS across synthesized sentences for each voice.

7.4 Evaluation

For both the English and the German voice we conducted two listening experiments, one using natural- and the other synthetic speech. For the German voice we conducted the same listening experiment also with blind individuals [1].

We evaluated intelligibility of time-compressed speech at four different speaking rates: 1.25, fast (the speed of fast speech, i.e. roughly 1.55), 2.0 and 3.0, where numbers refer to speed increase with respect to the normal speech calculated sentence by sentence with fast speech being around 1.55 times faster than normal speech. Rates were chosen to reflect conversational, fast, and two very fast speeds.

The acronyms for the corpus – compression method combinations we evaluated are presented in Table 7.1. Not all methods were evaluated at all speaking rates, for instance at rates smaller or equal to the fast rate W-F, V-F and I were not evaluated as that would mean slowing down the speech signal. Also for the natural speech evaluation only the WSOLA algorithm was used as the other two methods cannot be applied directly to natural speech. To generate compressed samples using the variance and the interpolation methods it was necessary to iteratively change the scale factor to obtain the desired duration. The implementation of WSOLA used here was provided as support material in [87]. We used the AMDF as the similarity function and 5 ms as the size of the tolerance window as proposed in [122].

Results are presented as percentage of word errors, calculated per listener as the percentage of words that were not found in listener's transcription, misspellings taken into account, and then averaged across listeners. The bars refer to standard deviation of the error calculated across listeners' results.

W-N	WSOLA applied to normal speech
W-F	WSOLA applied to fast speech
V-N	Variance scaling applied to model N
V-F	Variance scaling applied to model F
I	Interpolation of model N and F

Table 7.1: Methods evaluated.

German – evaluation

For the synthetic speech evaluation, we compared the three different compression methods described in Section 7.2, although not all methods were evaluated for all speaking rates.

Our natural speech evaluation took place at a later date where we compared two natural speech compressions: W-N and W-F - WSOLA compression applied to the normal and the fast

[1] Speech samples used in the evaluation can be found at: http://wiki.inf.ed.ac.uk/CSTR/SalbProject

speech databases. All speaking rates were evaluated except the 1.25 rate, which we found to be an too easy a task already in the evaluation of synthetic samples.

Listening experiment

The participants in the listening test for synthetic speech consisted of two groups: 16 blind or visually impaired participants, 15 of whom reported using synthesis systems (i.e. TTS) in their everyday life, and 16 sighted participants with no TTS expertise. Each participant transcribed 100 sentences, under the constraint that each sentence was assigned only once to each person. So each group transcribed a total of 1600 sentences, resulting in 3200 sentences for both groups. As all 100 sentences were produced 16 combinations of method and speaking rate, each sentence was heard once in each variant per group.

The participants in the listening test for natural speech also consisted of two groups: 10 blind or visually impaired participants, 10 sighted listeners. Each person transcribed 60 sentences, again under the constraint that each sentence was assigned only once to each person. Therefore each group transcribed a total of 600 sentences, 1200 sentences for each group. In this experiment we had 6 combinations of method and speaking rate for the 60 sentences, so each sentence was heard in each variant at least once, at maximum twice, per group.

Sentences were selected from news articles and parliamentary speeches for the synthetic speech experiment and from a similar corpus of recordings for the natural speech experiment. Listeners were presented a web interface where they were allowed to play a sample (i.e. a sentence) only once an then enter the transcription in a text field. A button then led the participant to the next sample. Blind listeners were assisted by an operator in playing the samples and entering the transcription.

Results for sighted listeners

Figure 7.5 shows the percentage of word errors for each speaking rate obtained in the natural speech (blue, wide bars) and synthetic speech (red, narrow bars) experiments.

For the synthetic speech (red, narrow bars), we can see that WSOLA compression of speech synthesized from the normal model (W-N) is the best method overall. However, up to speaking rate 2xs, both WSOLA of fast speech (W-F) and interpolation (I) yield results competitive to W-N. At the "fast" rate, where both W-F and I (and also V-F) are equivalent to simply the fast voice model, these methods even achieve significantly better results than W-N for the sighted listeners. At the "fast" and 2xs rates, W-F and I perform significantly better than variance scaling of the normal model (V-N), confirming the results of [80]. However, we see a very clear advantage of the WSOLA methods at the fastest rate 3xs, where the error percentages of V-N, V-F and I are much higher, yielding a picture similar to the English results at 2xs. There is no significant difference between W-N and W-F at the 2xs and 3xs rates.

The natural speech error scores (blue, wide bars) were significantly lower than the scores obtained for synthetic speech and this gap grows with higher compression rates. At moderate rates, fast and 2.0, W-N and W-F results are comparable but at the highest rate compressing fast speech creates more intelligible stimuli, unlike the findings for the English data. This will be further investigated in Section 7.5.

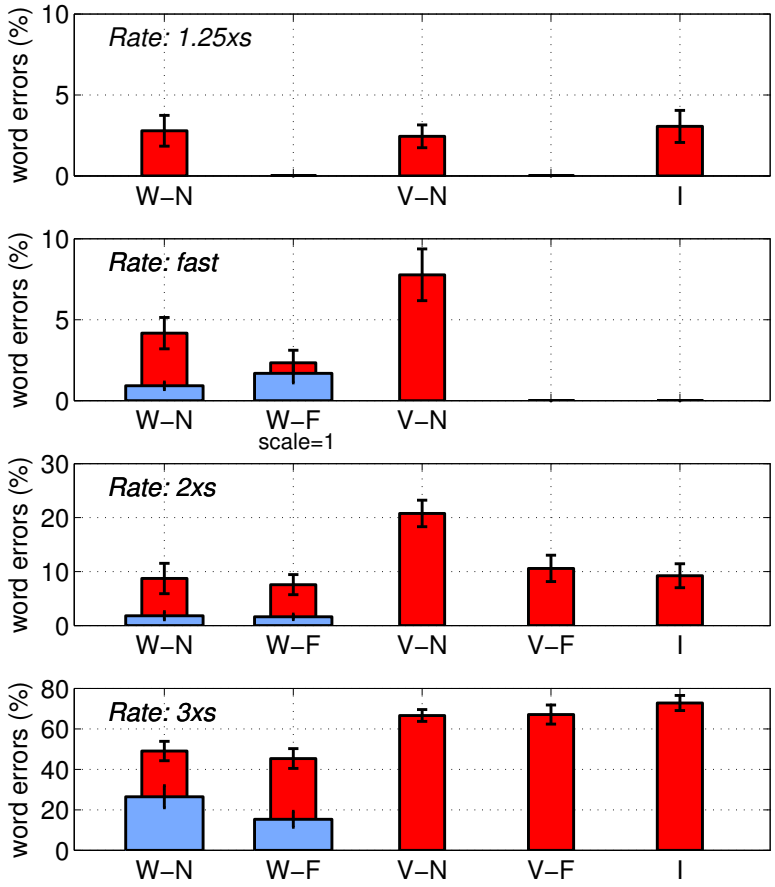

Figure 7.3: German results with sighted listeners: synthetic (red, narrow bar) and natural speech (blue, wide bar).

Results for blind/visually impaired listeners

As shown in the literature [80, 102, 118], blind listeners generally achieve lower word error percentages than sighted listeners. In Figure 7.4 we can see that our evaluation confirmed these findings. Word error rates were generally lower for the blind listeners for all methods. This is especially noticeable for the WSOLA compressed samples at 3xs rate, where the blind listeners achieved word error rates between 20 and 25%, while the word error rates of the sighted listeners were between 40 and 45%.

The overall picture concerning the different methods is very similar to the results of the experiments with sighted listeners – still the WSOLA methods perform best at higher speaking rates. The advantage of using a linear compression method seems even larger here, particularly at the highest speaking rate, where W-N and W-F errors were around 22%, a result much better than the one obtained by sighted participants, but V and I methods' results were still between 50 and 60% a result comparable to the one obtained by sighted listeners.

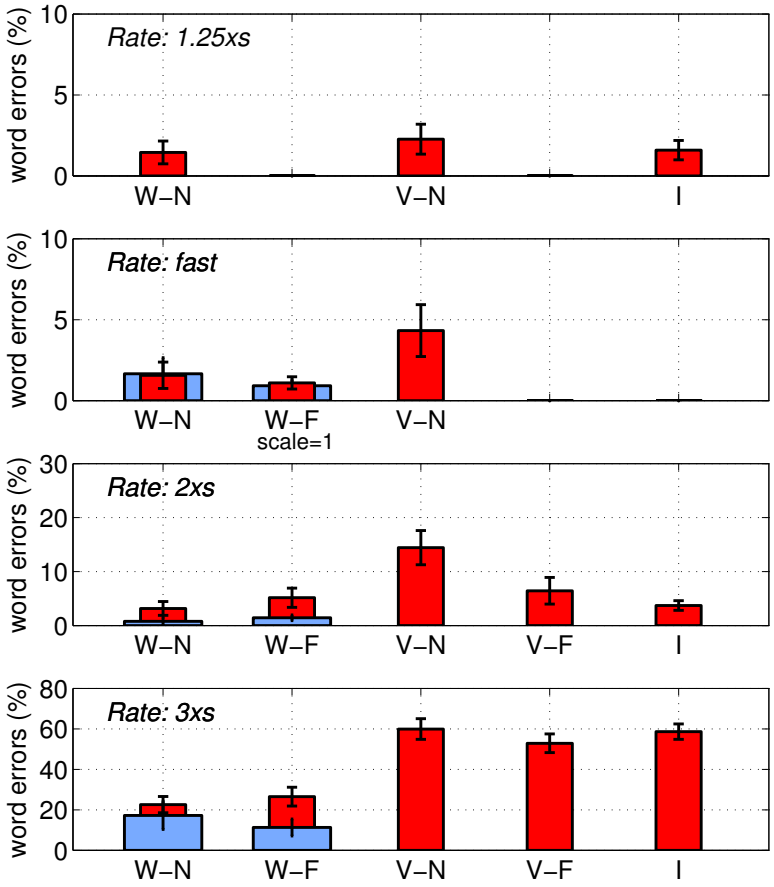

Figure 7.4: German results with blind/visually impaired listeners: synthetic (red, narrow bar) and natural speech (blue, wide bar).

English – evaluation

A similar evaluation was carried out for English to assess the intelligibility achieved by the methods described in Section 7.2.

Listening experiment

We performed two listening experiments, one with natural speech and the other with the synthetic voices. Each experiment was performed by 20 native English speakers without TTS expertise. Each participant transcribed 100 sentences for each of the tested methods. The natural speech sentences were selected from news articles while for the synthetic speech experiments sentences were chosen from the first few sets of the Harvard dataset [43].

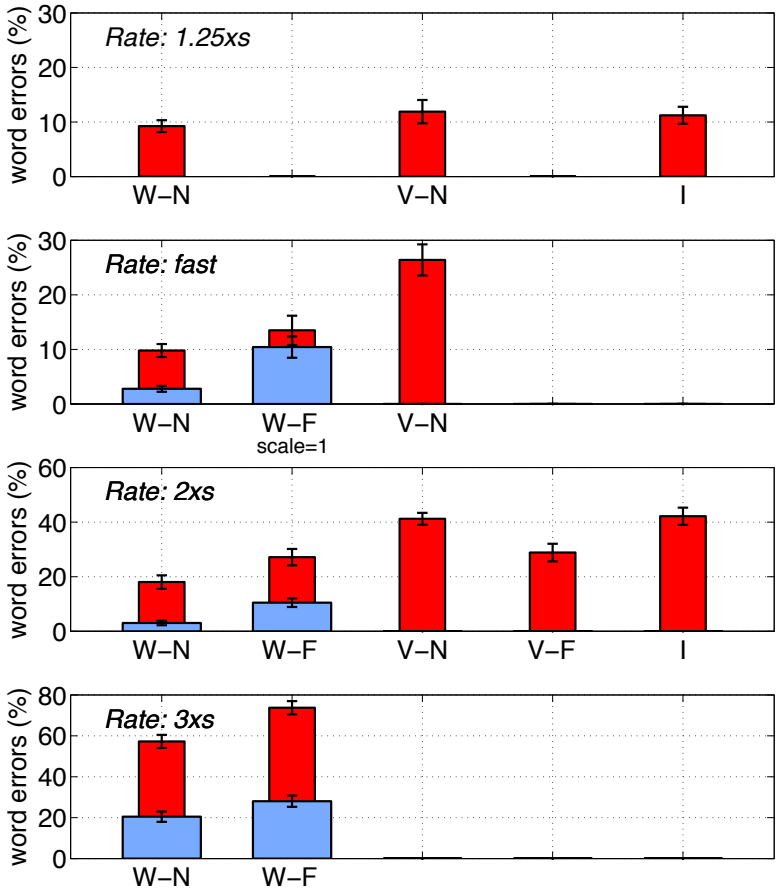

Figure 7.5: English results: synthetic (red, narrow bar) and natural speech (blue, wide bar).

Results

Figure 7.5 shows the percentage of word errors for each speaking rate obtained in the natural speech (blue, wide bars) and synthetic speech (red, narrow bars) experiments.

We can see that the synthetic samples (red, narrow bars) created using WSOLA are the most intelligible across all tested speaking rates and that this advantage grows with increasing speaking rate. At the 2xs rate the synthetic voice W-N results in less than 20 % word errors while the word errors obtained by V-N and I are higher than 40 %, i.e., errors doubled. Results for W-F and V-F are better, just around 30%. Word errors are smaller when compressing speech synthesized from the normal model (W-N) as opposed to a fast model (W-F), as results for speaking rate 2xs and 3xs show. Error level for V-F is however smaller than V-N. At the fast speaking rate, we can see that the fast voice is less intelligible than the normal voice with linear compression applied.

The error scores for natural speech (blue, wide bars) are significantly lower than the ones

obtained for the synthetic voices, and the intelligibility gap grows with speaking rate. The increase in error seen for W-F when compared to W-N for synthetic voices can also be observed for natural speech, pointing to the fact that the fast natural speech is also less intelligible than linearly compressed natural, normal-speed speech.

7.5 Annotation analysis of fast speech

An intelligibility advantage when using fast speech to produce either natural or synthetic speech at higher speaking rates was only found for the German database. To better understand these results we performed a manual annotation of the natural speech data at a phonemic level, i.e. determining which phonemes were pronounced and where their boundaries lie. This section presents details of the annotation and its results.

Annotation procedure

We annotated phone and syllable level information, i.e. time stamps and phone identification, of recordings of each speaker reading the same sentence material in two different speaking styles: plain and fast.

To choose the sentences to be manually annotated, we randomly drew 40 sentences out of the sentences used for the listening test with natural speech. As it is easier for the annotator to start with an automatic segmentation of phone borders, we provided initial segmentations by retrieving phone duration generated by the models built for that speaker and those sentences. With the label files generated after alignment and the wavefiles, the annotators used Praat [10] to perform the task. Figures 7.6 and 7.7 show a Praat window for one sentence in the English and German corpus. Here we can see the phone, syllable and orthographic levels and the boundaries set by the annotators.

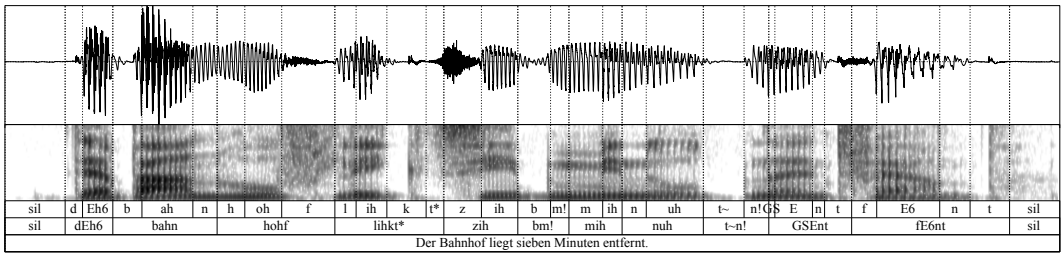

Figure 7.6: German fast sentence in Praat, phone symbols are from the Austrian German lexicon [72] .

As we are interested in comparing differences across styles it was important to be able to annotate the changes observed by our annotators. In general, transcription was fairly broad i.e. phonemic, however, particularly for the fast speech, it was noted that a slightly narrower transcription would be desirable, as the speaker employed certain strategies in order to speed up, such as not achieving closure for plosives, and occasionally not releasing plosives. Because it is not necessary to have a fully narrow transcription at the phonetic level, but we do want to

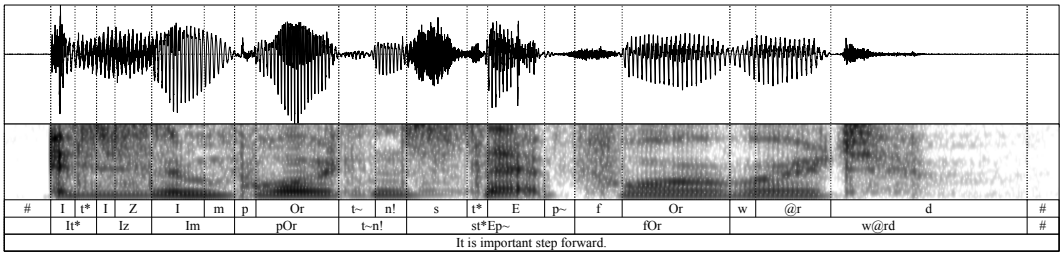

Figure 7.7: English fast sentence in Praat, phone symbols are from the Combilex lexicon [90].

know about such cases where plosives are not fully realised, symbols were added to describe unreleased plosives and plosives without complete closure.

Syllable boundaries were generally kept faithful to the canonical syllabification as provided by the pronunciation dictionary. The syllable boundaries were not changed unless there were deletions which then changed the syllabification (e.g. something deleted a syllable boundary). Although one approach could have been to attempt to model resyllabification by the speaker - particularly in fast speech, where for example the coda of one syllable becomes the onset of the next - this was not attempted due to the arising ambiguity around such cases. Syllable boundaries are therefore always within-word or at word boundaries and do not span word boundaries.

The English data was annotated by an experienced annotator who is a native English speaker. The German data was annotated by a German native speaker and cross checked by the English annotator for consistency across languages. It was noted by the annotators that phoneme boundary decisions were easier to make in the German fast data than in the English fast data, as the articulation changes were more pronounced and therefore visually present in spectrograms, as seen in the example of Figures 7.6 and 7.7. In this example the syllable per second and increase in speaking rate relative to normal speech is comparable across languages, yet the spectrogram of the German sentence is more contrastive.

Results

We present results for both languages (speakers) in Table 7.2. The total number of phones annotated in the normal rate data is presented in the first column. The other columns present hits (phone correct assignment), deletions, substitutions and insertions of phones annotated in the fast speech compared to phones annotated in the normal speech for both English and German. The last column refers to the phone error rate calculated for the fast speech annotation having the normal speech annotation as the reference. This should give us a reasonable indication of how intelligible the fast data is. For both German and English we annotated 40 sentences but some sentences of the German data were particularly long, so more phones were annotated for that language.

Figures 7.8 and 7.9 present the pattern of compression rate and absolute decrease in duration (in ms) across different phonetic units. To calculate this we took the average across phone occurrences within each phonetic class, the standard deviation refers to the deviation across

	phones	hits	deletions	substitutions	insertions	PER
English	857(56)	717 / 84%	45(16) / 5%	95(39) / 11%	0	16.34%
German	1480(67)	1359 / 91%	62(18) / 4%	55(36) / 4%	4(4)	8.20%

Table 7.2: Annotation results: number of phones annotated in the normal data, number of phones hits, deletions, substitutions and insertions in the fast data. Numbers in parenthesis refer to unique counts. PER refers to phone error rate of the fast speech transcription compared to the normal speech.

phone appearances. Only phones that occurred more than five times in the annotation were considered to remove any outliers. Deletions were not counted here, but pointed out separately in the table mentioned previously. The phonetic classes considered were: vowels, nasals, stops, approximants and fricatives.

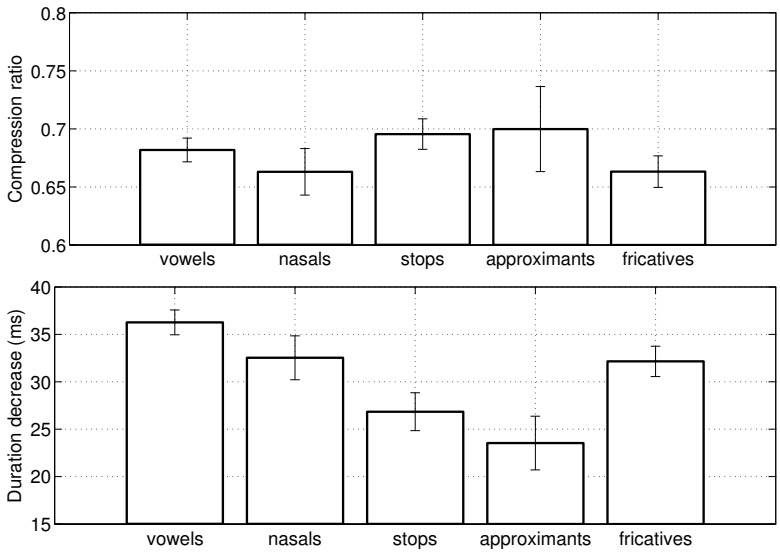

Figure 7.8: German annotation results: compression ratio (top) and absolute duration decrease in ms (bottom).

German

For the German data 1480 phones were annotated from 40 sentences of normal speech, 67 of which were unique. Contrasting the plain with the fast annotation we found that 91% were present in the fast data while the rest was either deleted, substituted or inserted. The most common substitutions were incomplete closure in plosives (7 times), replacement of schwa by

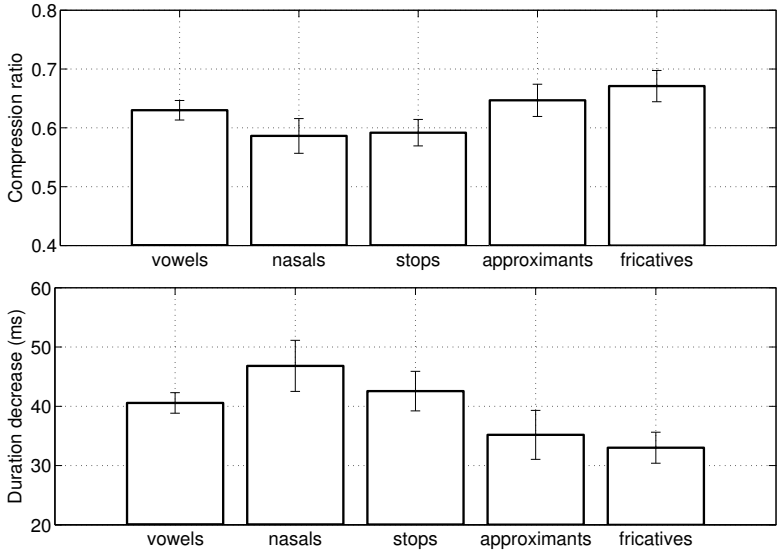

Figure 7.9: English annotation results: compression ratio (top) and absolute duration decrease in ms (bottom).

[e] (5 times), replacement of [eɒ] by [eːɒ] (4 times), and vowel reductions to schwa (4). The most common deletions were pause removals (27 times), removal of glottal stop (7), and removal of [t] (6), [r] (4), [d] (3). Furthermore we had insertions of glottal stop and [g] (1 each), and two insertions for splitting up compound phones. The phone error rate of the fast speech data is 8.2%, which is much lower than for the English speaker. This also shows that the German speaker produces more consistent speech output when speaking fast.

Figure 7.8 shows the compression ratio and duration decrease per phonetic class for the German speaker. We can see that this speaker compressed the most nasals and fricatives and the least stops and approximants. The absolute decrease in duration for stops and vowels was different, as expected according to the literature on fast speech of other speakers.

Figure 7.10 (left) presents the distribution of syllables per second across the annotated sentences for normal and fast data. The average syllables per second for the normal case is 4.2 and fast 7.0, which was quite similar to the results obtained with the automatic segmentation presented in Section 7.3. For German, these values were the same as when calculated from the labels, as can be seen in Figure 7.10 (right).

English

For the English data, as seen in Table 7.2, 857 phones were annotated from 40 sentences of normal speech, 56 of which were unique. Contrasting the plain with the fast annotation we found that 84% were present in the fast data while the rest was either deleted or replaced. The most

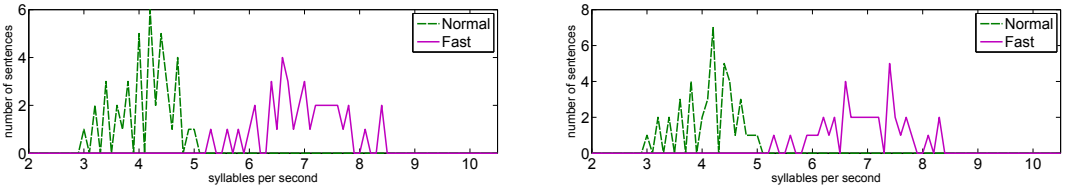

Figure 7.10: German: Syllables per second. Annotation average values for Normal and Fast: 4.2 and 7.0 (left). Labels average values for Normal and Fast: 4.2 and 7.0 (right).

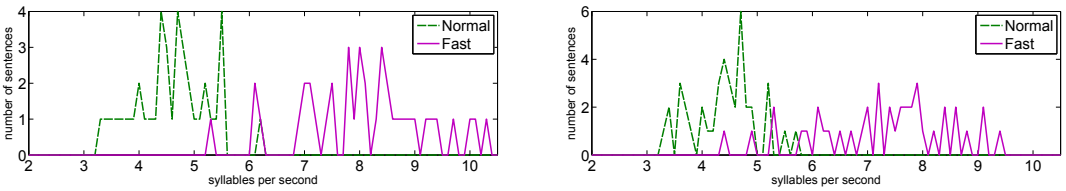

Figure 7.11: English: Syllables per second. Annotation average values for Normal and Fast: 4.7 and 8.1 (left). Labels average values for Normal and Fast: 4.5 and 7.3 (right).

common substitutions were plosives being replaced by unreleased plosives (35 substitutions) or incomplete closure in plosives (18), vowel reductions to schwa (14) and [t] becoming a glottal stop (5). The most common deletions were pause removals (9 times), [t] (7), schwa (6) and, [l] and [h] (4 times each). The phone error rate of the fast speech data is of 16.34%.

To understand the durational changes made by the speaker we present in Figure 7.9 the compression ratio and duration decrease per phonetic class. We can see that this speaker compressed nasals and stops the most and fricatives and approximants the least. The absolute decrease in duration for stops and vowels was similar, vowels were, however, expected to shorten more according to the literature on fast speech of other speakers.

Figure 7.11 (left) presents the distribution of syllables per second across the annotated sentences for normal and fast data. The average syllables per second for the normal case is 4.7 (4.8 with pauses) and fast 8.1. That means that the speaking rate with pause was found to be 1.76 (without pause 1.78) which is higher than we calculated from the labels that trained the models, which were automatically aligned. The labels give slightly smaller values for the same 40 sentences, as can be seen in Figure 7.11 (right), on average 4.5 and 7.3. We found that in some cases the initial phone includes a large part of the preceding silence, which means that speech duration according to labels is longer than the manual annotation and this issue is more pronounced when automatically segmenting the fast speech.

7.6 Discussion

As found in other studies of natural fast speech [45, 46], our results using the English data also indicate that linear compression produces more intelligible voices than nonlinear methods based on or directly derived from the acoustics of fast speech. English results show that there is no additional advantage to using recordings of fast speech to build a synthetic voice and it is possible to maintain intelligibility at higher speaking rates by applying a linear compression method such as WSOLA to the synthesized waveform. This is supported by results with the natural speech corpus, where we also found that fast natural speech is not as intelligible as linearly compressed normal speech.

For German, we also see that linear compression is beneficial at very high speaking rates (3xs) compared to interpolation and variance scaling. For lower speaking rates (2xs), we find that interpolation is equally as good as linear compression. This indicates a potential use of a combined method of interpolation for fast speaking rates and linear compression for very fast speaking rates. We hypothesized that different results were found for the German data due to the inherent higher intelligibility of the German fast speech, which can also be seen in the performance differences of linear compression of synthesized speech from fast models (W-F) which performs better for the German data. The results with natural speech confirmed this as we observed a significant improvement at the highest speaking rate when compressing the fast German data, a result that was not seen in the English data.

Concerning the performance of blind listeners we can confirm results presented in previous studies [64, 80], which show that blind listeners achieve lower word-error-rates than non-blind listeners. Moreover, we observed that the WSOLA compression intelligibility gains were higher for blind users, possibly due to the fact that they are expert users of speech synthesis systems that use similar types of compression. The intelligibility gap between WSOLA and variance scaling or interpolation was larger with blind individuals, particularly at higher rates, another compelling reason to use a linear method.

The annotation results confirmed that the German natural fast data is more intelligible than the English data in terms of how many phones were found to be present in the fast data and the number of deletions and substitutions. The phone error rate was found to be twice as high for the English data. The most common substitutions for both speakers were plosives becoming incomplete and vowels replaced by schwa. The deletions in German were dominated by pause removal, which could explain how the German speaker was able to reach more phonetic targets while still maintaining a relatively high speaking rate. The pattern of compression across different phonetic units showed that the German speaker presents a similar pattern as to the one often presented in the literature, i.e. highly compressed vowels and nasals, while the English speaker most compressed unit was found to be the stops. A larger amount of the corpus would have to be annotated to make further statements but at this point we were able to identify that there were in fact large differences between the compression strategies of each speaker and that one was better able to correctly produce speech at higher rate.

While linear compression by signal processing methods performed better than the non-linear methods, we can achieve this effect in HSMMs by simple linear scaling of the HSMM state durations. We integrated this method in our speech synthesis framework presented in Chapter 3.

CHAPTER 8

Conclusion and outlook

In this work we treated the main aspects of dealing with language varieties in speech synthesis. We identified the problems of transformation, interpolation and acoustic modeling of language varieties. We created a corpus consisting of multiple speakers in 2 Austrian German dialects (IVG, GOI) and 9 different European accents of Austrian German, and used an existing corpus of Standard Austrian German (SAG) and Viennese dialect (VD) (see Chapter 3). We investigated, developed and compared different approaches for cross-variety transformation to convert the dialectal voices to SAG voices and vice versa. We also applied the developed methods to foreign accent conversion. Interpolation aims to generate intermediate language varieties of controllable degree, allowing flexible adaptation of the speaking style. We developed, evaluated and presented an unsupervised interpolation method and different variations thereof, applying it to interpolation between the Austrian German dialects. With the investigation of interpolation methods arose the special application of synthesizing fast speech by extrapolating natural fast speech. Also, various acoustic modeling methods to generate dialectal voice models were developed and evaluated. Lastly, we have also developed and presented an HSMM-based speech synthesis framework that allows straightforward integration of new languages and varieties for practical applications. In the next paragraphs, the chapters corresponding to each problem are summarized and potential future work is discussed.

In Chapter 4, we presented a method to automatically generate intermediate variants of language varieties. We have seen that linear interpolation of HSMM states mapped by DTW is feasible and produces intelligible and linguistically meaningful results. This means that users are able to perceive the different degrees of dialect and also to understand the intermediate speech. Expanding HSMM states with duration N to N states with duration 1, while allowing more different mappings, did not yield better results but increased computational cost. Extending the method to linearly interpolate phonological processes and using a step function for input-switch rules increased the acceptance as a dialect speaker while naturally resulting in a less smooth interpolation.

Future challenges include the investigation of the interpolation function so that the subjective perception has a stronger correlation with the interpolation parameter α. For example, a piecewise linear function could be employed. Another challenge is to develop complete interpolation systems employing machine translation to translate standard variety into dialect on an orthographic level. The output of such a translation, which can also include syntactic changes, should automatically provide input-switch rules on word-level. Rules for phones could be derived from phonetic criteria. The presented method can also be applied to other applications like speaker or emotion interpolation, interpolation in visual speech synthesis [97] or generation of additional training data. These are other possible directions of research which are yet to be investigated.

In Chapter 5 we presented and compared different approaches for acoustic modeling of language varieties. While previous experiments with similar-sized data sets found significant improvement in quality with average voice based methods, we saw no advantage compared to speaker-dependent models for the data set at hand. We hypothesize that the number of speakers used as well as the small phone set overlap for some of our dialects caused this effect. Still, the various types of models possess different characteristics that might be beneficial for other applications like speaker interpolation. Further work in this area should focus on experiments with larger datasets where clustering of similar dialects or possibly also speakers [17] becomes possible. This would allow to assess the factors that influence the quality of an adapted voice model for a given average voice model.

In Chapter 6, we treated cross-variety transformation in HSMM-based speech synthesis by investigating transformation between Austrian German varieties. While we found that, with the presented methods, overall voice quality and language similarity increased significantly, speaker similarity is still an important topic for further improvement. In [125] it was already shown that listeners perform worse when discriminating speakers between natural and synthetic speech. This can partially be attributed to the over-smoothing nature of HSMM-based speech synthesis and the quality of the vocoder, which generates the final waveform from the acoustic features. For evaluations and actual applications, this problem could be alleviated by a hybrid TTS system which replaces the waveform generation with a unit selection approach. The synthesized feature vectors are then used to find the most similar speech snippets in a large speech corpus.

The strong coupling of accent and speaker identity is an inherent problem that leads to listeners being significantly less accurate in cross-lingual speaker discrimination [126][26]. This can not be solved but can be considered when interpreting results from listening tests.

Our experiments to apply cross-variety transformation to accent conversion showed that the presented methods are able to significantly reduce foreign accent. We have seen that the incorporation of constraints is able to reduce synthesis errors and is therefore suggested. It also became evident that a similar average voice model (e.g. same gender) should be used when employing cross-variety transformation without an average voice model of accented speech. This approach could also benefit from a hybrid TTS system. An important research question regarding these methods is to assess the quality differences between the constrained and unconstrained KLD methods and also to quantify the difference in quality when using an accented average voice

model. Experiments to assess the influence of speaker identity on the KLD metric compared to phone identity could also give further insight on how to improve this method.

While we have seen that augmenting and constraining the HSMM state mappings can produce better results, overall we feel that there is not much more room for improvement by further varying the mapping constraints, so future approaches should focus on different parts of the transformation. Especially with the advent of deep learning based methods in speech synthesis [145], the question arises how cross-lingual and cross-variety transformation can be achieved with neural networks and if existing approaches can be adapted.

In Chapter 7, we presented results of intelligibility experiments with natural and synthetic speech of English and SAG voices reproduced at higher rates by a variety of methods. In our experiments, linear compression outperformed variance scaling and interpolation, particularly for very fast (3xs) speaking rates. For fast speaking rates (2xs) linear compression outperformed the other methods for the English data and was on par with interpolation for SAG. We can see that the quality of the fast recordings is of high importance for the both interpolation and linear compression of fast data.

Considering the results for both languages we hypothesize that methods that employ recordings of fast speech such as adaptation or interpolation probably produce speech that is only as intelligible as the fast speech data they use. Relying on having fast speech data that is intelligible enough is challenging it is quite difficult to produce, particularly considering that both of our speakers are already professional speakers. As our English voice was trained from much more fast speech data than the German voice, we saw that increasing the amount of training data does not necessarily produce better results. Variance scaling showed weak performance at fast speaking rates (2xs, 3xs) in our experiments, supporting the results for HMM-based voices in [103].

One challenge that arises is how to obtain intelligible, fast speech recordings. In our recording sessions, the speakers were instructed to speak as fast as possible while maintaining intelligible speech. However, as speakers had to read the sentences, they were forced to read and speak at the same time for longer sentences. To solve this problem, speakers could be instructed to memorize the recording scripts. Unfortunately this would further increase the effort needed for such recordings, especially if large data sets shall be recorded. Another possibility would be to record conversational speech, which humans naturally produce at faster rates than the read speech usually used to train voice models. Training a synthesis system with conversational speech is however still a challenging task.

Another arising research question is how to improve linear compression methods. We found the highest intelligibility scores for both for sighted and visually impaired users when using a linear compression method. Investigations of methods such as MACH1 [16] which employ silence removal could help to further improve upon these results. The effect of the frame shift chosen when extracting acoustic feature vectors on fast speech production could also be assessed. Current systems usually train voice models with 5ms displaced windows. A smaller frame shift could be beneficial for synthesis of fast speech.

Further future work includes investigating other sorts of speaking styles such as conversational speech as well as the combination of linear and non-linear methods. Also, studies of

actual application scenarios could be considered, e.g. tracking users of screen reader software. While it can be assumed that the methods with higher intelligibility are also preferred in these situations, additional insight might be gained that could indicate directions of further research. For example, different speaking rates might be used dependent on the content to be read.

In Chapter 3 we presented the SALB software framework which we developed and released to the public. It enables HSMM voice models created with the HTS toolkit [41] to be used as Microsoft Windows system voices and to be integrated in Android and iOS apps. The framework architecture encourages implementation of further languages [83]. Future work includes evaluating the user experience of screen readers and audio books with the framework as well as integration of novel methods, e.g. hybrid TTS or deep neural networks.

The topics treated in this thesis synergize well, so combining the presented methods for acoustic modeling, transformation and interpolation into a single speech synthesis framework seems reasonable. A possible approach would be to record a set of utterances from a user, use the adaptation technique to generate a voice model. This voice model could then be transformed to the different language varieties. The resulting set of models could then be integrated in a spoken dialog system that allows the user to select language variety and its degree. This could for example allow a personal digital assistant on a mobile phone or in a car to adapt to the user, possibly increasing acceptance of the assistant.

List of acronyms and abbreviations

AMDF Cross average magnitude difference function

AMTV Acoustic modeling and transformation of varieties for speech synthesis

AVDS Adaptive audio-visual dialect speech synthesis

CPDM Common phone data mapping

DM Data mapping

F0 Fundamental frequency

FTW Forschungszentrum Telekommunikation Wien

GOI Bad Goisern dialect

HMM Hidden Markov model

HSMM Hidden semi-Markov model

HTK Hidden Markov model toolkit

HTS HMM-based speech synthesis system

IPA International phonetic alphabet

IVG Innervillgraten dialect

KLD Kullback-Leibler divergence

MLLR Maximum Likelihood Linear Regression

MSD Multi-space distribution

PDF Probability density function

SAG Standard Austrian German

SALB Speech synthesis of auditory lecture books for blind children

SD Speaker dependent

SI Speaker independent

SPS Syllables per second

TTS Text to speech

VD Viennese dialect

WPM Words per minute

WSOLA Waveform similarity overlap and add

List of Figures

List of Tables

Bibliography

[1] P. Adank, A. J. Stewart, L. Connell, and J. Wood. “Accent Imitation Positively Affects Language Attitudes”. In: *Frontiers in Psychology* 4.280 (2013).

[2] M. Astrinaki, J. Yamagishi, S. King, N. D’Alessandro, and T. Dutoit. “Reactive accent interpolation through an interactive map application.” In: *Proceedings of the 14th Conference of the International Speech Communication Association (INTERSPEECH)*. Ed. by F. Bimbot, C. Cerisara, C. Fougeron, G. Gravier, L. Lamel, F. Pellegrino, and P. Perrier. Lyon, France: ISCA, 2013, pp. 1877–1878.

[3] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. “A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains”. In: *The annals of mathematical statistics* 41.1 (1970), pp. 164–171.

[4] R. Bellman. *Dynamic Programming*. 1st ed. Princeton, NJ, USA: Princeton University Press, 1957.

[5] H. D. Benington. “Production of Large Computer Programs”. In: *Annals of the History of Computing* 5.4 (1983), pp. 350–361.

[6] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal. “The AT&T next-gen TTS system”. In: *Proceedings of the Joint meeting of ASA, EAA, and DAGA*. Berlin, Germany, 1999, pp. 18–24.

[7] M. P. Bissiri and H. R. Pfitzinger. “Italian Speakers Learn Lexical Stress of German Morphologically Complex Words”. In: *Speech Commununication* 51.10 (2009), pp. 933–947.

[8] A. W. Black. “Unit Selection and Emotional Speech”. In: *Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH)*. 2003, pp. 1649–1652.

[9] A. W. Black, K. Lenzo, and V. Pagel. “Issues in Building General Letter to Sound Rules”. In: *The Third ESCA Workshop in Speech Synthesis*. 1998, pp. 77–80.

[10] P. Boersma. “Praat, a system for doing phonetics by computer.” In: *Glot International* 5.9/10 (2001), pp. 341–345.

[11] A. P. Breen and P. Jackson. “A Phonologically Motivated Method of Selecting Non-Uniform Units”. In: *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*. Sydney, Australia, 1998, pp. 2735–2738.

[12] Carnegie Mellon University. *Flite*. `http://www.festvox.org/flite/`.

[13] F. Charpentier and M. Stella. "Diphone synthesis using an overlap-add technique for speech waveforms concatenation". In: *Proceedings of the 1986 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 11. 1986, pp. 2015–2018.

[14] M. H. Cohen, J. P. Giangola, and J. Balogh. *Voice User Interface Design*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004.

[15] G. Coorman, J. Fackrell, P. Rutten, and B. Van Coile. "Segment Selection in the L&H RealSpeak Laboratory TTS System". In: *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP)*. Beijing, China, 2000, pp. 395–398.

[16] M. Covell, M. Withgott, and M. Slaney. "Mach1: Nonuniform time-scale modification of speech". In: *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 1. IEEE. Seattle, USA, 1998, pp. 349–352.

[17] R. Dall, C. Veaux, J. Yamagishi, and S. King. "Analysis of speaker clustering strategies for HMM-based speech synthesis". In: *Proceedings of the 13th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2012, pp. 995–998.

[18] S. B. Davis and P. Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences". In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 28.4 (1980), pp. 357–366.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society* 39.1 (1977), pp. 1–38.

[20] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer. "Speaker adaptation using constrained estimation of Gaussian mixtures". In: *IEEE Transactions on Speech and Audio Processing* 3.5 (1995), pp. 357–366.

[21] R. E. Donovan and E. M. Eide. "The IBM Trainable Speech Synthesis System". In: *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*. Sydney, Australia, 1998, pp. 1703–1706.

[22] W. U. Dressler and R. Wodak. "Sociophonological methods in the study of sociolinguistic variation in Viennese German". In: *Language in Society* 11 (1982), pp. 339–370.

[23] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[24] T. Dutoit. *An introduction to text-to-speech synthesis*. Norwell, MA, USA: Kluwer Academic Publishers, 1997.

[25] G. Fant. "Formants and Cavities". In: *Proceedings of the 5th International Congress on Phonetic Sciences*. Münster, Germany, 1965, pp. 120–141.

[26] D. Felps, C. Geng, and R. Gutierrez-Osuna. “Foreign Accent Conversion Through Concatenative Synthesis in the Articulatory Domain”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.8 (2012), pp. 2301–2312.

[27] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna. “Foreign Accent Conversion in Computer Assisted Pronunciation Training”. In: *Speech Communication* 51.10 (2009), pp. 920–932.

[28] J. Freng, B. Ramabhadran, J. Hansen, and J. D. Williams. “Trends in Speech and Language Processing [In the Spotlight]”. In: *Signal Processing Magazine, IEEE* 29.1 (2012), pp. 177–179.

[29] T. Fukada, Y. Komori, T. Aso, and Y. Ohora. “A Study On Pitch Pattern Generation Using Hmm-Based Statistical Information”. In: *Proceedings of the Third International Conference on Spoken Language Processing (ICSLP)*. 1994, pp. 723–726.

[30] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai. “An adaptive algorithm for mel-cepstral analysis of speech”. In: *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 1. San Francisco, CA, USA, 1992, pp. 137–140.

[31] J.-L. Gauvain and C.-H. Lee. “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains”. In: *IEEE Transactions on Speech and Audio Processing* 2.2 (1994), pp. 291–298.

[32] T. Gay. “Effect of speaking rate on vowel formant movements”. In: *Journal of the Acoustical Society of America* 63.1 (1978), pp. 223–230.

[33] L. Gillick and S. Cox. “Some statistical issues in the comparison of speech recognition algorithms”. In: *Proceedings of the 1989 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Glasgow, UK, 1989, pp. 532–535.

[34] N. Goel, S. Thomas, M. Agarwal, P. Akyazi, L. Burget, K. Feng, A. Ghoshal, O. Glembek, M. Karafiat, D. Povey, A. Rastrow, R. C. Rose, and P. Schwarz. “Approaches to automatic lexicon learning with limited training examples.” In: *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Dallas, Texas, USA, 2010, pp. 5094–5097.

[35] F. Goldman-Eisler. *Psycholinguistics: Experiments in spontaneous speech*. London: Academic Press, 1968.

[36] Google Inc. *Android NDK*. `https://developer.android.com/tools/sdk/ndk/`.

[37] R. Greisbach. “Reading aloud at maximal speed”. In: *Speech Communication* 11.4-5 (1992), pp. 469–473.

[38] L. He and A. Gupta. “Exploring benefits of non-linear time compression”. In: *Proceedings of the ninth ACM international conference on Multimedia*. ACM. Ottawa, Canada, 2001, pp. 382–391.

[39] J. Hershey, P. Olsen, and S. Rennie. “Variational Kullback-Leibler divergence for Hidden Markov Models”. In: *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, 2007. ASRU.* Kyoto, Japan, 2007, pp. 323–328.

[40] HTS working group. *hts-engine*. `http://hts-engine.sourceforge.net/`.

[41] HTS working group. *The HMM-based speech synthesis system (HTS)*. `http://www.cstr.ed.ac.uk/projects/festival/`.

[42] A. J. Hunt and A. W. Black. “Unit selection in a concatenative speech synthesis system using a large speech database”. In: *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 1. Atlanta, GA, USA, 1996, pp. 373–376.

[43] IEEE. “IEEE Recommended Practice for Speech Quality Measurement”. In: *IEEE Transactions on Audio and Electroacoustics* 17.3 (1969), pp. 225–246.

[44] S. Isard and D. Miller. “Diphone synthesis techniques”. In: *Proceedings of IEEE International Conference on Speech Input/Output*. 1986, pp. 77–82.

[45] E. Janse. “Word perception in fast speech: artificially time-compressed vs. naturally produced fast speech”. In: *Speech Communication* 42.2 (2004), pp. 155–173.

[46] E. Janse, S. Nooteboom, and H. Quené. “Word-level intelligibility of time-compressed speech: Prosodic and segmental factors”. In: *Speech Communication* 41.2 (2003), pp. 287–301.

[47] R. Karhila and M. Wester. “Rapid Adaptation of Foreign-accented HMM-based Speech Synthesis”. In: *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Florence, Italy, 2011, pp. 2801–2804.

[48] 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”. In: *Speech Communication* 27.3–4 (1999), pp. 187–207.

[49] M. Kurimo, W. Byrne, J. Dines, P. N. Garner, M. Gibson, Y. Guan, T. Hirsimäki, R. Karhila, S. King, H. Liang, K. Oura, L. Saheer, M. Shannon, S. Shiota, J. Tian, K. Tokuda, M. Wester, Y.-J. Wu, and J. Yamagishi. “Personalising speech-to-speech translation in the EMIME project”. In: *Proceedings of the ACL 2010 System Demonstrations*. Association for Computational Linguistics. Uppsala, Sweden, 2010, pp. 48–53.

[50] W. Labov. *Sociolinguistic patterns*. Oxford, Blackwell, 1972.

[51] J. Lebeter and S. Saunders. “The effects of time compression on the comprehension of natural and synthetic speech”. In: *Working Papers of the Linguistics Circle* 20.1 (2010), pp. 63–81.

[52] M. L. G. Lecumberri, R. Barra-Chicote, R. P. Ramón, J. Yamagishi, and M. Cooke. “Generating segmental foreign accent”. In: *Proceedings of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2014, pp. 1302–1306.

[53] L. Lee and R. Rose. “A frequency warping approach to speaker normalization”. English. In: *IEEE Transactions on Speech and Audio Processing* 6.1 (1998), pp. 49–60.

[54] C. J. Leggetter and P. C. Woodland. “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models”. In: *Computer Speech & Language* 9.2 (1995), pp. 171–185.

[55] H. Leung and V. W. Zue. “A procedure for automatic alignment of phonetic transcriptions with continuous speech”. In: *Proceedings of the 1984 IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP)*. Vol. 9. San Diego, CA, USA, 1984, pp. 73–76.

[56] S. Levinson. “Continuously variable duration hidden Markov models for automatic speech recognition”. In: *Computer Speech & Language* 1.1 (1986), pp. 29–45.

[57] H. Liang and J. Dines. “Phonological Knowledge Guided HMM State Mapping for Cross-Lingual Speaker Adaptation”. In: *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Florence, Italy, 2011, pp. 1825–1828.

[58] L. Loots and T. Niesler. “Automatic Conversion Between Pronunciations of Different English Accents”. In: *Speech Communication* 53.1 (2011), pp. 75–84.

[59] T. Masuko. “HMM-Based Speech Synthesis and Its Applications”. PhD thesis. Tokyo, Japan: Tokyo Institute of Technology, 2002.

[60] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai. “Speech synthesis using HMMs with dynamic features”. In: *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 1. Atlanta, GA, USA, 1996, pp. 389–392.

[61] Microsoft Corporation. *Visual Studio*. `https://www.visualstudio.com/`.

[62] D. Moers, I. Jauk, B. Möbius, and P. Wagner. “Synthesizing Fast Speech by Implementing Multi-Phone Units in Unit Selection Speech Synthesis”. In: *Proceedings of the 7th ISCA Workshop on Speech Synthesis (SSW)*. 2010, pp. 355–358.

[63] D. Moers, P. Wagner, B. Möbius, F. Müllers, and I. Jauk. “Integrating a Fast Speech Corpus in Unit Selection Speech Synthesis: Experiments on Perception, Segmentation, and Duration Prediction”. In: *Proceedings of Speech Prosody*. Vol. 100189:1-4. Chicago, USA, 2010.

[64] A. Moos and J. Trouvain. “Comprehension of Ultra-Fast Speech – Blind vs. ’Normally Hearing’ Persons”. In: *Proceedings of the International Congress of Phonetic Sciences*. Vol. 1. 2007, pp. 677–680.

[65] S. Moosmüller and R. Vollmann. “Natürliches Driften im Lautwandel: Die Monophthongierung im Österreichischen Deutsch”. In: *Zeitschrift für Sprachwissenschaft* 20/I (2001), pp. 42–65.

[66] S. Moosmüller. *Hochsprache und Dialekt in Österreich. Soziophonologische Untersuchungen zu ihrer Abgrenzung in Wien, Graz, Salzburg und Innsbruck*. Wien: Böhlau, 1991.

[67] S. Moosmüller. “Methodisches zur Bestimmung der Standardaussprache in Österreich”. In: *Standarddeutsch in Österreich – Theoretische und empirische Ansätze*. Ed. by M. Glauniger and A. Lenz. Vienna, Austria: Vandenhoeck & Ruprecht (Wiener Arbeiten zur Linguistik), 2015, pp. 163–182.

[68] S. Moosmüller and J. Brandstätter. “Phonotactic information in the temporal organization of Standard Austrian German and the Viennese dialect”. In: *Language Sciences* 46 (2014), pp. 84–95.

[69] S. Moosmüller, C. Schmid, and J. Brandstätter. “Standard Austrian German”. In: *Journal of the International Phonetic Association* 45/3 (2015), in print.

[70] M. Müller. “Dynamic Time Warping”. In: *Information Retrieval for Music and Motion*. Springer, 2007, pp. 69–84.

[71] F. Neubarth, B. Haddow, A. Hernandez-Huerta, and H. Trost. “A hybrid approach to statistical machine translation between standard and dialectal varieties”. In: *Proceedings of the 6th Language & Technology Conference (LTC)*. Poznan, Poland, 2013, pp. 414–418.

[72] F. Neubarth, M. Pucher, and C. Kranzler. “Modeling Austrian dialect varieties for TTS”. In: *Proceedings of the 9th Annual Conference of the International Speech Communication Association INTERSPEECH*. Brisbane, Australia, 2008, pp. 1877–1880.

[73] T. T. T. Nguyen, C. d’Alessandro, A. Rilliard, and D. D. Tran. “HMM-based TTS for Hanoi Vietnamese: issues in design and evaluation.” In: *Proceedings of the 14th Conference of the International Speech Communication Association (INTERSPEECH)*. Lyon, France, 2013, pp. 2311–2315.

[74] L. Nygaard and D. Pisoni. “Talker-specific learning in speech perception”. In: *Perception & Psychophysics* 60.3 (1998), pp. 355–376.

[75] J. J. Odell. “The Use of Context in Large Vocabulary Speech Recognition”. PhD thesis. Cambridge, UK: University of Cambridge, 1995.

[76] K. Oura, J. Yamagishi, M. Wester, S. King, and K. Tokuda. “Analysis of unsupervised cross-lingual speaker adaptation for HMM-based speech synthesis using KLD-based transform mapping”. In: *Speech Communication* 54.6 (2012), pp. 703–714.

[77] I. J. Pitt and A. D. N. Edwards. “Improving the Usability of Speech-based Interfaces for Blind Users”. In: *Proceedings of the Second Annual ACM Conference on Assistive Technologies*. Assets ’96. Vancouver, British Columbia, Canada: ACM, 1996, pp. 124–130.

[78] R. F. Port. “Linguistic timing factors in combination”. In: *Journal of the Acoustical Society of America* 69.1 (1981), pp. 262–274.

[79] M. Pucher, F. Neubarth, V. Strom, S. Moosmüller, G. Hofer, C. Kranzler, G. Schuchmann, and D. Schabus. “Resources for Speech Synthesis of Viennese Varieties”. In: *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*. Valletta, Malta, 2010, pp. 105–108.

[80] M. Pucher, D. Schabus, and J. Yamagishi. “Synthesis of fast speech with interpolation of adapted HSMMs and its evaluation by blind and sighted listeners”. In: *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Makuhari, Japan, 2010, pp. 2186–2189.

[81] M. Pucher, D. Schabus, J. Yamagishi, F. Neubarth, and V. Strom. “Modeling and interpolation of Austrian German and Viennese dialect in HMM-based speech synthesis”. In: *Speech Communication* 52.2 (2010), pp. 164–179.

[82] M. Pucher, M. Toman, D. Schabus, B. Zillinger, C. Valentini-Botinhao, and J. Yamagishi. “Influence of speaker familiarity on blind and visually impaired childrens perception of synthetic voices in audio games”. In: *Proceedings of the 16th Conference of the International Speech Communication Association (INTERSPEECH)*. Dresden, Germany, 2015, pp. 1625–1629.

[83] M. Pucher, V. Xhafa, A. Dika, and M. Toman. “Adaptive speech synthesis of Albanian dialects”. In: *Proceedings of the 18th International Conference of Text, Speech and Dialogue (TSD)*. Lecture Notes in Artificial Intelligence. Plzeň, Czech Republic, 2015, pp. 158–164.

[84] Y. Qian, J. Xu, and F. K. Soong. “A frame mapping based HMM approach to cross-lingual voice transformation”. In: *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Prague, Czech Republic, 2011, pp. 5120–5123.

[85] Y. Qian, Z.-J. Yan, Y.-J. Wu, F. K. Soong, X. Zhuang, and S. Kong. “An HMM trajectory tiling (HTT) approach to high quality TTS”. In: *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2010, pp. 422–425.

[86] L. R. Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition”. In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[87] L. Rabiner and R. Schafer. *Theory and Applications of Digital Speech Processing*. 1st. Upper Saddle River, NJ, USA: Prentice Hall Press, 2010.

[88] L. Rabiner, A. E. Rosenberg, and S. E. Levinson. “Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition”. In: *IEEE Trans. on Acoustics, Speech, and Signal Processing* 26 (1978), pp. 575–582.

[89] D. Reynolds. “Gaussian Mixture Models”. English. In: *Encyclopedia of Biometrics*. Ed. by S. Li and A. Jain. Springer US, 2009, pp. 659–663.

[90] K. Richmond, R. A. J. Clark, and S. Fitt. “On generating Combilex pronunciations via morphological analysis.” In: *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Makuhari, Japan, 2010, pp. 1974–1977.

[91] J. Rissanen. “Modeling by shortest data description”. In: *Automatica* 14.5 (1978), pp. 465–471.

[92] M. Ross, H. Shaffer, A. Cohen, R. Freudberg, and H. Manley. “Average magnitude difference function pitch extractor”. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 22.5 (1974), pp. 353–362.

[93] M. Russell, A. DeMarco, C. Veaux, and M. Najafian. *What’s Happening In Accents & Dialects? A Review Of The State Of The Art.* Presentation, UKSpeech Conference, Cambridge. Cambridge, UK, 2013.

[94] Y. Sagisaka, N. Kaiki, N. Iwahashi, and K. Mimura. “ATR ν-Talk Speech Synthesis System”. In: *Proceedings of the 2nd International Conference on Spoken Language Processing (ICSLP)*. Banff, AB, Canada, 1992, pp. 483–486.

[95] A. A. Sanderman and R. Collier. “Prosodic phrasing and comprehension”. In: *Language and Speech* 40.4 (1997), pp. 391–409.

[96] D. Schabus. “Audiovisual Speech Synthesis Based on Hidden Markov Models”. PhD thesis. Vienna University of Technology, 2015.

[97] D. Schabus, M. Pucher, and G. Hofer. “Joint Audiovisual Hidden Semi-Markov Model-based Speech Synthesis”. In: *IEEE Journal of Selected Topics in Signal Processing* 8.2 (2014), pp. 336–347.

[98] H. Scheutz. “Umgangssprache als Ergebnis von Konvergenz– und Divergenzprozessen zwischen Dialekt und Standardsprache.” In: *Dialektgenerationen, Dialektfunktionen, Sprachwandel*. Ed. by T. Stehl. Tübingen, Germany: Gunter Narr Verlag, 1999, pp. 105–131.

[99] K. Shinoda and T. Watanabe. “MDL-based context-dependent subword modeling for speech recognition”. In: *Journal of the Acoustical Society of Japan (E)* 21.2 (2000), pp. 79–86.

[100] B. Soukup and S. Moosmüller. “Standard language in Austria.” In: *Standard languages and language standards in a changing Europe*. Ed. by T. Kristiansen and N. Coupland. Oslo, Norway: Novus Press, 2011, pp. 39–46.

[101] A. Stan, O. Watts, Y. Mamiya, M. Giurgiu, R. Clark, J. Yamagishi, and S. King. “TUNDRA: a multilingual corpus of found data for TTS research created with light supervision”. In: *Proceedings of the 14th Conference of the International Speech Communication Association (INTERSPEECH)*. Lyon, France, 2013, pp. 2331–2335.

[102] A. Stent, A. Syrdal, and T. Mishra. “On the intelligibility of fast synthesized speech for individuals with early-onset blindness”. In: *Proceedings of the ACM SIGACCESS conference on computers and accessibility*. ACM. Dundee, UK, 2011, pp. 211–218.

[103] A. K. Syrdal, H. T. Bunnell, S. R. Hertz, T. Mishra, M. F. Spiegel, C. Bickley, D. Rekart, and M. J. Makashay. “Text-To-Speech Intelligibility Across Speech Rates.” In: *Proceedings of the 13th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Portland, USA, 2012, pp. 623–626.

[104] Y. Tabet and M. Boughazi. “Speech synthesis techniques. A survey”. In: *Proceedings of the 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA)*. 2011, pp. 67–70.

[105] M. Tachibana, J. Yamagishi, T. Masuko, and T. Kobayashi. “Speech Synthesis with Various Emotional Expressions and Speaking Styles by Style Interpolation and Morphing”. In: *IEICE Transactions on Information and Systems* E88-D.11 (2005), pp. 2484–2491.

[106] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayash. “Speaker Adaptation for HMM-based Speech Synthesis System Using MLLR”. In: *Proceedings of the 3rd ESCA/COCOSDA Workshop on Speech Synthesis (SSW)*. Jenolan, Australia, 1998, pp. 273–276.

[107] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. “Text-to-speech synthesis with arbitrary speaker’s voice from average voice”. In: *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH/INTERSPEECH)*. Aalborg, Denmark, 2001, pp. 345–348.

[108] P. Taylor. *Text-to-Speech Synthesis*. Cambridge, United Kingdoms: Cambridge University Press, 2009.

[109] K. Tokuda, T. Kobayashi, and S. Imai. “Speech parameter generation from HMM using dynamic features”. In: *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 1. Detroit, MI, USA, 1995, pp. 660–663.

[110] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi. “Hidden Markov models based on multi-space probability distribution for pitch pattern modeling”. In: *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 1. Phoenix, AZ, USA, 1999, pp. 229–232.

[111] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura. “Speech Synthesis Based on Hidden Markov Models”. In: *Proceedings of the IEEE* 101.5 (2013), pp. 1234–1252.

[112] M. Toman and M. Pucher. “An Open Source Speech Synthesis Frontend for HTS”. In: *Proceedings of the 18th International Conference of Text, Speech and Dialogue (TSD)*. Lecture Notes in Artificial Intelligence. Plzeň, Czech Republic, 2015, pp. 291–298.

[113] M. Toman and M. Pucher. “Evaluation of state mapping based foreign accent conversion”. In: *Proceedings of the 16th Conference of the International Speech Communication Association (INTERSPEECH)*. Dresden, Germany, 2015, pp. 304–308.

[114] M. Toman and M. Pucher. “Structural KLD for Cross-Variety Speaker Adaptation in HMM-based Speech Synthesis”. In: *Proceedings of the 10th IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA)*. Innsbruck, Austria, 2013, pp. 382–387.

[115] M. Toman, M. Pucher, S. Moosmüller, and D. Schabus. “Unsupervised and phonologically controlled interpolation of Austrian German language varieties for speech synthesis”. In: *Speech Communication* 72 (2015), pp. 176–193.

[116] M. Toman, M. Pucher, and D. Schabus. “Cross-variety speaker transformation in HSMM-based speech synthesis”. In: *Proceedings of the 8th ISCA Workshop on Speech Synthesis (SSW)*. Barcelona, Spain, 2013, pp. 77–81.

[117] M. Toman, M. Pucher, and D. Schabus. “Multi-variety adaptive acoustic modeling in HSMM-based speech synthesis”. In: *Proceedings of the 8th ISCA Workshop on Speech Synthesis (SSW)*. Barcelona, Spain, 2013, pp. 83–87.

[118] J. Trouvain. “On the comprehension of extremely fast synthetic speech”. In: *Saarland working papers in linguistics (SWPL)* 1 (2007), pp. 5–13.

[119] University of Edinburgh. *Festival*. `http://www.cstr.ed.ac.uk/projects/festival/`.

[120] C. Valentini-Botinhao, M. Toman, M. Pucher, D. Schabus, and J. Yamagishi. “Intelligibility Analysis of Fast Synthesized Speech”. In: *Proceedings of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Singapore, 2014, pp. 2922–2926.

[121] C. Valentini-Botinhao, M. Toman, M. Pucher, D. Schabus, and J. Yamagishi. “Intelligibility of time–compressed synthetic speech: compression method and speaking style”. In: *Speech Communication* 74 (2015), pp. 52–64.

[122] W. Verhelst and M. Roelands. “An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech”. In: *Proceedings of the 1993 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 2. 1993, pp. 554–557.

[123] A. J. Viterbi. “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm”. In: *IEEE Transactions on Information Theory* 13.2 (1967), pp. 260–269.

[124] O. Watts, A. Stan, R. Clark, Y. Mamiya, M. Giurgiu, J. Yamagishi, and S. King. “Unsupervised and lightly-supervised learning for rapid construction of TTS systems in multiple languages from ‘found’ data: evaluation and analysis”. In: *Proceedings of the 8th ISCA Workshop on Speech Synthesis (SSW)*. ISCA. Barcelona, Spain, 2013, pp. 101–106.

[125] M. Wester and R. Karhila. “Speaker similarity evaluation of foreign-accented speech synthesis using HMM-based speaker adaptation”. In: *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Prague, Czech Republic, 2011, pp. 5372–5375.

[126] M. Wester. “Cross-lingual talker discrimination”. In: *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Makuhari, Japan, 2010, pp. 1253–1256.

[127] Y.-J. Wu, S. King, and K. Tokuda. “Cross-Lingual Speaker Adaptation for HMM-Based Speech Synthesis”. English. In: *6th International Symposium on Chinese Spoken Language Processing*. IEEE, 2008, pp. 1–4.

[128] Y.-J. Wu, Y. Nankaku, and K. Tokuda. “State mapping based method for cross-lingual speaker adaptation in HMM-based speech synthesis”. In: *Proceedings of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Brighton, United Kingdom, 2009, pp. 528–531.

[129] C. Wutiwiwatchai, A. Thangthai, A. Chotimongkol, C. Hansakunbuntheung, and N. Thatphithakkul. “Accent level adjustment in bilingual Thai-English text-to-speech synthesis”. In: *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. 2011, pp. 295–299.

[130] C. Wutiwiwatchai, P. Cotsomrong, S. Suebvisai, and S. Kanokphara. “Phonetically Distributed Continuous Speech Corpus for Thai Language.” In: *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*. 2002, pp. 869–872.

[131] C. Wutiwiwatchai and S. Furui. “Thai speech processing technology: A review”. In: *Speech communication* 49.1 (2007), pp. 8–27.

[132] J. Yamagishi. “Average-Voice-Based Speech Synthesis”. PhD thesis. Tokyo, Japan: Tokyo Institute of Technology, 2006.

[133] J. Yamagishi and T. Kobayashi. “Average-Voice-Based Speech Synthesis Using HSMM-Based Speaker Adaptation and Adaptive Training”. In: *IEICE Transactions on Information and Systems* E90-D.2 (2007), pp. 533–543.

[134] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai. “Analysis of Speaker Adaptation Algorithms for HMM-Based Speech Synthesis and a Constrained SMAPLR Adaptation Algorithm”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.1 (2009), pp. 66–83.

[135] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi. “Modeling of various speaking styles and emotions for HMM-based speech synthesis”. In: *Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH)*. Geneva, Switzerland, 2003, pp. 2461–2464.

[136] J. Yamagishi, H. Zen, Y.-J. Wu, T. Toda, and K. Tokuda. “The HTS-2008 System: Yet Another Evaluation of the Speaker-Adaptive HMM-based Speech Synthesis System in The 2008 Blizzard Challenge”. In: *Proceedings of the Blizzard Challenge Workshop*. Brisbane, Australia, 2008.

[137] T. Yoshimura. “Simultaneous Modeling of Phonetic and Prosodic Parameters, and Characteristic Conversion for HMM-Based Text-to-Speech Systems”. PhD thesis. Nagoya, Japan: Nagoya Institute of Technology, 2002.

[138] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. “Duration modeling for HMM-based speech synthesis”. In: *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*. Sydney, Australia, 1998, pp. 29–32.

[139] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. “Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis”. In: *Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH)*. Budapest, Hungary, 1999, pp. 2374–2350.

[140] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. “Speaker interpolation for HMM-based speech synthesis system”. In: *Journal of the Acoustical Science of Japan (E)* 21.4 (2000), pp. 199–206.

[141] S. J. Young. “Cognitive User Interfaces”. In: *IEEE Signal Processing Magazine* 27.3 (2010), pp. 128–140.

[142] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. *The HTK Book (for HTK version 3.4)*. Cambridge, UK: Cambridge University Engineering Department, 2006.

[143] H. Zen. “Speaker and language adaptive training for HMM-based polyglot speech synthesis”. In: *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2010, pp. 410–413.

[144] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda. “The HMM-based Speech Synthesis System (HTS) Version 2.0”. In: *Proceedings of the 6th ISCA Workshop on Speech Synthesis (SSW)*. Bonn, Germany, 2007, pp. 294–299.

[145] H. Zen, A. Senior, and M. Schuster. “Statistical Parametric Speech Synthesis Using Deep Neural Networks”. In: *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 2013, pp. 7962–7966.

[146] H. Zen, K. Tokuda, and A. W. Black. “Statistical parametric speech synthesis”. In: *Speech Communication* 51.11 (2009), pp. 1039–1064.

[147] 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”. In: *Computer Speech & Language* 21.1 (2007), pp. 153–173.

[148] H. Zen, K. Tokuda, T. Masuko, T. Kobayasih, and T. Kitamura. “A hidden semi-Markov model-based speech synthesis system”. In: *IEICE Transactions on Information and Systems* E90-D.5 (2007), pp. 825–834.

Curriculum Vitae

Markus Toman (Dipl.-Ing./MSc)
born May 27, 1983 in Vienna

mail: m.toman@neuratec.com

Work experience

since 2012	Telecommunications Research Center Vienna, Researcher *Research in adaptive speech synthesis for blind and visually impaired users as well as acoustic modeling, transformation and interpolation of language varieties.* *Web- and native mobile development in various industry projects.*
2004-2012	Freelancer, Software Developer *Design and development of embedded and distributed software for network and energy management; medical software and eGovernment.*
2006	Vienna University of Technology, Tutor *Assisting lab course „Distributed Systems" at the Distributed Systems Group, Institute of Information Systems*
2003-2004	Austrian Red Crosss, Wr. Neustadt, Medic *Working as medic in civil service.*

Education

since 2012	Vienna University of Technology PhD studies, Institute of Software Technology and Interactive Systems
2009-2012	Medical University Vienna Master program „Medical Informatics" *Specialization: Artificial intelligence, Medical image processing*
2005-2009	Vienna University of Technology Bachelor program „Medical Informatics"
1997-2002	Higher Technical Federal Education and Research Institute, Wr. Neustadt Electronic Data Processing and Organization