

BOSTON UNIVERSITY  
COLLEGE OF ENGINEERING

DISSERTATION

SPEECH RECOGNITION SYSTEM DESIGN BASED ON  
AUTOMATICALLY DERIVED UNITS

BY

MICHIEL A. U. BACCHIANI

ir., Technische Universiteit Eindhoven, 1994

Submitted in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy

1999

Approved by

First Reader

---

Dr. Mari Ostendorf, Associate Professor  
Department of Electrical and Computer Engineering  
Boston University

Second Reader

---

Dr. W. Clem Karl, Assistant Professor  
Department of Electrical and Computer Engineering  
Boston University

Third Reader

---

Dr. Hamid Nawab, Associate Professor  
Department of Electrical and Computer Engineering  
Boston University

Fourth Reader

---

Dr. Carol Espy-Wilson, Assistant Professor  
Department of Electrical and Computer Engineering  
Boston University

## Acknowledgments

First and foremost, I would like to thank my advisor, Mari Ostendorf. Even after spending several years working in the lab with her, her commitment to quality of writing and research as well as her commitment to her students still astonish me. Although continuously submerged in a sea of work, I was able to get her attention at any time without a lot of effort. In addition, I am very grateful for all the kind and thoughtful gestures she made towards me throughout my stay in her lab. Although continuously striving for the best possible work, I never found it mixed with a lack of attention to the “personal situation” and that is I think the key that made me feel so at home in her lab.

Second I would like to thank my many friends at ATR and in particular Yoshinori Sagisaka. His continuous support were a very important factor in realizing the work described in this thesis. Not only did he create the opportunity for this work by persuading ATR to providing a grant to the group, also technically his input was very valuable.

Third, my many interactions with Kuldeep Paliwal have given me much more insight in the ideas described here. His previous work experience in this particular topic together with his great enthusiasm for the field in general were a true motivation to continue to work on this topic.

Fourth I am very grateful to the many SPI lab members that overlapped with my stay. Ashvin Kannan for the many in-depth discussion on good programming and experiment design. Rukmini Iyer for the many technical as well as music related conversations. Izhak Shaik for giving me in-depth feedback on my code together with very useful comments on how to improve it. Becky Bates for her endless stories about the exciting world of ice-skating. Cameron Fordyce for delightful lunch conversations about both work and American-European differences. Randy Fish making life so much more enjoyable with his high spirits and witty conversations. David Palmer for his seemingly endless supply of jokes as well as giving me more understanding

of language processing beyond the speech world. Hariharan Shivakumar for being himself.

A very special thanks goes out to our system administrator, Justin Hahn who has served far beyond the call of duty on many occasions and without whom, this thesis could not have been completed at this time. Our joint interest in the wonders of UNIX system administration together with his continuous desire to do things right and joy doing so has taught me many new skills.

Another special mention goes out to our secretary Loretta Hawkes for helping me out of so many administrative tough spots. Seemingly independent of the number of things to organize, she always helped me without delay getting me all the required forms with the details on how to fill them out. For somebody like me, who somehow always ends up in chaos with last minute solutions, her organization skill were absolutely flabbergasting.

Finally I would like to thank my wife Yuriko for being so patient and supportive, never blaming me for spoiling so many of her weekends by going to work. And let it also be said that I could have finished this work at least a year earlier hadn't it been for our over energetic son Ray who continuously keeps us in his grip with his seductive smile leaving us no option but to enjoy playing with him.

SPEECH RECOGNITION SYSTEM DESIGN BASED ON  
AUTOMATICALLY DERIVED UNITS

(Order No.      )

MICHIEL A. U. BACCHIANI

Boston University, College of Engineering, 1999

Major Professor: Mari Ostendorf      Professor of: Electrical Engineering

ABSTRACT

In most speech recognition systems today, acoustic modeling and lexical modeling are viewed as separable problems. Currently the most popular approach is to manually define canonical word pronunciations in terms of phonetic units and let the acoustic models capture differences between actual spoken and canonical pronunciations implicitly with Gaussian mixture models. As a result, these models can be very broad, particularly for casual spontaneous speech. An alternative approach, explored in this thesis, is to learn a unit inventory and pronunciation dictionary from training data using a maximum likelihood objective function.

In particular, this thesis addresses previously unsolved problems in automatic unit design with three main contributions. First, to make design of a large unit inventory practical, a new approach is described that combines the problems of unit selection and lexicon design. The design of the units is acoustically driven but constrained to guarantee a matched, limited complexity pronunciation model. Instead of using an acoustic unit training algorithm followed by separate pronunciation model design, the algorithm proposed here incorporates a pronunciation constraint within the unit design algorithm. The resulting unit inventory, unit models and lexicon are matched since they are designed by a single joint design step. The second problem addressed involves synthesizing models for unobserved contexts, needed to model contextual variation at word boundaries. As in phone-based systems, decision tree clustering is

used, but this requires classes or sets of units that have a similar influence in context. The solution is to learn these classes from data by a parallel context clustering process. Third, the ability to generalize at the word-level, i.e. to handle words not observed in the training data, is provided by a hybrid system design algorithm. In the hybrid system, automatically derived units are designed for the most frequent words, and phonetic units are designed for all words in the vocabulary. Using an estimation step, the word models constructed by the independent automatic and phonetic units are evaluated and the most likely model is included in the lexicon.

The new automatic unit design algorithm showed improved performance over phonetic units in experiments on a medium vocabulary (1000 words) task (Resource Management) for both small and large unit inventory systems, outperforming an alternative approach to automatic unit design reported on this task. The algorithm for learning context conditioning groups is successful in that the performance of a system derived by decision tree clustering is equivalent to that of the best unconstrained clustering system and an additional gain is observed when modeling contextual effects across word-boundaries. Finally, when automatically derived units were used in experiments on a large vocabulary (20,000 word) conversational speech task (Switchboard), the recognition accuracy improved over the phonetic unit baseline. In summary, the joint unit and lexicon design algorithm gives higher recognition performance or can be configured to give similar performance at lower cost (lower system complexity) than phone-based units for applications where several examples of each vocabulary word can be provided.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>8</b>
2.1	Acoustic Features . . . . .	8
2.2	Statistical Model for Speech Recognition . . . . .	12
2.2.1	Acoustic Model . . . . .	14
2.2.2	Sub-Word Units . . . . .	16
2.2.3	Acoustic Model Parameter Estimation . . . . .	23
2.3	Search . . . . .	27
<b>3</b>	<b>Automatically Derived Units</b>	<b>32</b>
3.1	Acoustic Sub-Word Units . . . . .	34
3.1.1	Unsupervised Learning of Automatic Units . . . . .	34
3.1.2	Pronunciation Modeling . . . . .	35
3.2	Joint Unit and Lexicon Design . . . . .	39
3.2.1	Initial Unit Inventory and Lexicon Design . . . . .	40
3.2.2	Re-training . . . . .	45
3.2.3	Progressive Refinement . . . . .	45
3.3	Experiments . . . . .	48
3.3.1	Phone systems . . . . .	49
3.3.2	ASWU systems . . . . .	50

3.4	Summary and Conclusions . . . . .	60
<b>4</b>	<b>Explicit Context Modeling</b>	<b>63</b>
4.1	General Clustering Methods . . . . .	64
4.2	Defining Long Distance Context . . . . .	66
4.3	Learning Context Classes . . . . .	69
4.4	Experimental Results . . . . .	73
4.4.1	Local Context Experiments . . . . .	74
4.4.2	Distant Context Experiments . . . . .	78
4.4.3	Cross-Word Context Modeling . . . . .	80
4.5	Summary and Conclusions . . . . .	81
<b>5</b>	<b>Use of ASWUs in a Large Vocabulary System</b>	<b>84</b>
5.1	Vocal Tract Length Normalization . . . . .	85
5.1.1	Frequency warping . . . . .	87
5.1.2	Formant Based Warp Estimation . . . . .	89
5.1.3	Maximum Likelihood Warp Estimation . . . . .	90
5.1.4	Normalizing Test Speakers . . . . .	92
5.2	Phone-Based Sub-System Design . . . . .	93
5.3	Hybrid System Design . . . . .	96
5.4	Experiments . . . . .	97
5.4.1	Feature Normalization Experiments . . . . .	98
5.4.2	The Switchboard Corpus . . . . .	101
5.4.3	Phone-based System Results . . . . .	105
5.4.4	Hybrid System Experiments . . . . .	110
5.5	Summary and Conclusions . . . . .	114
<b>6</b>	<b>Conclusions</b>	<b>116</b>
6.1	Summary and Contributions . . . . .	117
6.1.1	Large Automatically Derived Unit Inventories . . . . .	118

6.1.2	Explicit Context Modeling . . . . .	122
6.1.3	Extension to Large Vocabulary Systems . . . . .	125
6.2	Future Work . . . . .	126

# List of Tables

3.1	HMM system results (% accuracy) on the February 1989 and full test sets for phone-based different system configurations. . . . .	50
3.2	Recognition results of the CI-phone baseline and low complexity ASWU systems using different approaches to temporal adjustment. . . . .	52
3.3	Recognition performance on the full test set of the 3 systems derived by different progressive refinement approaches. . . . .	58
3.4	Pronunciation examples of words with the same morpheme base. . . . .	58
4.1	Best case results for the explicit local context systems, which can be compared to 90.1% accuracy for the implicit context modeling with 1385 distributions. . . . .	79
5.1	Median formant locations of the first, second and third formants in the training part of the TIMIT corpus. . . . .	99
5.2	Classification improvements due to speaker normalization using a formant-based approach. . . . .	99
5.3	Recognition accuracy of the word-internal system on the Switchboard test set using either phonetic units alone or a hybrid system. The features are vocal tract length normalized mel-scale cepstral coefficients.112	

# List of Figures

2.1	Mel-frequency cepstral feature computation block diagram. . . . .	12
2.2	Statistically-based speech recognition system block diagram. . . . .	13
2.3	Schematic representation of a 3-state Hidden Markov Model and a possible sequence of generated observations. . . . .	15
3.1	Recognition performance on the February 1989 test set using unit inventories and lexicons derived at different stages of progressive refinement using “early” binary temporal adjustment. The square in the figure indicates the stage where binary temporal adjustment was performed. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training. . . . .	54
3.2	Recognition performance on the February 1989 test set using unit inventories and lexicons derived at different stages of progressive refinement using “late” temporal adjustment. The square in the figure indicates the stage where binary temporal adjustment was performed. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training. . . . .	55

3.3	Recognition performance on the February 1989 test set using unit inventories and lexicons derived at different stages of progressive refinement using a variable temporal adjustment approach. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training. . . . .	57
3.4	Range of the number of ASWU units that map to the different phones when the average is 3.9 states/phone. A (*) indicates the median number of units per phone and the bars indicate the range of one standard deviation about the mean. Phone labels use the DARPAbet standard, except that /-d/ indicates an unreleased closure and /ts/ is a /t/-/s/ sequence. . . . .	59
4.1	A possible alignment of observations (feature vectors depicted as rectangular boxes) with a hierarchical and non-deterministic coarse and fine level unit sequence. . . . .	68
4.2	Clustering unique contexts in two context groups. Shaded squares correspond to observed contexts. . . . .	71
4.3	Recognition performance of the shared distribution, local word-internal, tri-unit unit inventory starting from a low complexity (124 unit) base unit inventory. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training. . . . .	75
4.4	Recognition performance of the shared distribution, local word-internal tri-unit inventory starting from a high complexity (635 unit) base unit inventory. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training. . . . .	77

4.5	Recognition performance of the shared distribution, local word-internal, quin-unit inventory starting from a high complexity (635 unit) base unit inventory. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training. . . . .	78
4.6	Recognition performance of the shared distribution, distant word-internal context inventory based on a high complexity (743 unit) unit inventory. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training. . . . .	80
5.1	A simple tube model of the vocal tract. . . . .	87
5.2	The Helmholtz resonator model of the vocal tract. . . . .	88
5.3	Piecewise linear frequency warping function. . . . .	88
5.4	Equally spaced sampling of the warped frequency axis using spectrum estimation by linear interpolation with a warp factor $\alpha < 1.0$ . . . . .	89
5.5	Warp mixture model training overview. . . . .	91
5.6	Acoustic segmentation of the data for the purpose of training the likelihood-based warp estimation model. . . . .	92
5.7	Test speaker warp factor estimation using a likelihood approach. . . . .	93
5.8	F3-based warp factor histograms for training and test sets. . . . .	100
5.9	Warp factor histogram of training and test speakers using the mixture model estimated after 5 iterations. . . . .	102
5.10	Classification scores and data likelihoods using the different mixture models estimated after each training iteration step. . . . .	103
5.11	ML estimate of the warping factor for 4 different speakers as a function of the amount of data used in estimation. . . . .	104
5.12	Warp factor histogram of training data using the male gender dependent mixture model estimated after iteration 4. . . . .	106

5.13	Warp factor histogram of training data using the female gender dependent mixture model estimated after iteration 4. . . . .	106
5.14	Warp factor histogram of all data using the male gender dependent mixture model estimated after iteration 4. . . . .	107
5.15	Warp factor histogram of all data using the female gender dependent mixture model estimated after iteration 4. . . . .	107
5.16	Data likelihoods of the training data given the model estimated after the different training iterations. . . . .	108
5.17	Average likelihood per frame of training data for the male word-internal phonetic-unit system during the mixture splitting and EM re-estimation process. . . . .	109
5.18	Average likelihood per frame of training data for the female word-internal phonetic-unit system during the mixture splitting and EM re-estimation process. . . . .	109
5.19	Average likelihood per frame of training data for the male phonetic-unit system during the mixture splitting and EM re-estimation process. . . . .	111
5.20	Average likelihood per frame of training data for the female phonetic-unit system during the mixture splitting and EM re-estimation process. . . . .	111
5.21	Average likelihood per frame of training data for the male word-internal ASWU system during the mixture splitting and EM re-estimation process. . . . .	113
5.22	Average likelihood per frame of training data for the female word-internal ASWU system during the mixture splitting and EM re-estimation process. . . . .	113
5.23	Histogram of estimated probabilities of the automatic unit pronunciations. . . . .	114

# Chapter 1

## Introduction

Over the past two decades, automatic speech recognition has matured from being capable of recognizing only a few isolated words from one particular speaker to recognition of tens of thousands of words spoken in a continuous fashion by any speaker. The developed techniques have proved to be very effective provided that the recording conditions and the speaking style remains controlled. Word error rates of systems recognizing continuously spoken speech with a vocabulary of 64,000 words are on the order of 10% when the speech is read from newspaper text and the speaker dialect variation is limited [51]. However, if the speaking style is varied, allowing spontaneous speech, the systems perform poorly, as the algorithms have been developed and tuned for a read speaking style. Many applications call for removing the style constraint because users are unable to provide speech in the required speaking style. An example of an application that requires robustness towards different speaking styles is the transcription of human-to-human dialogues. Another example is a system intended to provide a service for speakers from a wide geographical area which will have to be able to deal with many different accents and dialects. In both these cases, a wide variety of speakers with varying speaking styles is inherent to the task and requiring the users to adapt their speaking style is an unrealistic or even impossible requirement.

Most current speech recognition systems use a statistical approach. In this frame-

work, the recognition system relies on two model components: a language model, describing how likely it is to observe a particular sequence of words, and an acoustic model, describing how likely it is that the observed acoustic evidence was originating from that word sequence. To limit the number of free parameters of the system, the acoustic model for a word sequence will generally be derived by composition of smaller sub-word unit models. The acoustic model used in a system will consist of two parts: a set of unit models and a dictionary that describes how to compose a word model from the unit models. The most common choice of a unit is the phone (from the speech sound inventory used in standard pronunciation dictionaries) for which hand-crafted dictionaries are available. The models describe the acoustic evidence corresponding to the sub-word units, and each sub-word unit model will appear as a part of the word model for multiple words.

Estimates of the distributions describing the acoustic observations corresponding to a unit are obtained from a set of training data that is representative of the recognition task the system is designed for. The set of units and the dictionary describing the unit model composition are generally not learned from data however and are hand crafted. Even though a unit inventory and dictionary derived this way are unlikely to be optimal for the recognition task, on read speech tasks (fixed speaking style) the parameter estimation techniques have proved to be powerful enough to overcome this suboptimal choice. In a read speech task, the increased acoustic variability due to the suboptimal unit inventory and dictionary design process can still be captured within the unit model distributions without introducing too much overlap of the distributions for different sub-word units. For spontaneously spoken speech however, the increased variability makes this approach problematic. Unlike in carefully read speech, the pronunciations in spontaneous speech often deviate from the citation form pronunciations found in a standard dictionary. This deviation could for example be due to a spontaneous speaking style (e.g. using “gonna” instead of “going to”) or due to unfamiliar words (e.g. names). Forcing the system to use canonical pronun-

ciations from a dictionary in such applications will require that additional acoustic variability is represented by the unit models since the mapping between the spoken phone sequence and model unit sequence is less consistent. The result is more overlap between distributions, which will likely reduce recognition accuracy. An algorithm for automatic pronunciation design can find a model that better fits the data. In the work presented here, the acoustic variability that is to be captured within the unit models is reduced by automatically deriving a unit inventory and its unit models together with the corresponding dictionary using a maximum likelihood objective function. The use of the objective function in the unit inventory and lexicon design will yield unit distributions that need to capture less acoustic variability leading to less overlap of unit distributions and presumably higher recognition accuracy.

The automatic nature of the unit and lexicon design provide a low-cost approach to derivation of high quality acoustic models in applications where several examples of a lexical item can be provided. Automatic derivation of pronunciations is of interest generally in applications where hand design of phonetic pronunciations are costly or even impossible, such as where no prior knowledge about appropriate pronunciations is available. However, even for applications where phonetically based units work well, automatic derivation of units is useful when the size of the system is important. As the algorithm generally provides a better (in the likelihood sense) set of units and corresponding lexicon, low complexity systems designed automatically perform significantly better than phonetic-unit-based systems of similar complexity.

Previous work has investigated the use of an automatically learned unit inventory and lexicon but has always approached these as separable problems[63, 50, 64]. Some work focussed on designing units; other work focussed on designing an appropriate dictionary given the units. As the two problems are related, independent design that disregards the relation between the two will result in a mismatched condition. This is especially true for a speaker-independent system as unconstrained unit inventory designed using data from multiple speakers tends to focus on acoustic differences due

to speaker identity rather than those that discriminate words. Conversely, forcing the use of an independently designed unit inventory within a fixed pronunciation structure will result in mismatched model estimates for the specified pronunciations.

Recent work by Holter and Svendsen [29] addressed the mismatch problem by taking an iterative unit inventory and lexicon design approach to arrive at a matched condition. As in previous work, the dictionary design process uses candidate pronunciations seen across examples in the training data in the unit design phase and uses likelihood to guide the search for the most suitable candidate. However, their approach becomes problematic when designing a large unit inventory, in which case the number of candidate pronunciations increases, reducing the chance of finding the optimal pronunciation among the examples and making the search computationally expensive. Use of a large unit inventory is required for a large vocabulary task in order to get the required modeling detail to distinguish each entry in the vocabulary.

Other problems, not addressed in earlier work, are related to extending the automatic unit framework to be suitable for use in a large vocabulary system. In a system based on phonetic units, acoustic modeling detail can be improved by explicitly representing local phonetic context. Modeling accuracy improves when only considering contextual effects of units within a word, but an important additional gain can be obtained by considering context effects from neighboring words. Techniques developed for explicit modeling of phonetic units in context do not apply to automatic units directly. As a consequence of the finite amount of training data, there is no guarantee that there will be sufficient examples of all units in all possible contexts, particularly cross-word contexts. In phonetic unit modeling, the knowledge of phonetics allows the definition of unit groups that are likely to have similar contextual effects and this knowledge can be used to guide a synthesis process for unobserved models. In the case of automatically derived units, no such knowledge is available, so the extension to modeling cross-word contextual effects is not straightforward.

Another problem that occurs when trying to use automatically derived units in a

large vocabulary system is that in order to reliably determine the pronunciation of a word in terms of the automatic units, a sufficient number of examples is required. Even without defining what a sufficient number of examples is, it is clear that having no or very few examples of a word will prohibit reliable pronunciation design. On the other hand, it is unrealistic to require a sufficient number of examples for each entry in the vocabulary of the system. In many large vocabulary recognition applications, the training data involves natural utterances or sentences and therefore has a very unbalanced word distribution. In order to make use of automatically derived units in a large vocabulary system, a mechanism should exist for dealing with low frequency words.

The work presented here addresses the use of automatically derived units in a large vocabulary setting by providing solutions to three previously unsolved problems: how to design a large unit inventory, how to deal with unseen events at the unit level (i.e. how to provide a model for a unit in an infrequently observed context) and how to handle infrequent events at the word-level (i.e. how to provide a model for an infrequently observed word). First a computationally efficient algorithm was developed to allow the design of a large unit inventory. The algorithm is particularly well suited to the problem of designing units on large corpora. Experimental results showed that the algorithm is successful in deriving both large and small unit inventories. Second, several approaches for modeling unit context were explored. Since part of the system design is the derivation of an appropriate dictionary from data, within-word context dependency can be learned implicitly. In addition, several approaches to explicit context modeling, analogous to that used in phonetic modeling, have been explored. In particular, a solution for extension of the explicit context modeling framework to incorporate cross-word effects when using automatic units has been developed. Unit groups that have equivalent contextual effects are learned from data and those groups can subsequently be used, as in the phonetic unit framework, to aid in the synthesis of models for units in unobserved contexts. Third,

a hybrid system design algorithm was developed, providing a mechanism to incorporate automatic units in a large vocabulary system. The hybrid system uses phonetic units to provide the desired generalization to infrequent words, and automatic units to provide a more detailed acoustic model for those entries with a sufficiently large number of examples. The new hybrid design algorithm uses a word-based evaluation as to whether or not the automatic unit design resulted in a better fit to the data compared to the word models built from phonetic units.

To allow comparisons with recent automatic unit design work from other sites and to limit the turn-around time of experiments, the initial algorithm design was tested on a small sized corpus (Resource Management). Phonetic unit-based systems achieve good performance (around 90% accuracy) on this 1000-word vocabulary, read speech corpus and the viability of the proposed algorithm could be tested by making a comparison of phonetically based systems with systems based on automatically derived units. Experiments confirmed that at low complexity, the automatic unit system outperforms a phonetic unit-based system (due to a better choice of unit inventory and lexicon) and that the performance gain diminishes at higher complexity (the larger number of free parameters of the system are able to model the acoustic variability even though increased by a suboptimal choice of a unit inventory). Later work involved incorporating the automatic units in a large vocabulary system. For these experiments, the Switchboard spontaneous human-to-human dialogue corpus was used. These experiments showed performance improvements over the phonetic baseline system confirming the benefits of the units and lexicon designed with a maximum likelihood objective for a system applied to speech with varying speaking styles.

In summary, the work presented here is aimed at using automatically derived units in speech recognition to improve accuracy of the acoustic model. The main focus of the work is to make practical the use of automatically derived units in speaker-independent large vocabulary tasks by providing solutions to the problems that these

units have in comparison to phonetic units. In particular, the inability to generalize both at the unit level (modeling explicit context), and at the word level (providing models for words infrequently seen in the training data) is addressed, as well as the computational complexity of the design process. Experiments applying these units in both small as well as large vocabulary applications show the benefits of using automatic units.

The rest of this thesis is organized as follows. In chapter 2, the most popular approach to speech recognition is described, providing background on the acoustic features that are used in the system as well as how the recognition problem can be cast in a statistical framework. The main focus of the discussion is on the acoustic model, as this is the focus of this thesis. In chapter 3 the proposed algorithm for automatic unit design is described. This chapter describes different approaches to derivation of a high complexity system and investigates how different parameter choices impact the performance of the resulting systems. Because of the success of explicitly modeling local context in the phonetic unit framework, several approaches to explicit modeling of context of automatic units are developed and assessed in chapter 4. Extensions that allow incorporation of the automatically derived units in a large vocabulary system are described in chapter 5. Finally, in chapter 6, the results presented in this thesis and possible future directions are discussed, including applications of the unit design algorithm to other areas such as speech synthesis and extensions of the unit design algorithm to problems of multiple pronunciation modeling and multi-pass search.

# Chapter 2

## Background

A speech recognizer attempts to automatically find the orthographic symbol sequence corresponding to a given speech signal. Generally, systems will rely on statistical models of the acoustics of the speech signal and of the language to perform this task. This chapter provides some explanation of the terminology used in the rest of this thesis and is intended as clarification and to provide literature references to previous work on the described topics. Section 2.1 will describe the acoustic cues commonly used in automatic speech recognition. Section 2.2 provides an explanation of how statistics are used for modeling speech. Finally, the process of searching for the most likely hypothesis given a sequence of acoustic observations and the statistical model is described in section 2.3. As the topic of the work described here is to provide a more accurate model of the acoustics, most emphasis will be placed on describing previous work on that part of the system.

### 2.1 Acoustic Features

The goal of the initial processing step of the speech signal is to reduce the representation of the signal to a more compact form (making subsequent processing steps computationally less expensive) and to transform the signal representation to a space

that increases the separability of the classes to be recognized (e.g. words). The feature processing step will take a digitized form of the pressure waveform corresponding to the speech signal and provide a corresponding sequence of feature vectors. It is essential that this processing step reduces the size of the signal representation in such a way that it retains the information relevant for the recognition process.

The speech signal can be considered as a slowly varying signal in the sense that it is fairly stationary within a sufficiently short interval (between 5 and 100 ms). On a longer time-scale, the signal characteristics vary reflecting the speech sounds that are spoken. On an even longer time-scale, the characteristics vary reflecting channel and speaker dependent variations. The signal processing step should therefore attempt to retain the slowly (but not very slowly) varying characteristics of the signal and discard the quickly varying detail. Using this fact, most systems will produce a real valued feature vector of dimensionality of about 15, commonly referred to as a frame, at a rate of approximately 100 to 200 frames per second. The actual stream of features used in the acoustic model are generally composed of this feature vector and its derivatives and double derivatives. The feature vectors are computed from a frame of speech (of typically 25 ms), obtained by multiplication of a window function, centered around the frame, with the speech signal. To avoid spectral artifacts which occur when using a rectangular window, a window function without discontinuities is generally used (e.g. a Hamming window). As the speech signal within the duration of a single frame is generally so limited that reliable estimation of the localized spectral contents becomes difficult, feature vectors are generally computed from a window with a duration that exceeds the frame duration. To retain the desired temporal granularity defined by the frame rate, overlapping windowed speech segments are used.

The type of signal processing employed in most contemporary speech recognition systems is **cepstral analysis**. The approach is based on a commonly used model for speech production. In this model, the speech signal is considered to be generated by

a system that excites a slowly time varying filter (the vocal tract) using one of two types of sources which both generate a rapidly varying signal. One source provides a periodic impulse train, the other a white noise signal, depending on whether or not the vocal cords are vibrating. If the system is excited by vibrating vocal cords the speech is said to be **voiced**. The magnitude of the (relatively) slowly varying filter that shapes the spectral characteristics of the resulting signal is the desired representation of the spoken speech sounds. Another equivalent interpretation of the task of signal processing is therefore to separate source and filter, which is more generally referred to as **homomorphic** processing. In this approach, the first step is to determine the spectral contents of the signal by a Fourier transformation or by estimation of a parametric model. The convolution of the signal source  $s$  and shaping filter  $h$  (i.e. the slowly varying filter that shapes the spectral characteristics of the signal) will, in the spectral domain, be represented by a product of the spectral representation of the source and filter components,

$$s * h \xrightarrow{\text{FFT}} SH. \quad (2.1)$$

Then by taking the logarithm of the magnitude of this spectral representation, the source and filter components will appear as a summation,

$$SH \xrightarrow{\log(\cdot)} \log(|S|) + \log(|H|). \quad (2.2)$$

Considering the spectral domain representation as a signal itself and taking into account the bimodal nature of the source signal, the source will either appear in the spectral domain as a rapidly varying (high frequency) component corresponding to the voiced excitation or as an approximately constant (low frequency) component for the unvoiced excitation. The spectral characteristic of the shaping filter will appear as the spectral envelope. Given this spectral difference between the source and filter, another transformation can be used to provide the desired separation of source and filter. Taking the inverse Fourier transform of the log magnitude spectrum will provide the cepstral representation of the signal in which the lower coefficients provide

a representation of the spectral envelope of the signal, representative of the shaping filter and higher coefficients representing the voiced source. This representation of the vocal tract filter provides a compact representation of the speech signal, retaining the spectral characteristics that differentiate the basic speech sounds.

From a physics perspective, the speech production process can be seen as a source generating a pressure waveform which is sent through a channel before being observed. The channel itself can be approximated by a stylized cavity such as, for example, a straight tube which is a reasonable model for the schwa vowel (like the *a* in *about*). Using such an approximate model allows for analytical solutions of locations of spectral peaks corresponding to the resonant frequencies of the formed cavity. These resonant frequencies are referred to as **formants**. As the locations of the formants are dependent on the dimensions of the cavity representing the vocal tract, differences in these dimensions between speakers will lead to speaker-dependent spectral differences. In particular, the average length of the vocal tract of a male speaker will be larger than that of a female speaker, generally causing the formants of a female speaker to be shifted upwards in frequency relative to a male speaker. For a speech recognition system that is to be applied to speech from a variety of speakers, the system should provide robustness to such speaker differences. As such variability in spectral characteristics can possibly lead to confusability between speech sounds, many systems apply mechanisms to reduce this variability either by making speaker-dependent modification to the feature processing [3, 22, 39, 66, 75] and/or by automatic, rapid adaptation of the parameters of the pattern classification system to match more closely with the spectral characteristics of a speaker [40, 21, 26, 38].

In the discussion above, no attempts were made to emphasize the salience of certain spectral regions in terms of the information content for performing the task of recognition. Using the results from psychophysical studies [61], it was found that human perception of frequency content does not follow a linear scale. Subsequent speech recognition research [17] has shown that representing spectral content along

a non-linear scale also improved automatic recognition performance in comparison to representing spectral content along a linear scale. The most frequently used non-linear frequency scale in contemporary speech recognition systems is the mel-scale. One example of simulating the non-linear perceptual frequency scale is by application of a filter bank of overlapping filters equally spaced along the non-linear scale with each filter having a constant bandwidth along the non-linear scale. The resulting mel-frequency cepstral coefficients are obtained by the inverse Fourier transformation of the outputs from the filters in the filter-bank. A block diagram of the feature extraction process is depicted in figure 2.1.

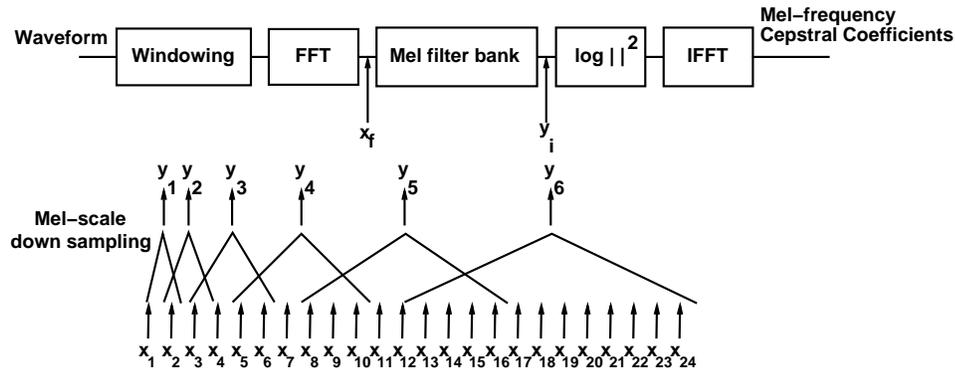


Figure 2.1: Mel-frequency cepstral feature computation block diagram.

## 2.2 Statistical Model for Speech Recognition

Using a statistical framework, the problem of speech recognition can be seen as estimating which orthographic representation is most likely given an acoustic representation of an utterance. In other words, which orthographic representation has the largest **posterior** probability;

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W | Y), \quad (2.3)$$

where  $\mathbf{W}$  denotes the symbol sequence making up the orthographic representation<sup>1</sup> and  $\mathbf{Y}$  is the representation of the acoustics (generally a sequence of feature vectors). Using Bayes' Rule the problem definition can be written as,

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}} P(\mathbf{Y} | \mathbf{W})P(\mathbf{W}) \quad (2.4)$$

where the model describing  $P(\mathbf{Y} | \mathbf{W})$  is referred to as the **acoustic model** and will be the focus of this thesis. The model describing  $P(\mathbf{W})$  is referred to as the **language model**. The parameters of both these models are estimated from **training data** which is representative of the recognition task the system is intended for. A schematic diagram of the components of a statistically-based speech recognition system is depicted in figure 2.2.

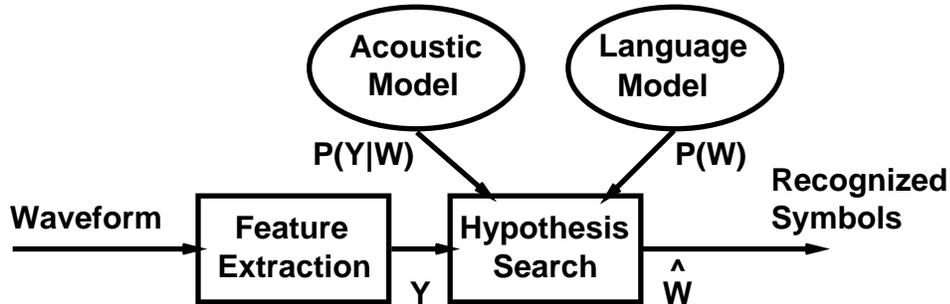


Figure 2.2: Statistically-based speech recognition system block diagram.

An extensively used approximation is to model the word sequence as an  $n$ -th order Markov process. The joint probability of an  $L$ -length word sequence is approximated as the product of  $L$  conditional probabilities where the conditioning terms contain not more than the  $k$  most recent words in the orthographic sequence. The size of  $k$  is commonly chosen between 1 and 3. Mathematically, the language model probability for an  $L$ -length word sequence using  $k = 2$  can be written as

$$P(\mathbf{W}) = P(w_1, w_2, \dots, w_L) = P(w_1)P(w_2|w_1) \prod_{i=3}^L P(w_i|w_{i-2}, w_{i-1}). \quad (2.5)$$

---

<sup>1</sup>Symbols are generally words, and will be referred to here as such, but they could also be short phrases or different pronunciations of the same orthographic word.

In section 2.2.1, the structure of the acoustic model is described, focusing mainly on the most commonly used form of an acoustic model, the Hidden Markov Model (HMM) [9, 53] which is the model used in this thesis. The basic building blocks of the acoustic model are described in section 2.2.2. Commonly used approaches to estimation of HMM acoustic model parameters are described in section 2.2.3.

### 2.2.1 Acoustic Model

Acoustic modeling involves computing the probability of a variable length observation sequence given a sequence of units, where the time alignment of observations to units is unknown. The most popular approach to acoustic modeling is by use of Hidden Markov Models (HMMs). This model represents an unobserved (i.e. hidden) state sequence corresponding to the word sequence as generating the observation sequence. The model makes two strong assumptions making it computationally attractive and it has empirically been shown to be an effective model. The first assumption is that the state sequence can be modeled as a Markov process (i.e. the probability of making a transition from state  $a$  to  $b$  only depends on being in state  $a$  and is independent of past state occupancies). The second assumption is that the observations generated by a state are conditionally independent and identically distributed (iid) given the state. Generally, a multivariate Gaussian or mixture of multivariate Gaussians is used to model the probability that an observation is generated by a particular state, often called the **emission** probability. Another commonly used approach to modeling the emission probability is by means of a discrete distribution. In this approach, the observations are quantized by a vector quantizer, and the emission probabilities of a state are defined as a discrete distribution over the different quantizer codewords.

In another approach to modeling the acoustic model probability, observation independence assumptions are made at the segment level rather than at the frame level. Such modeling approaches are referred to as **segment modeling** and allow more detailed acoustic modeling due to incorporating the dependence of neighboring

observations but are computationally less efficient than HMMs. Within this class of models, the dependence between observations are frequently modeled by a piecewise linear trajectory [47] but can also be modeled by means of a trajectory represented as a polynomial (**polynomial trajectory segment models**, PSMs [23, 34]). The work described in this thesis mainly focuses on the use of HMMs but the proposed techniques can be applied to PSMs as well as clustering algorithms are available for both modeling approaches [34].

In schematic form, the generation process of an HMM can be depicted as shown in figure 2.3 where circles denote the states of the model and arrows indicate possible transitions between states. In one possible (but not the only possible) generation scenario, the observations could originate from the three states as indicated in the figure by the dotted lines.

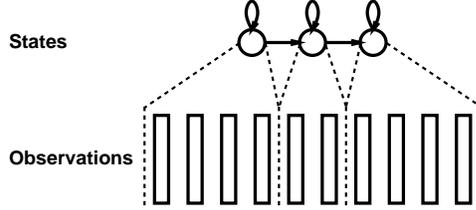


Figure 2.3: Schematic representation of a 3-state Hidden Markov Model and a possible sequence of generated observations.

Mathematically, the acoustic model probability of a  $T$ -length observation sequence  $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T\}$  can be written as,

$$P(\mathbf{Y} | \mathbf{W}) = \sum_{\mathbf{q} \in \{\mathcal{S}\}} P(\mathbf{Y} | \mathbf{q}, \mathbf{W}) P(\mathbf{q} | \mathbf{W}) \quad (2.6)$$

where  $\{\mathcal{S}\}$  denotes the set of all possible state sequences and a particular state sequence is denoted by  $\mathbf{q} = \{q_1, q_2, \dots, q_T\}$ . Using the model assumptions, this can be written as,

$$P(\mathbf{Y} | \mathbf{W}) = \sum_{\mathbf{q} \in \{\mathcal{S}\}} \left[ \pi(q_1) P(\mathbf{y}_1 | q_1) \prod_{t=2}^T P(q_{t-1} | q_t) P(\mathbf{y}_t | q_t) \right] \quad (2.7)$$

where  $\pi(q_1)$  denotes the probability of starting in state  $q_1$ . The Markov assumption of the state sequence has been used in that making a transition from state  $q_{t-1}$  to state  $q_t$  only depends on being in state  $q_{t-1}$ . The conditional independence of the observations given a state has been used in that the probability of observing  $\mathbf{y}_t$  only depends on the current state  $q_t$ . The HMM is therefore fully characterized by the number of states, the parameters describing the initial state probability function  $\pi(\cdot)$  defined over all states in the HMM, the state transition probability  $P(q_{t-1} | q_t)$  defined over all state pairs within the HMM, and the state emission probability functions  $P(\mathbf{y}_t | q_t)$  defined over all possible observations. The parameters of the HMMs of a system are generally estimated from data that is representative of the recognition task, as described later in section 2.2.3.

### 2.2.2 Sub-Word Units

The choice of an appropriate HMM topology is dependent on the recognition task. If the vocabulary size of the system is small, distinct HMMs can be constructed for each different word. Constructing HMMs for every distinct word is referred to as **whole-word** modeling. In many tasks however, the number of distinct words in the orthographics is very large and the training data is unbalanced, preventing reliable estimation of the parameters of some HMMs due to data sparsity. In such applications, distinct HMMs are built for units smaller than the word. These units are referred to as sub-word units. The most common choice of a sub-word unit is the phone. Using a set of sub-word units that is relatively small compared to the number of distinct words allows modeling of a large number of words with few HMMs, circumventing data sparsity problems. A model for a word can now be constructed by concatenation of sub-word unit models, i.e. different words share HMMs of units smaller than the word. This means that the acoustic model will consist of an additional component referred to as the **pronunciation dictionary** or **lexicon** which describes the unit sequences corresponding to the different words used

in the system. The representation of a word in terms of sub-word units is referred to as the **pronunciation**. Note that the pronunciation of a word can be a linear string of units but might also be represented by several strings or by a network of units. Most phonetically-based systems use a single linear pronunciation string for most of the entries in the lexicon and multiple linear pronunciations strings for very few entries (generally 1 to 2 percent).

Note that the formulation of the acoustic model is applicable to more general HMM topologies than shown in figure 2.3. Many recognition systems however use linear topology models allowing only a self-loop transition or a transition to neighboring states to the right. This is due to the fact that time can be seen as progressing from left to right and that the HMM is to provide a template of a sequence in time. The topology shown in figure 2.3 is referred to as a 3-state **left-to-right** topology which is a commonly used topology for phone HMMs. Other commonly used phone HMM topologies are 5-state left-to-right topologies. In addition, some systems include **skips** in the HMMs allowing slight deviations from the strict left-to-right state transitions.

The use of sub-word units is desirable over whole-word modeling, as it allows for construction of large vocabulary systems with few parameters and it provides a way to generalize to modeling words that are in the vocabulary but were never observed in the training data (provided that the pronunciations of such words are known). The advantage of whole-word modeling over modeling using sub-word units is that the model has to cover fewer contexts and thus less acoustic variability and can therefore be expected to describe the acoustics more accurately. The accuracy of the acoustic model using sub-word units can be improved by reducing the acoustic variability that is to be modeled by each sub-word unit. As the human articulatory system produces speech sounds as a continuous stream and because the elements of the articulatory system have a non-zero mass the changes from one speech sound to the next can only be realized in a continuous fashion. This continuity constraint means that the realization of a speech sound will be highly dependent on neighboring

speech sounds which is referred to as **co-articulation**. Modeling this unit **context** explicitly will therefore reduce the acoustic variability that is to be covered by such a context-dependent sub-word unit model. In other words, by allocating a distinct model for a speech sound dependent on the unit context, a unit is now represented by several models rather than a single model and each model is required to model only a part of the acoustic space corresponding to that unit.

Another approach to increasing the number of free parameters of the system is to allocate more free parameters to the emission probability representation allowing a more detailed description of the acoustic space in that way. Most systems will use Gaussian emission probability distributions out of computational considerations not because the emission probabilities are well modeled by this distribution. The modeling accuracy can therefore be improved by using a mixture of Gaussian distributions rather than a single Gaussian distribution. Increasing the number of mixtures will more accurately describe the part of the acoustic space that is to be represented by the unit but if units cover overlapping parts of the acoustic space due to contextual effects, increasing the accuracy of the distribution by increasing the number of mixture components may not improve recognition accuracy. In such cases, incorporating the dependency of neighboring units by explicit modeling of context will provide a more accurate description of the acoustics. Most systems will therefore generally use both explicit modeling of context as well as improve accuracy of the emission probability distribution by allocating more free parameters to representing that distribution.

Explicit modeling of context is desirable (and in many applications even required to obtain the desired level of performance), but it also introduces the problem of data sparsity similar to that encountered in whole-word modeling. For example, when using a phonetic unit inventory which for English consists of approximately 50 units, explicit modeling of left and right context will require  $125,000 \times 3$  distributions to cover all possible phones in context in a 3-state HMM. Many of these context-dependent phones will very infrequently or never be observed in the training data, preventing

reliable estimation of the parameters of those models. In order to still allow explicit modeling of context, systems allow sharing of model parameters between different context-dependent models. As the acoustic model likelihood is dominated by the emission probabilities and as most system parameters are allocated to the emission probability distributions, the parameter sharing techniques focus on deriving shared emission probabilities alone. Many different sharing scenarios have been investigated over the last few decades, differing in the part of the system that is shared (such as states or sequences of states) as well as parts of the models that are shared (such as sharing covariances but not means, or sharing mixture components but with differing mixture weights). As the work described in this thesis only uses state-clustered distributions, the focus will be on techniques to derive a system in which states share distributions, i.e. several states corresponding to different context-dependent unit models will share a single emission probability. This sharing mechanism is used in the most successful systems (e.g. [72]), so it is a reasonable choice.

One approach to derive such a system is by means of agglomerative clustering [32, 73]. This technique involves first computing a sufficient statistic of the data of every state of every unique context-dependent unit model. In the initial step, every observed unique state will constitute a state cluster. Then a measure is defined describing the similarity between state clusters and an estimation method is defined for finding cluster representatives given statistics from multiple states. The final set of clustered state distributions, shared by the unique states is then derived by successively merging the statistics from the most similar state clusters and re-estimating their representatives from the statistics contained in the clusters. Generally, emission probability densities are described by Gaussian distributions and maximum likelihood is used as an objective function, which determines the similarity measure and estimation method. In that setting, the statistics that are computed for each state are the mean, covariance and observation count.

An alternative to agglomerative clustering is to take a divisive approach which

starts from a single cluster containing all states and then successively splits that cluster. Divisive clustering can be conducted using unrestricted splits (e.g. [49]), analogous to the agglomerative approach. The advantage of such an approach over an agglomerative approach is that it reduces the computational cost of the clustering step. However, in the most frequently used divisive clustering approach of context-dependent phonetic units, the data divisions that are considered are derived from phonetic knowledge [6, 74]. The phonetic knowledge provides information as to which phonetic units are likely to have similar contextual effects. Using the phonetic groupings in a divisive clustering approach results in a decision tree that determines the emission probability distribution to use for a specific state in a given context-dependent phonetic unit. The phonetic knowledge is used to split the set of all possible contexts, into smaller subsets. The definition of phonetic groups provides a mechanism to generalize to unseen contexts. Even though not all possible contexts were observed, the decision tree will be able to predict a leaf distribution for every possible context, because the phonetic questions asked in the decision tree partition **all** possible contexts and take advantage of the class similarities of phonetic units to provide a distribution for the contexts not seen in the training data.

To build the decision tree, as in the agglomerative clustering case, initially a sufficient statistic is computed for each observed unique state in context. Then the pool of observed contexts is divided on the basis of phonetic questions about local context. For example, a possible question could be “is the left context a vowel?”. A separate model is estimated for each of the data partitions created by the question. An objective function such as likelihood is used to evaluate how much of a modeling accuracy improvement is achieved. In other words, the usefulness of asking a question about the phonetic identity of a context position is evaluated estimating the increase of the objective function by modeling the data using two models versus modeling the same data using a single model. The tree structure is derived by iteratively partitioning that leaf of the tree that yields the largest modeling accuracy increase

as measured by the objective function, i.e. the design is greedy. All possible contexts that map to a leaf of the decision tree share a probability distribution that is assigned to that leaf.

More specifically, consider the case of Gaussian emission probabilities and likelihood as the objective function. Assume data set  $A$  consists of two subsets  $A_l$  and  $A_r$  ( $A = A_l \cup A_r$  and  $A_l \cap A_r = \emptyset$ ) when split on the basis of a phonetic question. The modeling accuracy improvement of asking the phonetic question is then evaluated by computing a generalized likelihood ratio. The data likelihood of  $A_l$  and  $A_r$  with respect to the models derived by maximum likelihood estimation is compared to the likelihood of the data  $A$  with respect to its maximum likelihood estimated model. Generally separate decision trees are grown simultaneously for each state of each phone, splitting leaf nodes greedily across all trees. Note that in such an approach, not all trees will have the same number of leaves but likelihood gains dictate the tree topologies.

The advantage of clustering without restrictions on splitting (or merging) is that it will likely result in better quantization of the state distributions than the decision tree approach as it is not bound by the data divisions provided by the phonological groups alone. The disadvantage of the unrestricted clustering is that it does not allow for generalization to unobserved state distributions: splits (and merges) are defined in terms of observed data. Of course, the need for generalization is not always present. If no contextual effect of units in neighboring words is considered (referred to as a **word-internal** system) the inventory of context-dependent units can be enumerated by considering only contexts occurring within the pronunciation of each entry in the lexicon. If at least one example of each context-dependent unit in that inventory is observed in the training data, there is no need to generalize to context-dependent units not seen in training as such a unit will never be required. Generalization becomes a requirement, however, when there are unobserved vocabulary items and/or contextual effects of units in neighboring words are considered (i.e. in a **cross-word** system). In

addition to considering context effects within each entry in the lexicon (as in the word-internal case), all cross-word boundary effects can be enumerated from all possible combinations of words in the lexicon. It will generally not be the case, however, that an example of each of the context-dependent units in that inventory will have one or more examples in the training data.

Both the agglomerative as well as the divisive approach are sub-optimal due to their greedy nature. An agglomeration or split made at one stage of the algorithm will never be reconsidered at a later stage of the algorithm. A locally optimal clustering algorithm that does reconsider sharing scenarios is the K-means clustering algorithm. The K-means clustering algorithm iterates data partitioning and cluster representative re-estimation stages. At each partitioning stage, the data is divided over the clusters by computing the distance of each datum with respect to each cluster representative and assigning the datum to the minimum distance cluster. At the re-estimation stage, the cluster representatives are estimated from the data that was assigned to the respective cluster in the data partitioning stage. The K-means clustering algorithm is therefore defined by the distance measure used in the data partitioning stage and the objective function used in the cluster representative re-estimation stage. The K-means algorithm cannot guarantee to find the global optimum, but it does guarantee convergence to a local optimum. However, the K-means approach has some disadvantages: like the greedy agglomerative approach, it does not generalize to unseen context; it is generally computationally more expensive than either the greedy agglomerative or divisive approaches; and it requires knowledge of the number of clusters  $K$ .

Generally, the computational cost of all clustering techniques can be reduced by imposing additional constraints on sharing scenarios such as the restriction to allow sharing among different context-dependent models of the same center unit alone. This may, however, result in a sub-optimal system because of the reduced number of allowable sharing scenarios.

### 2.2.3 Acoustic Model Parameter Estimation

An obvious question that has gone unanswered up to this point is how the parameters of the HMMs are estimated. Although the use of many objective functions for parameter estimation have been described in the literature, the focus here will be on the use of maximum likelihood. If the observations were not modeled as generated by hidden states (i.e. if for every observation it was known deterministically which state generated it), the parameter estimation would simply involve computing the parameter estimate from the corresponding data. For Gaussian emission distributions and a ML estimation criterion, parameter estimation would in that case involve computing the mean and covariance of the data assigned to a state.

Given that the states are hidden, however, no deterministic relationship exists between states and observations, making parameter estimation less straightforward. One approach to the parameter estimation problem is referred to as **Viterbi training** which iteratively finds the most likely alignment between observations and states based on the last model and then re-estimates the model parameters based on that alignment. In another approach using the **Expectation-Maximization** (EM) algorithm [19, 10], a “probabilistic alignment” (i.e. the probability of every possible alignment) is estimated (the E-step) using the last model estimate. In contrast to the Viterbi training approach, the estimate is not used to arrive at a single alignment of states and observations. Instead it uses the probabilistic alignment to compute new model parameters that maximize the expected data likelihood (the M-step).

Mathematically, the Viterbi approximation makes the assumption that the data likelihood as defined in equation 2.7 as a sum over *all* possible observation and state alignments can be approximated by the *most likely path* alone,

$$P(\mathbf{Y} | \mathbf{W}) = \sum_{\mathbf{q} \in \{\mathbf{S}\}} P(\mathbf{Y}, \mathbf{q} | \mathbf{W}) \approx \max_{\mathbf{q}} P(\mathbf{Y}, \mathbf{q} | \mathbf{W}). \quad (2.8)$$

How the most likely alignment of states and observations can be derived is described in more detail in section 2.3. Given the Viterbi approximation, the state-observation

alignment is made explicit hence the data to estimate parameters from is made explicit. For example, if emission probabilities are modeled by a Gaussian distribution and ML estimation is used, the parameter estimate of the emission distribution of a state would involve computing the mean and covariance of the data aligned to that state. In the case that multiple states share an emission distribution, this distribution would be estimated from the data aligned to all different states sharing it. The initial state probabilities and state transition probabilities of the HMM can similarly be estimated by using the initial state and state transition counts observed in the explicit alignment. Note that the Viterbi training algorithm is not guaranteed to monotonically increase the ML objective function, since it is maximizing the likelihood of the observations and a single state sequence versus the likelihood of the observations (which is the sum over all state sequences).

The alternative to making the Viterbi approximation is to consider the state-observation alignment in a probabilistic way. Considering the alignment as a random vector and using the EM algorithm, the parameter estimation involves iteratively maximizing the expected likelihood of the data. As described in the work by Baum [10], maximization of the data likelihood is equivalent to maximization of an auxiliary function,

$$Q(\lambda', \lambda) = \sum_{\mathbf{q}} P(\mathbf{Y}, \mathbf{q} | \lambda') \log P(\mathbf{Y}, \mathbf{q} | \lambda), \quad (2.9)$$

where  $\lambda$  and  $\lambda'$  denote the current and new (in terms of the iterations of the algorithm) model parameters respectively. It can be shown that an increase of this  $Q$ -function corresponds to an increase of the data likelihood  $P(\mathbf{Y} | \lambda)$ . Considering an observation sequence of length  $T$ , again using the model assumptions as in equation 2.7, the log-likelihood of the data is,

$$\log P(\mathbf{Y}, \mathbf{q} | \lambda) = \log \pi(q_1) + \sum_{t=2}^T \log P(q_{t-1} | q_t) + \sum_{t=1}^T \log P(\mathbf{y}_t | q_t), \quad (2.10)$$

which can be used to decompose the auxiliary function into three parts,

$$Q(\lambda', \lambda) = Q_{\pi}(\lambda', \boldsymbol{\pi}) + \sum_{i=1}^N Q_{a_i}(\lambda', \mathbf{a}_i) + \sum_{i=1}^N Q_{b_i}(\lambda', \mathbf{b}_i) \quad (2.11)$$

where  $N$  denotes the number of states in the HMM,  $\boldsymbol{\pi}$  denotes the  $N$ -dimensional vector of initial state probabilities,  $\mathbf{a}_i$  denotes the  $N$ -dimensional vector of transition probabilities from state  $i$  to any of the  $N$  states in the HMM and  $\mathbf{b}_i$  denotes the parameter vector describing the emission probability distribution of the  $i$ -th state. The different parts of the decomposition of the auxiliary function can be written as,

$$Q_{\boldsymbol{\pi}}(\lambda', \boldsymbol{\pi}) = \sum_{i=1}^N P(\mathbf{Y}, q_1 = i \mid \lambda') \log \pi_i \quad (2.12)$$

$$Q_{\mathbf{a}_i}(\lambda', \mathbf{a}_i) = \sum_{j=1}^N \sum_{t=2}^T P(\mathbf{Y}, q_{t-1} = i, q_t = j \mid \lambda') \log a_{ij} \quad (2.13)$$

$$Q_{\mathbf{b}_i}(\lambda', \mathbf{b}_i) = \sum_{t=1}^T P(\mathbf{Y}, q_t = i \mid \lambda') \log b_i(\mathbf{y}_t) \quad (2.14)$$

where  $a_{ij}$  denotes the transition probability from state  $i$  to state  $j$  and  $b_i(\mathbf{y}_t)$  denotes the probability that observation  $\mathbf{y}_t$  was generated by state  $i$ . Finally, maximizing the  $Q$  function under the constraint that  $\sum_{j=1}^N \pi_j = 1$  and  $\sum_{j=1}^N a_{ij} = 1$  for all  $j$ , the re-estimated model parameters  $\hat{\boldsymbol{\pi}}$  and  $\hat{\mathbf{a}}_i$  are

$$\hat{\pi}_i = P(q_1 = i \mid \mathbf{Y}, \lambda') \quad (2.15)$$

$$\hat{a}_{ij} = \frac{\sum_{t=2}^T P(q_{t-1} = i, q_t = j \mid \mathbf{Y}, \lambda')}{\sum_{t=2}^T P(q_{t-1} = i \mid \mathbf{Y}, \lambda')}. \quad (2.16)$$

For the re-estimation of the parameters of the emission distribution of state  $i$ , assume the distribution is represented by a mixture of  $M$  Gaussians as

$$b_i(y) = \sum_{k=1}^M c_{ik} \mathcal{N}(y; \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \quad (2.17)$$

where  $\mathcal{N}$  denotes a Gaussian distribution,  $\boldsymbol{\mu}_{ik}$  denotes the mean of the  $k$ -th mixture component,  $\boldsymbol{\Sigma}_{ik}$  denotes the covariance of  $k$ -th mixture component and  $c_{ik}$  denotes the mixture weight of the  $k$ -th mixture component. The re-estimation formulas for these parameters are given by

$$\hat{c}_{ik} = \frac{\sum_{t=1}^T \gamma_t(i, k)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(i, k)} \quad (2.18)$$

$$\hat{\boldsymbol{\mu}}_{ik} = \frac{\sum_{t=1}^T \gamma_t(i, k) \mathbf{y}_t}{\sum_{t=1}^T \gamma_t(i, k)} \quad (2.19)$$

$$\hat{\Sigma}_{ik} = \frac{\sum_{t=1}^T \gamma_t(i, k) (\mathbf{y}_t - \hat{\boldsymbol{\mu}}_{ik})(\mathbf{y}_t - \hat{\boldsymbol{\mu}}_{ik})'}{\sum_{t=1}^T \gamma_t(i, k)}, \quad (2.20)$$

where  $'$  denotes transposition and  $\gamma_t(i, k)$  is defined as

$$\begin{aligned} \gamma_t(i, k) &= P(q_t = i \mid \mathbf{Y}, \lambda') P(m_t = k \mid q_t = i, \mathbf{Y}, \lambda') \\ &= \frac{P(q_t = i \mid \mathbf{Y}, \lambda') c_{ik} \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})}{\sum_{m=1}^M c_{im} \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im})}. \end{aligned} \quad (2.21)$$

The quantity of  $\gamma_t(i, k)$  can be interpreted as the estimated probability of occupying mixture component  $k$  of state  $i$  at time  $t$  given the data and model parameters. This estimated probability is referred to as the **occupancy probability**. Similarly, the estimated **transition probability** from state  $i$  to state  $j$ , that is used to re-estimate the HMM transition probabilities is frequently denoted as

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j \mid \mathbf{Y}, \lambda') \quad (2.22)$$

To estimate occupancy and transition probabilities required for re-estimation of the model parameters, it is required to evaluate the expression in equation 2.7. Note however that it would quickly become infeasible due to computational cost if the evaluation was implemented as defined. A computationally efficient method for evaluation of the data likelihood is the **forward-backward** algorithm. The efficiency of the algorithm is obtained by exploiting the two HMM assumptions. The likelihood of occupying state  $i$  at time  $t$  can be decomposed into two factors

$$\begin{aligned} P(\mathbf{Y}, q_t = i \mid \lambda) &= P(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t, q_t = i \mid \lambda) \\ &\quad P(\mathbf{y}_{t+1}, \mathbf{y}_{t+2}, \dots, \mathbf{y}_T \mid q_t = i, \lambda) \\ &= \alpha_t(i) \beta_t(i). \end{aligned} \quad (2.23)$$

As many of the state sequences will pass through state  $i$  at time  $t$ , the partial results can be used to avoid redundant computations. The first factor in equation 2.23 is obtained by the **forward** recursion,

### 1. Initialization

$$\alpha_t(i) = \pi_i b_i(\mathbf{y}_1), \quad i \leq i \leq N \quad (2.24)$$

## 2. Recursion

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(\mathbf{y}_{t+1}), \quad \begin{array}{l} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{array} \quad (2.25)$$

Note that the likelihood of the observations given the model can be computed as  $P(\mathbf{Y} \mid \lambda) = \sum_{i=1}^N \alpha_T(i)$ . The second factor in equation 2.23 is obtained by the backward recursion,

### 1. Initialization

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \quad (2.26)$$

### 2. Recursion

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(\mathbf{y}_{t+1}) \beta_{t+1}(j), \quad \begin{array}{l} t = T-1, T-2, \dots, 1 \\ 1 \leq i \leq N \end{array} \quad (2.27)$$

Given the  $\alpha$  and  $\beta$  values the occupancy probability of state  $i$  at time  $t$  can be computed as,

$$P(q_t = i \mid \mathbf{Y}, \lambda) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{k=1}^N \alpha_t(k) \beta_t(k)} \quad (2.28)$$

and the transition probability from state  $i$  to state  $j$  can be computed as,

$$P(q_{t-1} = i, q_t = j \mid \mathbf{Y}, \lambda) = \frac{\alpha_t(i) a_{ij} b_j(\mathbf{y}_{t+1}) \beta_{t+1}(i)}{\sum_{k=1}^N \alpha_t(k) \beta_t(k)}. \quad (2.29)$$

Given these estimates, the HMM parameters can be re-estimated as in equation 2.15, 2.16 and 2.18- 2.20.

## 2.3 Search

Given an acoustic model and a language model, the posterior probability of a hypothesized word sequence given an acoustic observation sequence can be computed and the task of the search is to find the most probable hypothesis. As described in equation 2.4, the search needs to take into account the contributions from both

the language and acoustic model components. The size of the search space generally prohibits an exhaustive search (a search that does not take advantage of the model structure nor makes any assumptions) and to arrive at a computationally feasible search algorithm, several assumptions and approximations need to be made. Note that the search complexity without making any assumptions is  $O(Q^T)$  as it needs to consider all length state-sequences  $Q$  corresponding to all possible word-sequences and all possible alignments of the  $T$  acoustic observations with those state sequences. The infeasibility of an exhaustive search becomes apparent from the realization that an observation sequence is generally several hundreds of observations long and that the number of possible alignments of this observation sequence with the word sequence grows exponentially with the observation length.

The first assumption that is made is that the acoustic model likelihood which is defined as a sum over all possible state-observation alignments can be approximated by the most likely state-observation alignment alone. This assumption, referred to as the *Viterbi* assumption, is mathematically described by equation 2.8. This Viterbi assumption results in a computational complexity reduction over the exhaustive search by taking advantage of the Markov assumptions made in both the acoustic and language models. Both these models have a limited *memory* which reduces both storage and computational requirements. Taking advantage of Bellman’s optimality principle [11], this means that only the most likely partial hypotheses found to be distinct by the model needs to be retained. Due to the limited memory of the model, different hypotheses can be found equivalent in the eyes of the model if they provide the same history. More specifically, the memory of the language model as defined in equation 2.5 for example is limited to the  $k$  most recent words, which means that hypotheses that do not differ in the final  $k$  words are equivalent in the eyes of the model. Then, using Bellman’s optimality principle, it is only required to retain the most likely hypotheses among those equivalent hypotheses. The principle proves that a future optimal hypothesis must be an extension of an optimal hypothesis in the past. This

means that if the objective is to find the optimal final hypothesis, suboptimal partial hypotheses can be discarded from further consideration. As the acoustic model makes an observation independence assumption given the state, it is sufficient to retain only the most likely hypothesis ending in an HMM state. In other words, among different hypotheses ending in the same state, only the most likely needs to be retained. If the language model has no or a single word memory it is sufficient to keep track of hypotheses ending at any of the  $P$  distinct HMM states of the models corresponding to any of the entries in the vocabulary of the system (if the vocabulary size is  $V$  and entries have an average number of  $S$  states in their models,  $P = SV$ ). The number of hypotheses, considered different by the limited memory models is referred to as the size of the **state space**. Note that explicit modeling of context when using sub-word models can cause an increase of  $P$  in a cross-word system as different word-initial and word-final states are needed to represent the different possible unit contexts across word-boundaries. In other words, the incorporation of explicit context information provides the model with more memory increasing the size of  $P$ . The possibly very large increase due to explicit cross-word context modeling is generally limited by the fact that many of the different contexts will share the same state distribution. Another way the memory of the model is increased is if the language model memory is increased. For every additional word explicitly considered in the language model history, the number of different hypotheses that can exist at any time increases with  $P$ . As in cross-word context modeling however, the possible exponential increase is generally not seen due to the ability to take advantage of parameter sharing in the language model (i.e. multiple word-histories will share the same probability distribution). Using the limited memory language and acoustic models as described, a state in the state space is identified by the word history relevant in the eyes of the language model and final HMM state where each state of each unique vocabulary entry (and each distinctly modeled context) is considered distinct. Note that even though many states of the HMMs corresponding to vocabulary entries may be shared in terms of

having the same distribution parameters (as in the case of sub-word modeling), they are to be considered distinct states in terms of the search state space because they occur in different vocabulary entries.

Using the most likely state sequences together with the HMM assumptions, and considering the contributions of the two modeling components, the acoustic model and language model, the **Viterbi search** can be seen as decomposed in two iterative update steps where the iterations are conducted in a time-synchronous fashion<sup>2</sup> for every time  $t$ .

- **within-word**: update every state in the state space that is **not** word-initial and retain the most likely hypothesis ending in that state. Let  $R(v, t)$  denote the most likely hypothesis ending in state  $v$  at time  $t$  which is updated as

$$R(v, t) = \max_{u \in \text{prev}(v)} R(u, t-1) a_{uv} b(\mathbf{y}_t | v) \quad (2.30)$$

where  $\text{prev}(v)$  denotes the set of states that can precede  $v$  in the state space,  $a_{uv}$  is the HMM state transition probability from state  $u$  to  $v$ , and  $b(\mathbf{y}_t | v)$  denotes the emission probability of observation  $\mathbf{y}_t$  from state  $v$ .

- **cross-word**: updating every word-initial state in the state space to retain the most likely hypothesis ending in that state. Again, let  $R(v, t)$  denote the most likely hypothesis ending in state  $v$  at time  $t$  which is updated as

$$R(v, t) = \max_{u \in \text{prev}(v)} R(u, t-1) P(W(v) | H(u)) b(\mathbf{y}_t | v) \quad (2.31)$$

where  $\text{prev}(v)$  denotes the set of states that can precede  $v$  in the state space,  $W(v)$  denotes the word of which  $v$  is the initial state, and  $H(u)$  is the most recent word history fitting in the language model memory of the hypothesis represented by  $u$ .

---

<sup>2</sup>Note that it is not required to approach the search for the most likely hypothesis in a time synchronous way and many systems in fact use an asynchronous approach. To limit the scope of the discussion here however, only the widely used time-synchronous search algorithm will be described.

Although taking advantage of the model assumptions greatly reduces the computational cost as compared to an exhaustive search, the cost will generally still remain too high for practical application. To reduce the search cost even further, an additional assumption is made that the final, most likely hypothesis is an extension of a partial hypothesis that was within a limited distance (rank-wise) to the most likely partial hypothesis. Therefore, after every update step, only the hypotheses associated with the most likely states in the state space are retained. The limited distance from the most likely hypothesis is generally defined by means of relative likelihood (the resulting pruning strategy is referred to as **beam-pruning** with the likelihood window referred to as the **beam**) but is frequently also defined by the maximum number of hypotheses that can be retained at any time (referred to as **histogram-pruning**).

The difference between the recognition search and the alignment process required in Viterbi training as described in section 2.2.3 is that in the search process, the word sequence is unknown. In the alignment procedure required in the Viterbi training process (frequently referred to as a **forced alignment**), the observation-state alignment is unknown but the word sequence is known (i.e. the search is *forced* to find an alignment for that word sequence alone). This also means that the language model will have no impact on this procedure since its value only impacts the alignment in the case where different word sequences are allowed.

## Chapter 3

# Automatically Derived Units

Large vocabulary speech recognition systems typically represent words in terms of sub-word units (SWUs), for which acoustic models can be reliably estimated. Part of the system design is therefore to decide on a suitable unit inventory and to define the lexicon. As discussed earlier, this problem is simplified in most systems by using phone-based units and a hand-crafted lexicon, frequently with a single linear phoneme string for the majority of the lexicon entries. Although the parameters of the unit models are generally estimated from data using an objective function such as maximum likelihood, no such function is used in the unit inventory and lexicon design. Given the lack of a clear objective in this part of the system design, the resulting unit inventory and lexicon are unlikely to be optimal in terms of the objective function used throughout the design of the rest of the system.

An alternative to manual derivation of a unit inventory and lexicon is to learn them from data. A unit derived in this way is generally referred to as an acoustic sub-word unit (ASWU). Over the last decade, a number of researchers have looked into this problem and found algorithms that automatically define model inventories and estimate unit model parameters using an objective function. The related problem of defining a lexicon in terms of these ASWUs has also received attention. An overview of the findings of these investigations is included in section 3.1. Common to all

lexicon design algorithms is that they evaluate candidate pronunciations of a word by means of an objective function. In this evaluation process, a candidate pronunciation is applied to all instances (or *tokens*) corresponding to a word and the candidate that receives the highest score (according to the criterion function) is chosen as the learned pronunciation. Common approaches to derivation of candidate pronunciations are deriving them from the pronunciations seen across training tokens during the unit design stage or by exhaustive enumeration of all possible pronunciations. One problem with this type of approach is that it decouples the unit inventory and lexicon design problems, which are clearly related. The unit inventory is no longer optimal after the pronunciations are restricted. Although this problem can be addressed by iterative re-estimation of the acoustic models and pronunciations [28], the approach remains problematic when the objective is to design large unit inventories. The computational issues related to evaluation of a large number of candidate pronunciations might be reduced by only evaluating a sub-set of all possible pronunciations, but the probability that the optimal pronunciation is one of the candidates reduces, especially when the size of the unit inventory increases. Also, a large pronunciation variability due to, for example, acoustic differences between speakers can reduce the probability of finding the optimal candidate.

In the work described in this thesis, a *joint* unit inventory and lexicon design algorithm is described that addresses the problem of a large initial pronunciation variability by introducing lexical constraints into the unit inventory design. By designing the units and lexicon jointly, the derived units and their associated acoustic models are matched to the lexicon. Section 3.2 describes the algorithm in detail. Section 3.3 describes experimental results. Finally, section 3.4 summarizes the main developments and findings.

## 3.1 Acoustic Sub-Word Units

In contrast to the approach described here, the problems of automatic unit design and lexicon design were previously considered as two related but separate problems. The description of previous work is therefore divided in two sections. In section 3.1.1 previously designed algorithms to automatically learn unit inventories and their models from data are described. Section 3.1.2 describes previous work related to lexicon design algorithms.

### 3.1.1 Unsupervised Learning of Automatic Units

Work in unsupervised learning of sub-word units generally involves two steps:

- segmentation without the use of lexical information to find stationary regions, and
- clustering of these segments to get the model inventory.

Work by Svendsen and Soong [62] describes an algorithm for finding stationary regions by using the dynamic programming (DP) algorithm. The criterion function used in the DP search is an approximation of the likelihood assuming a multivariate Gaussian model for each segment. Model mean parameters are estimated from segments hypothesized in the DP search but as data sparsity problems prohibits the estimation of a covariance from a single segment, a single diagonal covariance, estimated from the entire utterance, is used throughout the search. A variation of this algorithm was used by Lee *et al.* [36] who assumed an autoregressive stochastic process model and used minimum Itakura-Saito distortion as the objective function. Yet another approach to automatic segmentation was described by IBM: vector quantization was used to obtain a segmentation [5].

All the reported work also differs in the choice of algorithm for the subsequent segment clustering step. Lee *et al.* use the Itakura-Saito distortion measure in clustering

and then estimate the parameters of an HMM from the segments in a cluster [37]. Paliwal and others [50, 28, 63] describe a similar approach but apply a clustering technique using a Euclidean distance to the centroids of the segments derived by the automatic segmentation algorithm described by Svendsen and Soong [62].

In related work in the speech coding field, work by Shiraki and Honda [59] describes an approach to the unit learning problem that iteratively adjusts the parameters of the units in the inventory and the boundaries of the segmentation. As the segmentation is derived without a unit inventory, segment boundaries are sub-optimal for the unit inventory subsequently derived by clustering of the segments. Their work shows that iterative adjustment of segment boundaries and model parameters improves the models and converges to a segmentation that is locally optimal for the corresponding unit inventory.

### **3.1.2 Pronunciation Modeling**

Before providing an overview of pronunciation modeling work that can be found in the literature, it is important to understand how acoustic variability is modeled within a phone-based system.

Although some of the acoustic variability that is to be captured in the phone models is reduced by explicit modeling of context (see section 2.2.2), the densities used in the context-dependent phone models are complex since all the acoustic variability not explicitly modeled by context (such as speaker dependency and pronunciation variations) is to be captured within the phone-based models. This approach has proven successful for read speech but is problematic for spontaneous speech due to an increase of acoustic variability. For example, the reduction or deletion of phones will cause erroneous phone boundaries to be estimated in the training process, resulting in the introduction of noise in the densities of the phone models. As phone reduction does not occur very frequently in read speech, the described training procedure was suitable but as phone reduction is much more frequent in spontaneous speech, the

described training procedure becomes problematic.

As the variability in the pronunciation of read speech is much smaller than in spontaneous speech, the topic of pronunciation modeling has received more attention during the last few years now that the research focus has shifted towards spontaneous speech. The first body of work that is of interest while studying this topic originates from the late 80's though, due to the interest in using automatically derived units which required a solution to the lexical mapping problem.

To formulate the pronunciation problem in more detail, consider a sub-word unit inventory denoted as  $\{\phi_1, \dots, \phi_N\}$  and a set of words known to the system denoted as  $\{W_1, W_2, \dots, W_L\}$ . The task of the pronunciation modeling is to derive a model for the pronunciation of the entries in the lexicon in terms of the sub-word units. The training material to estimate the parameters of such a model is denoted as a set of observations  $\{\{\mathbf{Y}^{W_1}\}, \{\mathbf{Y}^{W_2}\}, \dots, \{\mathbf{Y}^{W_L}\}\}$  where  $\{\mathbf{Y}^{W_x}\}$  denotes the collection of observations (or tokens) of word  $W_x$  in the training data (with possibly  $\{\mathbf{Y}^{W_x}\} = \emptyset$ ). The algorithms described in the literature for solving this problem can be classified into three groups:

1. **Word-specific ML models:** using a likelihood approach with word-specific training data.
2. **Generic models:** using a likelihood approach with a model that *does not* require word-specific training data.
3. **Word-specific inference techniques:** using structural inference techniques that account for word confusability again using word-specific data.

For the first class of approaches, the  $M$  most likely pronunciations for word  $W_x$  are derived as

$$\begin{aligned}
 \{\Phi_1^{W_x}, \dots, \Phi_M^{W_x}\} &= \underset{\Phi}{\text{M\_Argmax}} Pr(\Phi \mid \mathbf{Y}^{W_x}, W_x) \\
 &= \underset{\Phi}{\text{M\_Argmax}} [Pr(\mathbf{Y}^{W_x} \mid \Phi)Pr(\Phi \mid W_x)] \quad (3.1)
 \end{aligned}$$

where  $\Phi_q^{W_x}$  denotes the  $q$ -th pronunciation for word  $W_x$  and  $M\_Argmax$  denotes the vector equivalent of the argmax operator returning the  $M$  most likely sequences instead of the most likely sequence alone. There are two likelihood components in the expression. The component  $Pr(\Phi | W_x)$  is a model of how likely the sub-word unit sequence  $\Phi$  is for a given word  $W_x$ . The  $Pr(\mathbf{Y}^{W_x} | \Phi)$  component models the likelihood that pronunciation  $\Phi$  generates these observations. An interpretation of this type of modeling is that the tokens of word  $W_x$  are used to evaluate the pronunciations provided by the *generic* pronunciation model  $Pr(\Phi | W_x)$ . This model is therefore expected to be an exact model but requires availability of training data for every word.

The second type of algorithm derives the  $M$  most likely sequences for word  $W_x$  as

$$\{\Phi_1^{W_x}, \dots, \Phi_M^{W_x}\} = M\_Argmax_{\Phi} Pr(\Phi | W_x). \quad (3.2)$$

This type of algorithm is *generic* as it does not require word-specific training data, but it is likely to be less exact than the previous model. A model of this type is appropriate if large coverage is desired while the previous type of algorithm is more appropriate if sufficient training data is available for all words in the system's vocabulary.

The third type of algorithm disregards likelihood as the objective function and uses other criteria such as coverage and number of free parameters in the pronunciation model.

Examples of the first type of algorithm can be found in the work by IBM [7] which uses a tree to estimate the probability of a phone from the word transcription ( $Pr(\Phi | W_x)$ ) and uses acoustic models in the form of HMM's to estimate the  $Pr(\mathbf{Y}_n^{W_x} | \Phi)$  component. Another example of this type of model is described by Paliwal [50]. The model for  $Pr(\Phi | W_x)$  is 1 for the sequences  $\Phi$  that are actually observed and 0 for all others. The  $Pr(\mathbf{Y}_n^{W_x} | \Phi)$  component is modeled as in the work by IBM using HMM acoustic models.

An example of the generic pronunciation model is described by Riley [55, 56]. In this work, phone pronunciation networks, augmented with pronunciation probabilities

are estimated from a dictionary pronunciation by use of a decision tree estimator model. A similar approach is described to deal with pronunciation variability due to accent differences by Humphries *et al.* [31]. The trees in this approach predict from a phonemic dictionary pronunciation and neighboring predicted phones. An alternate approach is proposed by Weintraub *et al.* [68] where a maximum entropy trained phone n-gram model is used to model the dependency between the neighboring phones, predicted by the decision tree predictor operating on the phonemic dictionary pronunciation. In the work of Lucassen and Mercer [41], a tree is used to predict an HMM state rather than a (set of) deterministic phone sequences. This state is then used to model the probabilities of observing phone sequences.

Other generic modeling approaches, not incorporating a tree predictor are also found in the literature. Paliwal [50] describes a bigram model for  $P(\Phi | W)$ . In the work of Cohen [16], probabilities of handwritten phonological rules are learned from training data. In the work of Wooters *et al.* [71], the probabilities of a predetermined set of pronunciations is derived from the training data by means of a forced alignment. The probability augmented pronunciations of a word are then merged so as to maximize the posterior probability of the word model.

The third class of models use criterion functions other than ML in the derivation of pronunciations. In the work of Sloboda and Waibel [60], candidate pronunciations are derived from the output of a phoneme recognizer. These candidate pronunciations are then accepted as new valid pronunciations on the basis of confidence (an estimate of the reliability of the recognized word) and a measure of the confusability of the new entry to existing entries. Westendorf *et al.* [69] describe a technique that produces a product graph of the lattice output of a phoneme recognizer and the pronunciation network from the pronunciation dictionary. A path through this product graph that minimizes a cost function is then sought using the dynamic programming algorithm. Costs are hand-derived scores of possible confusions. The pronunciation described by the derived path is then added to the dictionary pronunciation network, only if the

number of arcs and nodes that are to be added are below some threshold.

## 3.2 Joint Unit and Lexicon Design

The two basic steps of any unit inventory design algorithm, as described in section 3.1, are an acoustic segmentation step followed by a clustering step (e.g. [37, 63, 4, 28]). In most systems, lexicon design would be a subsequent step, with the goal being to find a single linear pronunciation for each word. Similarly, our focus is on deriving a single<sup>1</sup> linear pronunciation for each word, but *within* the inventory design process. More specifically, the key elements that differ in our approach compared to previous work are the use of pronunciation-related constraints in unit design, the consistent use of a maximum likelihood objective function, and progressive unit inventory and model refinement.

Using the linear pronunciation assumption, two important new constraints are introduced that allow joint inventory and lexicon design. First, the segmentation step is constrained to use the same number of segments for every token of a word. This constraint will be referred to as a *pronunciation length* constraint. Then, the clustering step is constrained by pre-grouping all the segments in the different training tokens of a particular word according to their position in the fixed-length sequence. This constraint will be referred to as a *pronunciation consistency* constraint. The lexicon is implicitly defined after completion of the clustering step, since the data from different training tokens representing the same position within a word are assigned to a single cluster. In addition, since the maximum likelihood objective function is used, the acoustic model parameters are also defined as a result of clustering. Section 3.2.1 describes segmentation and clustering with constraints in more detail.

---

<sup>1</sup>Since single-pronunciation dictionaries are relatively successful for many speech recognition tasks, particularly for first-pass decoding stages of a multi-pass search, we defer the problem of representing pronunciation variation within words for the future (see chapter 6).

Progressive unit refinement is important for at least two reasons. First, once data is clustered, the segmentation that the initial units were based on may no longer be appropriate. Section 3.2.2 describes how the unit model inventory and corresponding segmentation can be refined further using retraining to obtain a better match between the segmentation and the models. Second, even phone-based systems benefit from progressive techniques for increasing the acoustic model complexity, both in terms of contextual and temporal resolution [72, 65, 49]. Approaches to progressively increasing the resolution of the system both in contextual and temporal structure are described in section 3.2.3. The scope of the contextual refinement algorithms described in that section are limited to those *not* using explicit modeling of context. How context can be incorporated into the model explicitly is described in detail in chapter 4.

### 3.2.1 Initial Unit Inventory and Lexicon Design

As mentioned above, the initial inventory and lexicon design is a two-step process, involving segmentation and then clustering. The first step provides a segmentation in which all training tokens of a word contain the same number of segments. In the second step, the segments are clustered subject to a pronunciation consistency constraint to define the unit inventory and lexicon.

The first step in designing an ASWU system is **acoustic segmentation**, that is, finding segmentation times that divide each word token into piecewise stationary regions that can be reasonably well modeled with a single HMM state. In our case, the number of regions per token is fixed for a particular word – a pronunciation length constraint. The pronunciation length could be specified in terms of some baseline phone pronunciation length, but some of the potential power of automatic learning is lost in this case. Instead, we first run an unconstrained acoustic segmentation, then choose the median number of segments associated with a word for the length constraint, and finally rerun acoustic segmentation subject to the length constraint,

as described below.

Unconstrained acoustic segmentation functions as an initialization step. Taking an approach similar to that in [50], the maximum likelihood segmentation of the training data is found by use of dynamic programming. Let  $\mathbf{y}_t$  be a  $d$ -dimensional observation vector, such as a vector of cepstral coefficients representing a window of speech at time  $t$ . The unconstrained acoustic segmentation algorithm involves recursive updating for every time  $t$  and every allowable number of segments  $n$ :

$$\delta(t, n) = \max_{t-l_{max} \leq \tau \leq t-l_{min}} [\delta(\tau - 1, n - 1) + \log P(\mathbf{y}_\tau, \dots, \mathbf{y}_t \mid \mu_{\tau,t}, \Sigma)], \quad (3.3)$$

where  $l_{min}$  and  $l_{max}$  denote minimum and maximum allowable segment lengths. In addition to updating  $\delta(t, n)$ , the index  $\tau$  that maximizes equation 3.3 is stored, allowing the most likely segmentation to be found in the end by tracing back. The (generalized) likelihood of the segments during the dynamic programming is computed using a multivariate Gaussian model with a single diagonal covariance  $\Sigma$  used for all the segments. This covariance matrix can be estimated either on a per utterance basis, a per speaker basis, or from the entire training corpus. The mean parameter of the Gaussian model is computed from the hypothesized segments; the use of HMMs corresponds to the assumption that speech is piecewise stationary.

During segmentation, the likelihood increases monotonically with the number of allowable segments, since there are more free parameters with which to fit the data. A thresholding mechanism is used to control the average number of segments, defining the temporal resolution (average state duration) of the resulting system. To control the average number of segments, we investigated using an average-likelihood-per-frame threshold  $L_{avg}$ , as well as a weighted minimum description length (MDL) criterion. In the weighted MDL case, the number of segments  $\hat{S}$  is determined by

$$\hat{S} = \underset{S}{\operatorname{argmin}} -\delta(T, S) + \frac{\alpha}{2} m(S) \log(T), \quad (3.4)$$

where  $T$  denotes the number of feature vectors in the utterance and  $m(S)$  denotes the number of free parameters used in the segmentation  $S$ . The penalty term is weighted

by  $\alpha$  to allow some external control over the temporal resolution of the system, so as to have comparable complexity with the two methods. Both the threshold  $L_{avg}$  and MDL weight  $\alpha$  are chosen empirically to obtain an average segment length close to what is associated with a 3-state per phone system for the target recognition task.

Next, the acoustic segmentation is aligned with automatically-derived word segmentation times by assigning each acoustic segment to the word token with which it overlaps most. (The word segmentation times are given by forced alignment to the word transcriptions using a context-dependent HMM system.) For each word, the median of the number of acoustic segments over all the training tokens is used to define the pronunciation length. The training data is then segmented again using dynamic programming (equation 3.3) under the constraints that:

- the number of acoustic segments for a word is equal to the median pronunciation length, and
- each word boundary coincides with an acoustic segment boundary.<sup>2</sup>

In the resulting segmentation, all training tokens of a word have the same number of segments.

The second step involves **clustering** the results of the segmentation step to define the unit inventory. The clustering algorithm used here differs from that used in [63, 50, 28] in that maximum likelihood is used as an objective rather than minimum Euclidean distance. Specifically, the repartitioning step involves computing the likelihood of segments given the model parameters of a cluster, i.e. the “distance” is a negative log likelihood. The cluster re-estimation procedure consists of finding the ML parameter estimates of a Gaussian distribution from the data contained in the cluster. Cluster centroids therefore directly represent unit models and clustering

---

<sup>2</sup>In a small number of cases, the median segment length is longer than the number of speech frames in a token, making segmentation impossible, in which case the word boundary times are relaxed.

addresses both the inventory and model design problems, whereas in [63, 50, 28] unit model parameters had to be estimated in a separate step from the data partition defined by clustering.

Before clustering, the data is grouped to ensure pronunciation consistency, our second pronunciation constraint. Using the fact that clustering is based on a maximum likelihood objective, the grouping is implemented by computing a sufficient statistic for each collection of segments originating from different training tokens in the same position within a word. The sufficient statistics for the HMM are the sample mean  $\mu_p$  and covariance  $\Sigma_p$  and the total number of vector observations contained within the group  $N_p$ . These statistics are stored for each unique position within each unique word: if there are  $V$  entries in the vocabulary and the average median pronunciation length is  $R$ , the data is grouped into  $VR$  groups. These statistics will be referred to as **atomic group sufficient statistics**. As the sufficient statistic representations of these atomic groups cannot be split in clustering, this grouping ensures pronunciation consistency.

Let a group of  $K$  segment observations  $\mathbf{y}_p = \{\mathbf{Y}^1, \dots, \mathbf{Y}^K\}$ , of lengths  $\{L_1, \dots, L_K\}$ , have sufficient statistics  $(\mu_p, \Sigma_p, N_p)$  where  $N_p = \sum_{i=1}^K L_i$ . The distance of the group with respect to the cluster with parameters  $\mu_c$  and  $\Sigma_c$  is computed as the negative log likelihood

$$-\mathcal{L}(\mathbf{y}_p \mid \mu_c, \Sigma_c) = \frac{N_p}{2} \left[ D \log(2\pi) + \log(|\Sigma_c|) + \text{tr}(\Sigma_c^{-1} \Sigma_p) + (\mu_p - \mu_c)' \Sigma_c^{-1} (\mu_p - \mu_c) \right]. \quad (3.5)$$

Once observations are assigned to a cluster, the ML parameter estimate given  $G$  groups of observations are

$$\mu_c = \frac{1}{\sum_{q=1}^G N_q} \sum_{p=1}^G N_p \mu_p \quad (3.6)$$

$$\Sigma_c = \frac{1}{\sum_{q=1}^G N_q} \left[ \sum_{p=1}^G N_p (\Sigma_p + \mu_p \mu_p' - \mu_c \mu_c') \right]. \quad (3.7)$$

Starting from a single cluster, the cluster inventory is increased by binary divisive clustering. Iteratively, the cluster with the lowest average likelihood per frame is selected to be split. Two new clusters are defined by first obtaining an initial split by perturbing the cluster mean, and then running a number of binary K-means clustering iterations using only the data that was contained in the original cluster before splitting. Clusters with fewer observations than a minimum occupancy threshold (100) are not considered for splitting. After the cluster inventory is increased up to a heuristically determined inventory size, a number of K-means iterations (10) using all the data and the full cluster inventory are run. If any cluster during this stage has fewer observations than the minimum occupancy threshold, the cluster is removed from the inventory and the data previously held within that cluster is repartitioned over the remaining clusters. The clustering algorithm derives the final unit model inventory by alternating between divisive and K-means iterations, increasing the number of clusters in stages. Final inventory sizes are chosen to be similar to that used in the comparable phone-unit-based systems. Note that inventory sizes are difficult to tightly control, since they are a reduced version of the specified target due to the elimination of clusters falling below the minimum occupancy threshold.

The final data partition over the cluster inventory defines the lexicon by virtue of the data grouping. When neighboring segments within a word are assigned to the same cluster, the segments are collapsed into a single entry in the lexicon. As the units assume a stationary distribution, repetitions of the same unit label in the lexicon are equivalent to a single instance of the label. Note that segments that were found to be distinct in unrestricted acoustic segmentation can be labeled as equivalent after the quantization introduced by clustering. During this unit collapsing step, some temporal resolution is lost relative to the temporal resolution of the acoustic segmentation. Options to control the temporal resolution of the system are described in section 3.2.3.

### 3.2.2 Re-training

The acoustic segmentation was optimal given an unconstrained model inventory (size  $S$  for  $S$  segments), since model parameters in acoustic segmentation are derived separately for each instance of a segment in the dynamic programming search. After clustering, the acoustic space is quantized into  $C$  models with  $C \ll S$ , making the acoustic segmentation suboptimal. To achieve a matched condition between the unit model inventory and the segmentation, a re-training algorithm can be used, either Viterbi or EM. The Viterbi training algorithm, used here, iteratively re-segments the data to find the optimal segmentation given the current unit model inventory and then re-estimates unit model parameters using the new segmentation. Given a lexicon consisting of linear pronunciation strings derived by the algorithm described in section 3.2.1, a word-level transcription can be expanded into an  $S$ -length unit-level transcription  $\{u_1, \dots, u_S\}$ . The (Viterbi) segmentation step involves recursive updating for every time  $t$  and every unit index  $i \in \{1, \dots, S\}$  of

$$\Psi(t, i) = \max_{j \in \{i-1, i\}} \Psi(t-1, j) + \log P(\mathbf{y}_t \mid \mu_{c(i)}, \Sigma_{c(i)}) \quad (3.8)$$

where  $\mu_{c(i)}$  and  $\Sigma_{c(i)}$  denote the mean and covariance of the  $i$ -th segment which has unit label  $c(i)$ . In addition to updating  $\Psi(t, i)$ , the boundary times between units in the sequence are stored so that the segmentation that maximizes equation 3.8 is known at completion of the recursion. Given this new segmentation, the unit model parameters are re-estimated from this new segmentation using standard ML estimation.

### 3.2.3 Progressive Refinement

For most recognition tasks, HMM systems tend to give improved performance as complexity is increased, in the sense of having a larger unit inventory. The increase of modeling accuracy is provided by the larger number of free parameters. Generally, the approach to increasing the number of free parameters of a phonetic unit-based sys-

tem is by explicitly representing context. How this type of approach can be exploited within the ASWU framework is described in detail in chapter 4. In addition to explicit incorporation of context, the joint design of units *and lexicon* within the ASWU framework provides another approach to increasing model complexity by means of implicit modeling of context (i.e. the implicit context is encoded in the position dependency within the words). One way to obtain such a high complexity ASWU system is by design of a large unit inventory based on a single divisive clustering run, starting from the segment boundaries derived by acoustic segmentation. Alternatively, one could run Viterbi training after an intermediate size unit inventory is designed, re-estimate the atomic group sufficient statistics, and continue divisive clustering to increase the inventory using these new statistics. In preliminary experiments, better results were obtained using this second approach.

The iterative ML clustering and Viterbi training approach is beneficial as it allows retaining near-optimal segment boundaries in the unit inventory design algorithm. A problem it introduces however is that throughout the unit design, the temporal resolution of the system decreases. When neighboring units within a word are clustered in the same cluster, they are merged to form a single unit causing loss of temporal resolution. Note that segments merged given a small unit inventory might not have been merged given a large unit inventory, so it may be useful to allow segment splits for the high complexity system.

One approach to avoid the problem of loss of temporal resolution during the unit design process is to set the threshold that controls the temporal resolution in the first unconstrained acoustic segmentation so that initially segments are very short (high temporal resolution), compensating for the loss in temporal resolution incurred during the unit inventory design algorithm. This approach will be referred to as refinement with **no temporal adjustment**. Unfortunately, the resulting system has lower accuracy, probably because of poor decisions on parameter sharing during the clustering stage.

Another approach to circumvent the problem of the loss of temporal resolution is to increase the temporal resolution by splitting the segments derived after a Viterbi training stage. Two options for increasing the temporal resolution by segment splitting were investigated. One approach, splits each individual segment in each word token in two using the constrained acoustic segmentation algorithm described in section 3.2.1. New word-position-dependent atomic group sufficient statistics are then computed for the new segmentation, and a new lexicon can be defined by running one or more K-means clustering iterations, starting from the existing unit model inventory. Successive identical units will be merged as before, so the effective increase in pronunciation length is much less than a factor of two. Progressively refining the system using this approach will be referred to as **binary temporal adjustment**.

An alternative segment splitting approach is to let the unique unit labels derived by the last Viterbi re-segmentation function as the word labels did in the initial design stage described in section 3.2.1. Where the median number of acoustic segments across tokens of a word were considered in the initial design stage, here the median across tokens of a unit label are considered. Then using the median pronunciation lengths for each unit, a temporally finer segmentation can be derived by a constrained acoustic segmentation, constraining tokens of a unit label to be split into this median number of segments. Given this finer segmentation, new word-position-dependent atomic group sufficient statistics can be computed and clustered to derive a larger unit inventory. This approach will be referred to as **variable temporal adjustment**. The binary approach does not consider possible segment splits into more than two segments which is possible in the second approach. The variable approach, however, does not necessarily consider a multi-unit representation of every segment and might therefore miss possible modeling accuracy improvements gained by allowing this.

### 3.3 Experiments

Experiments were conducted on the DARPA Resource Management corpus [52], which is a read speech corpus with a 991 word vocabulary. Although the proposed algorithm is developed with the ultimate goal of attacking the increased acoustic variability problem in spontaneous speech corpora, initial experiments were conducted on the smaller Resource Management corpus to investigate the viability of the algorithm and explore different options at a lower experimental cost. In addition, it allows direct comparison with the current best ASWU system.

The 109-speaker training set of approximately 228 minutes of speech was used as training material for unit inventory and lexicon design. The training set contains 46814 training tokens for 991 unique words. The most frequently observed word has 4112 training tokens, the least frequent has 1 training token. The average number of training tokens per word is 47, the median 11. The February 1989 test set containing 300 utterances was used as a development test set. The four available test sets (Feb 89, Oct 89, Feb 91 and Sept 92) were used in the main system comparisons to have more reliable results, and the average of the four results will be referred to as the **full test set** performance. The word-pair grammar provided with the Resource Management corpus was used in the search.

Feature vectors were computed every 10 ms and included 12 Mel-warped cepstral coefficients and normalized energy and their first and second order differences (39 dimensions in total). No techniques to reduce the variability due to speaker identity were applied to the feature computation step. The speech signal was windowed using a Hamming window of 25 ms, and a first-order pre-emphasis filter (0.97) was applied. Diagonal covariances were used throughout the experiments.

The word recognition performance is derived from a dynamic programming word-level string alignment of the recognizer output to the reference transcription. Denoting the number of correct labels as  $H$ , the number of insertions as  $I$  and the total number of labels in the reference transcription as  $N$ , the percent accuracy figure is

defined as  $(H - I)/N \times 100\%$ .

### 3.3.1 Phone systems

For performance comparisons, phone-based HMM systems were trained using the HTK toolkit [30]. The 48-phone lexicon provided with the Resource Management corpus was used. Except for the silence model which was a single state model, the 48 phone HMMs had a 3 state left-to-right topology without allowing skips. Starting from context-independent (CI) model parameter (i.e. one 3-state model per phone) estimates derived from the TIMIT corpus, 4 EM training iterations were performed to derive a 145 state CI system. These models were then cloned for each unique triphone context (i.e. one model for every unique left and right phone context), and 2 EM training iterations were run to train the 2254 model/6760 state system. Two parameter sharing techniques were used on the 2254 triphone models. One involved agglomerative clustering, resulting in 1514 distributions, and the other used tree-based clustering, which gave a 1369, 1100 and 748 distribution system dependent on the parameter settings of the clustering algorithm. All clustered systems outperformed the unclustered triphone system, with 88-93% accuracy for the clustered systems compared to 87.5% accuracy for the unclustered triphones on the February 1989 test set, when not modeling cross-word context effects. When using a 1583 distribution cross-word system, performance improved with approximately 3% accuracy over the word-internal triphone system but in this chapter the baseline for comparison are the systems that do not model cross-word context.

The recognition performance on the February 1989 and full test sets is given in table 3.1. As expected, the clustered triphone systems significantly outperform the context-independent system. The agglomerative clustering approach gave slightly better results than tree-based clustering, but this may simply reflect the larger number of distributions in the agglomerative system. This difference in size is a consequence of differences in the agglomerative vs. divisive clustering strategies, as well as HTK

control parameters for determining tree sizes. Note that these results are lower than that reported for systems using mixture distributions, but the extension to mixtures is straightforward and will benefit both phonetic SWU and ASWU systems.

System	Number of states	Accuracy	
		Feb 89 Test	Full Test
Context Independent	145	75.6	75.6
Agglomerative clustered word-internal triphones	1514	90.2	89.1
Tree-based clustered word-internal triphones	748	88.5	87.9
Tree-based clustered word-internal triphones	1100	89.8	88.1
Tree-based clustered word-internal triphones	1369	90.0	88.6
Tree-based clustered cross-word triphones	1583	92.7	91.1

Table 3.1: HMM system results (% accuracy) on the February 1989 and full test sets for phone-based different system configurations.

### 3.3.2 ASWU systems

To derive an ASWU unit model inventory and lexicon, the algorithm described in section 3.2 was applied. Three sets of experiments were conducted. The first set focussed on the design of a *low complexity system*, for comparison to the HMM context-independent phone models and is described in section 3.3.2. The second set of experiments investigate the effect of different thresholding and variance estima-

tion techniques in the acoustic segmentation steps and is described in section 3.3.2. The third set of experiments described in section 3.3.2 provides results obtained when designing a high complexity system in which context is modeled implicitly by means of word-position dependence. Finally, section 3.3.2 shows some examples of learned pronunciations and compares those learned pronunciations among words with equal morpheme bases. During each ML clustering stage, the minimum cluster occupancy was set to 100 frames; clusters with fewer observations were removed. The process of removing infrequent clusters often resulted in a significant decrease in inventory size, so that it was sometimes difficult to obtain inventories of the same size when comparing different ASWU design methods.

### **Low Complexity System Design**

Two low complexity systems were designed by first deriving of a “coarse” inventory. For both systems, a unit inventory comparable in complexity to the CI phone model system was derived by one refinement step. The two cases differed in whether they used no temporal adjustment or variable temporal adjustment to obtain the model inventory. For the system using no temporal adjustment, the threshold at the acoustic segmentation stage was set so that the resulting segmentation contained approximately 3 acoustic segments per phone, which gives a temporal resolution comparable to the phone-based systems. The unit inventory sizes were 126 units for the coarse intermediate stage and 141 for the resulting system. The second system using a variable temporal adjustment approach used an acoustic segmentation threshold for the coarse unit design stage that resulted in approximately 1 acoustic segment per phone. The acoustic segmentation threshold used in the refinement step was set so that the resulting segmentation contained approximately 3 acoustic segments per phone. The unit inventory sizes were 50 units for the coarse intermediate stage and 150 for the resulting system. The recognition results using these systems and the comparable 145 state CI-phone-based system on either the February 1989 test set or the full test

Temporal adjustment	February 1989 (% accuracy)	Full test set (% accuracy)
CI-phones	75.6	75.6
none	80.4	78.2
variable	82.7	80.3

Table 3.2: Recognition results of the CI-phone baseline and low complexity ASWU systems using different approaches to temporal adjustment.

set are given in table 3.2.

Both systems used average-likelihood-per-frame thresholding and a grand covariance (i.e. covariance derived from the whole corpus) in the acoustic segmentation stages. Preliminary experiments, using different acoustic segmentation thresholds to vary the temporal resolution of the system via the initial segmentation resulted in performance degradation. All ASWU systems compare favorably to the CI phone model system in terms of accuracy. The improved performance of the variable temporal adjustment system over the system that does no intermediate temporal adjustment seems to indicate that selectively allowing the temporal resolution of some segments in the system to increase with increasing contextual resolution results in better model separability. Although some of the performance improvement of the variable temporal adjustment system over the system designed without temporal adjustment can be due to the slightly higher contextual resolution (150 vs. 141 units), it is unlikely that this accounts for the observed performance difference (20% reduction in error rate).

### Thresholding and Variance Estimation

To investigate the effects of different thresholding and variance estimation approaches, three low complexity systems (containing between 124 and 126 units) were trained without use of refinement steps and then evaluated on the February 1989 test set. Two

systems used a covariance estimated on a per utterance basis. Of these two systems, one used the average-likelihood-per-frame thresholding and one used the weighted MDL measure. The third system also used an average-likelihood-per-frame threshold but used a covariance estimated from the complete training set (grand covariance). For all systems, the unit model inventory and corresponding lexicon was derived in 5 stages of alternating divisive and K-means clustering, followed by 3 iterations of Viterbi training. The performance ranged from 75.6% accuracy for the MDL-based system to 76.8% accuracy for both the average-likelihood-per-frame thresholded systems. Thus, we discontinued unconstrained acoustic segmentation with the MDL criterion. The unit inventory for the two average-likelihood-per-frame threshold initial segmentations – per-utterance covariance and grand covariance – were then increased by means of refinement steps to 646 and 631 units, respectively. No temporal resolution adjustments were made during the refinement steps (i.e. word-position-dependent sufficient statistics were computed directly from the Viterbi segmentations). The performance of the grand covariance based system was 84.2% accuracy compared to an accuracy of 85.4% for the per-utterance variance system. Since the per-utterance variance estimation seemed to have a slight advantage at this level, the remaining experiments use that technique.

### **High Complexity Systems**

To investigate whether the ASWU design algorithm is effective for larger unit inventories, three high complexity systems were designed. Two of those systems used discrete temporal adjustment steps but differed in the stages where temporal refinement was performed. The third system used a variable temporal refinement approach. All systems used per-utterance covariance estimation and average-likelihood-per-frame thresholding in the acoustic segmentation steps.

The first system designed a 1385 unit inventory in 4 refinement steps and applied 1 temporal adjustment step. The first intermediate unit inventory of size 124 was

designed using 5 stages of alternating divisive and K-means clustering and 3 iterations of Viterbi training. A second and third intermediate inventory of 342 and 646 units respectively were derived by 5 and 7 stages of alternating divisive and K-means clustering and 2 iterations of Viterbi training. Temporal adjustment by means of segment splitting was then performed on the segmentation derived by the final Viterbi re-segmentation step of the 646 unit inventory. This reduced the inventory size to 635 units after removing clusters with low counts ( $< 100$ ). In a final refinement stage (2 stages of alternating divisive and K-means clustering and 2 iterations of Viterbi training), the unit inventory was increased to 1385 units. The recognition performance on the February 1989 test set of the different unit inventories derived by this refinement process are depicted in figure 3.1. This system will be referred to as the “early” binary temporal adjustment system.

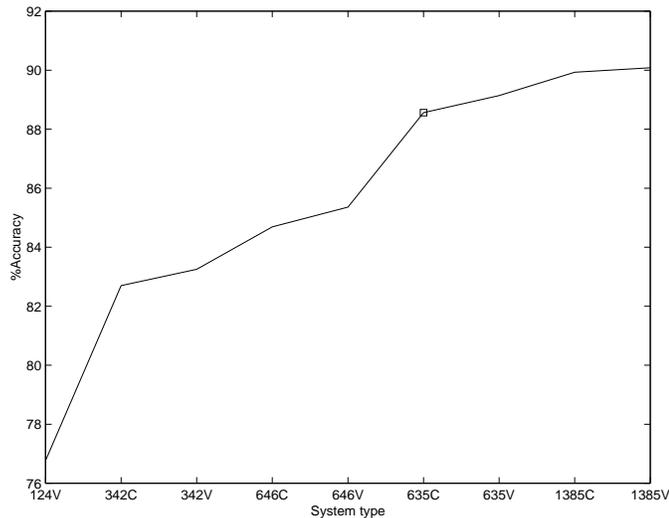


Figure 3.1: Recognition performance on the February 1989 test set using unit inventories and lexicons derived at different stages of progressive refinement using “early” binary temporal adjustment. The square in the figure indicates the stage where binary temporal adjustment was performed. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training.

The second high complexity system was derived from the 646 unit inventory of the first high complexity system. Instead of adjusting the temporal resolution at that point, the second high complexity system increased the unit inventory further to 1147 units by two additional refinement step without adjusting the temporal resolution. The temporal resolution of this 1147 unit inventory system was then increased by a binary temporal splitting using acoustic segmentation and K-means clustering. The resulting inventory, which included 1098 units after low frequency clusters were removed, was then refined using 2 passes of Viterbi training. Recognition results of the different unit inventories derived along this system refinement track are depicted in figure 3.2. This system will be referred to as the “late” binary temporal adjustment system.

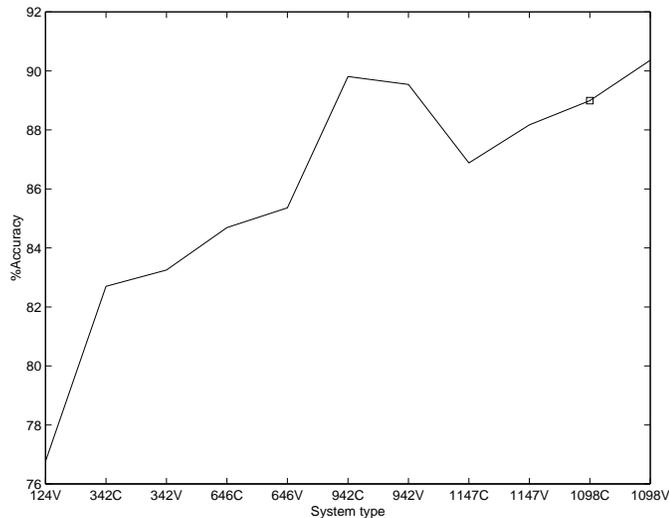


Figure 3.2: Recognition performance on the February 1989 test set using unit inventories and lexicons derived at different stages of progressive refinement using “late” temporal adjustment. The square in the figure indicates the stage where binary temporal adjustment was performed. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training.

The atypical decrease in recognition accuracy when adjusting the segmentation of

the larger unit inventory ( $\leq 942$ ) and increasing the unit inventory further without adjusting the temporal resolution indicate that an increase of the contextual resolution does not result in improved modeling accuracy when not accompanied by an “appropriate” increase in temporal resolution. Note that applying the temporal resolution adjustment earlier (as in the first system design) or later (as applied here) both result in similar performance in terms of recognition accuracy. The performance decrease when increasing the contextual resolution without allowing an increase of the temporal resolution seems to indicate that an improved acoustic modeling accuracy requires a balanced increase in both types of system resolution.

To investigate whether improved acoustic modeling accuracy is obtained by a variable adjustment of temporal detail within the system refinement steps, as for the low complexity case a third system was trained using the variable temporal adjustment refinement approach described in section 3.2.3. In the design of this system, the temporal resolution of the system was adjusted at every refinement iteration. This system with a unit inventory of size 1499 was derived in 4 refinement steps. The acoustic segmentation threshold for the first intermediate inventory of 50 units was set so that the resulting segmentation had on average approximately 1 segment per phone segment. The thresholds used in the subsequent acoustic segmentation steps were chosen so that the resulting temporal resolution of the system remained on average approximately 3 to 4 acoustic segments per phone segment. The first intermediate inventory was obtained by 5 stages of alternating divisive and K-means clustering, followed by 4 iterations of Viterbi training. The subsequent intermediate unit inventories of sizes 150, 743 and 1499 were all derived by 7 stages of alternating divisive and K-means clustering followed by 3 iterations of Viterbi training. The recognition performance on the February 1989 test set of the systems derived at different stages of the progressive refinement process are depicted in figure 3.3.

Comparing the three high complexity systems, the need for appropriate refinement of temporal detail with increasing contextual detail becomes apparent. The improved

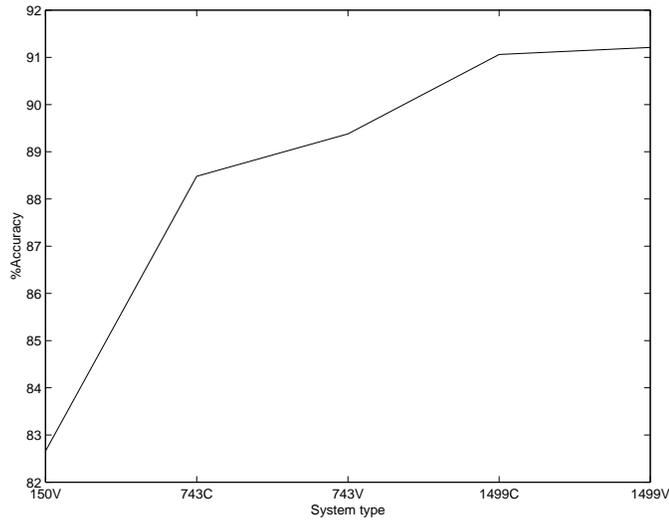


Figure 3.3: Recognition performance on the February 1989 test set using unit inventories and lexicons derived at different stages of progressive refinement using a variable temporal adjustment approach. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training.

performance of the variable temporal adjustment system in comparison to the discrete or no temporal adjustment systems described in this section and in section 3.3.2 also support this observation. The benefit of variable temporal detail adjustment is also supported when comparing the performance of the three high complexity system on the full test set as illustrated in table 3.3, although the performance differences could also be due to the differences of contextual resolution (i.e. the number of free parameters) of the systems.

### Learned Pronunciation Consistency

A number of words in the vocabulary have the same morpheme bases, such as *alert* and *alerts*. Even though the pronunciation of the morpheme base can vary as a function of the affix due to effects such as co-articulation, some consistency in pronunciation can be expected. However, since the pronunciation constraints are applied independently

Temporal adjustment	Inventory size	Full test set performance (% accuracy)
early discrete	1385	88.2
late discrete	1098	89.3
variable	1499	89.6

Table 3.3: Recognition performance on the full test set of the 3 systems derived by different progressive refinement approaches.

for each word, there is no guarantee of pronunciation consistency across words. As one would hope, experiments showed that in many cases the resulting pronunciations are in fact quite similar, particularly for the low complexity systems. Some examples of words with equal morpheme bases and their pronunciation in terms of units from the low complexity system are given in table 3.4.

ALERT	U9	U104		U47	U115	U21	U112	...
ALERTS	U9	U104	U25	U47	U115		U112	...
LETTER	U79	U47	U115	U66	U104	U87	U103	...
LETTERS	U79	U47	U115	U66	U104	U87	U103	...
NINETEEN	U123		U33	U115	U66		U133	...
NINETEENTH	U123	U36	U33	U115	U66	U75	U133	...
NINETY	U123		U33	U115	U66		U133	...

Table 3.4: Pronunciation examples of words with the same morpheme base.

The similarity of the pronunciations of words with the same morpheme base might suggest that these automatically learned units are similar to phones. In fact, they are similar to phone *states* in an HMM. For that reason, it is difficult to look for a

mapping between phones and ASWUs. The only meaningful comparison is between phones and sequences of ASWUs, but in that case the ASWU sequences are likely to be allophones. However, it is straightforward and interesting to look at the variation in the number of ASWUs that map to a particular phone in different contexts. Figure 3.4 illustrates the range of ASWUs per phone, for a high complexity system (comparable to a context-dependent phone system) with an average number of 3.9 states per phone. The results show that, as expected, the number of ASWUs per phone varies substantially, which is roughly equivalent to variable allocation of the number of states per phone depending on context. This more flexible (and therefore more efficient) mechanism for representing variation as a function of time is a key reason for the improved performance of the ASWU systems.

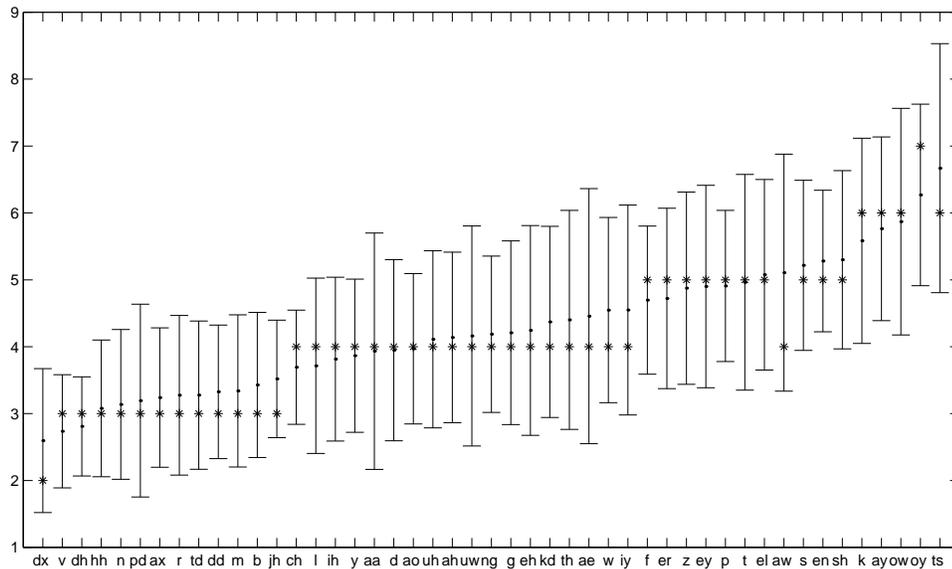


Figure 3.4: Range of the number of ASWU units that map to the different phones when the average is 3.9 states/phone. A (\*) indicates the median number of units per phone and the bars indicate the range of one standard deviation about the mean. Phone labels use the DARPAbet standard, except that /-d/ indicates an unreleased closure and /ts/ is a /t/-/s/ sequence.

### 3.4 Summary and Conclusions

In summary, the proposed algorithm approaches the problems of automatic unit and lexicon design as a joint problem ensuring a matched condition between unit models and the lexicon. As in previous work, a unit inventory is designed by an acoustic segmentation step followed by a clustering step. A joint solution is obtained by constraining the unit design algorithm to guarantee a limited complexity of the pronunciation model. Two constraints are imposed in the design, with the objective of finding a single linear pronunciation per word. First, a pronunciation length constraint enforces uniformity in the number of segments across training tokens of a word. Second, a pronunciation consistency constraint ensures that segments in the same word position of different tokens are assigned the same unit label. A final unit inventory and lexicon are designed by progressive refinements, alternating expansion of the unit inventory by clustering with adjustment of the segmentation by Viterbi training. The refinement step also provides several approaches to controlling the temporal resolution throughout the system design.

Comparing the performance of the proposed automatic unit inventory and lexicon design algorithm at low complexity, the 141 and 150 ASWU systems outperform the 145 state phone-based system (19% error reduction improving accuracy from 75.6% to 80.3%, significant with 95% confidence). The 124-unit ASWU system performs comparably to the 128-unit system described in [28] and to the 145-state CI phone-based HMM system. At high complexity, the performance of the ASWU systems with 1098 and 1385 states is comparable to the triphone systems with 1100 and 1514 states, respectively. The 1499 ASWU system is slightly better than the 1514 phone system (89.1% vs. 89.6%). Since the ASWU system is in fact an HMM, further improvements in performance can be obtained by using mixture distributions, as in [72].

The comparable performance of the low complexity system with the results of Holter and Svendsen [28] demonstrates that the constrained unit design approach

is competitive with previous ASWU work. The performance at high complexity, lightly better than a triphone system, shows the success of the algorithm for the design of large unit inventories, which is an important advance on previous work. While the results reported here are not the best reported on this task because they are based on single Gaussian distributions, since the resulting model is simply an HMM it is straightforward to introduce mixture distributions [72], which would undoubtedly lead to or beat state-of-the-art performance. It might also be possible to improve performance by increasing the number of units, which were constrained here for comparison to phone-based systems. Given that phone-based systems represent the state-of-the-art in read speech tasks, especially at high system complexity, the slightly better performance of the ASWU-based system shows the viability of the proposed algorithm.

Several variations of the ASWU system were explored, yielding two main conclusions. First, the use of an average-likelihood-per-frame thresholding approach in the acoustic segmentation stage performed comparably or better than using weighted MDL thresholding, and the use of a per-utterance estimated covariance gave slightly better performance than a grand covariance estimate. Second, allowing temporal resolution adjustments guided by acoustic segmentation information (providing both new segment boundaries and definition of the amount of temporal adjustment) at each stage in the system refinement process yields better performance than allowing no temporal resolution adjustments or temporal resolution adjustments without acoustic segmentation guidance.

A problem of the proposed algorithm is that it requires several training tokens for each word. In order to understand how much data is “sufficient” for ASWU pronunciation modeling, the best case system was analyzed to determine the relation between training token counts and two different indicators of pronunciation “goodness”. First, the relation between the percentage correspondence between pairs of words with the same morpheme base (such as the examples in Table 3.4) and training token count

was modeled. Second, the training token count as a function of the relative error in the February 1989 test set was modeled. A linear model gave a good fit for both relationships (correlation of 0.97 and 0.85 for the two regression models respectively), and the data did not show a clear breakpoint for use as a training token count threshold. Thus, the trade-off point between phonetic and automatically-derived units should be assessed experimentally.

Another possible limitation of the algorithm described here is the assumption of a single linear pronunciation per word. Certainly the use of mixture distributions, which can be incorporated after the initial unit/lexicon design step, can compensate for some pronunciation variability, as in phone-based HMM systems. However, it may be useful to explicitly represent pronunciation variants, as discussed further in chapter 6. Further modeling accuracy improvements resulting from explicitly modeling effects of cross-word context are addressed in chapter 4. Within-word pronunciation variation can be successfully modeled with ASWUs, as demonstrated by Holter and Svendsen [29]. For the approach described here, a straightforward extension of the temporal unit splitting algorithm to contextual unit splitting would allow for multiple pronunciations within the context of unit design with pronunciation constraints, assuming a redefinition of atomic units.

# Chapter 4

## Explicit Context Modeling

In phone-unit-based systems, high acoustic modeling accuracy can be obtained by explicit modeling of context. The accuracy improvement over modeling with context-independent units can be attributed to providing more detail of the acoustic space by allocation of a larger number of parameters as well as constraining the allowable sequences of the more detailed models. The sequence constraint accounts for the co-articulation effects that occur in continuously spoken speech. As described in chapter 3, improved modeling accuracy can be obtained within the ASWU framework by modeling of context in an implicit fashion. As the lexicon, unit inventory and unit models are designed jointly, both the choice of unit and its model can change with increasing system complexity providing a mechanism for incorporating contextual effects dependent on word position. This approach can capture contextual effects from the units within a word model but cannot capture context effects due to units in neighboring words (cross-word effects). To arrive at a model that does allow modeling of cross-word effects the context needs to be represented explicitly. Approaches that use an explicit context representation are described in this chapter. By modeling automatic unit context explicitly, techniques shown to be effective for phonetic units can be applied to automatic units.

Shared distributions for the units in explicit context can be derived using ap-

proaches analogous to those used in phone-unit-based systems. Section 4.1 describes how shared distributions can be derived by either agglomerative or decision-tree-based clustering techniques.

Section 4.2 discusses different approaches to defining the “context” of automatically derived units. In contrast to phonetic units which generally have complex state topologies, the automatically learned units are represented by single state models making the definition of “phone-like” context an open question. The problem is non-trivial, since using the label of a single state to represent context is too local in time [27], but a specific sequence of states is too detailed.

The use of decision tree distribution predictors for context-dependent units is desirable since it provides a mechanism that provides distributions for units in contexts, not observed in the training data. This ability to generalize allows the explicit context modeling framework to be extended to contextual effects across words. The techniques developed for phone-unit-based systems that provide this generalization are based on phonetic knowledge and are therefore not directly applicable to automatically learned units. To use the decision tree approach, an algorithm, described in section 4.3, was designed that attempts to learn groups of ASWUs that provide equivalent contextual effects allowing the definition of a model that provides the desired generalization to unseen contexts.

Experiments using the proposed algorithms are described in section 4.4. A summary and conclusions are given in section 4.5.

## 4.1 General Clustering Methods

Clustering context-dependent ASWUs can be framed as analogous to the context-dependent phone units described in section 2.2.2. The available approaches can be grouped as either agglomerative or divisive. The agglomerative approaches start by allocating a distribution for every unique observed context-dependent unit and

deriving the shared distributions by merging similar distributions. The divisive approaches start from one model for all unique contexts and progressively divide the unique contexts into groups, deriving a separate distribution for each group. The divisions considered in the divisive clustering process are either obtained from the data itself (i.e. any division of a pool of contexts can be considered) or constrained to subsets defined using knowledge of the relationship between units. In addition, a common constraint applied to modeling phonetic units in context is to only allow sharing among context-dependent versions of the same center phone and same state position with the unit. An important design issue that applies to both the agglomerative and divisive approaches is to what extent the clustering process should be constrained in considering sharing scenarios.

The advantage of the unrestricted clustering approaches is that they consider a larger number of sharing scenarios: *any* possible sharing scenarios may be considered. The implicit modeling of context described in the progressive refinement algorithm in chapter 3 used unrestricted divisive and K-means clustering to find the shared distributions. The main disadvantage of that approach lies in the fact that the resulting shared distributions do not generalize to unseen context-dependent models. To allow modeling of contextual effects across word-boundaries, the problem of generalization must be addressed. The lack of generalization to unseen contexts is present in any clustering approach that derives possible data divisions from data (e.g. [14]). An additional advantage of constrained clustering is that the computation can be reduced even further if additional constraints on sharing scenarios are imposed, such as the restriction to allow sharing among different context-dependent models of the same center unit alone.

In summary, the main advantage of using a divisive approach together with a pre-defined set of classes is that the shared distributions obtained by the clustering step generalize to unseen contexts. The classes define which groups of units are correlated and provide the mechanism to generalize to unseen contexts. The disadvantage of

the decision tree clustering approach is that the sharing scenarios are limited to those allowable by the class definitions. It is therefore important to provide this clustering approach with a set of classes that appropriately represent the correlation between the contextual effects of units on neighboring units in order to obtain a system capable of the desired generalization to unseen contexts. An algorithm to automatically learn such classes is described in section 4.3.

## 4.2 Defining Long Distance Context

The simplest definition of context is in terms of neighboring units, as a triphone is defined in terms of the left and right phonetic neighbors. However, the ASWUs were designed on the basis of stationary segments found in the speech signal, so that they could be appropriately modeled by a single state HMM. Explicit context modeling using these units directly will therefore not be truly analogous to modeling phonetic units in explicit context as the phonetic unit models are generally more complex and are represented by multi-state HMMs. Deriving emission distributions that are dependent on the neighboring ASWUs is therefore based on a more local context compared to phonetic model states in phone unit context.

One approach to incorporating longer span contextual effects is by considering a larger number of neighboring ASWUs than is generally considered in a phonetic unit system. For example, one could use two neighbors to the left and right, i.e. called quin-units, unfortunately, even quin-units do not cover the same contextual “distance” as a triphone, and direct incorporation of larger, specific sequences of neighboring units is infeasible because the number of unique “contexts” becomes too large. It is therefore required to find groups of sequences that map to a single context, just as different state sequences of phonetic models map to a single phone. Grouping the large number of unique contexts is also needed for the search process when considering cross-word effects. As described in section 2.3, incorporation of cross-word contextual

effects will lead to an increase of the state-space proportional to the number of unique contexts. If a larger number of neighboring units are considered and these sequences are not grouped, the increase of the state-space will make the search computationally infeasible.

Another approach to the definition of context is to use the fact that the automatic units were derived in progressive refinement steps and that units derived at different stages of this refinement process are hierarchically related. Note that use of this hierarchical relationship only provides longer span context information if temporal adjustments are made. The temporal adjustment step by means of a constrained acoustic segmentation provides a temporal structure relating a coarse level to finer levels. A single word position at the coarse level can correspond to a sequence of word-positions at a finer level. If a segment corresponding to a coarse level word-position is split into  $S$  segments by the constrained acoustic segmentation and different distributions are used to represent the  $S$  finer segments, the single word-position at the coarse level has an  $S$ -length unit sequence corresponding to it at the finer level. Note, however, that all unit design is based on word-position and not on unit identities, leading to a hierarchical but non-deterministic relationship between the fine and coarse levels. In other words, a unique coarse-level unit will have several fine-level sequences as descendants, and a single fine-level unit can correspond to one of several coarse level parents. Considering all word-positions that are represented by coarse level unit  $X$ , the corresponding fine level unit sequences will be dependent on the word-positions in which they occur and not the fact that they are descendants of coarse unit  $X$ . In comparison to the phone-unit-based system, the phone units can be considered as the coarse level units which have states as fine level descendants. The difference between the phone-unit-based system and the ASWU-based system is that a triphone state maps to a unique phone unit whereas a fine-level ASWU can map to several coarse-level ASWUs. In that sense, the phone unit to phone state relationship is hierarchical and deterministic, whereas the coarse to fine ASWU relationship is hierarchical but

non-deterministic. To illustrate this, a possible alignment of observations with the fine and coarse level units is depicted in figure 4.1. The context of a fine-level ASWU can therefore be defined in terms of the neighboring coarse-level ASWUs, analogous to triphones, but can in addition be dependent on the center coarse unit as well, even if no context sharing across the center fine-level unit is allowed. In most phone-based systems context sharing across center phones is not allowed and modeling dependency of phone states on the center phone does not provide any additional information as the relationship between phone state and phone is deterministic.

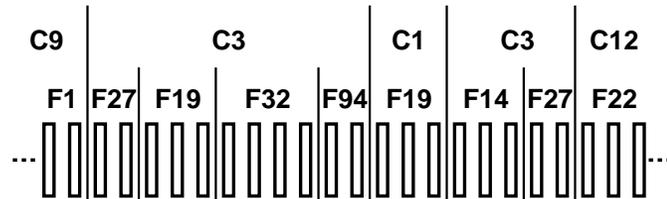


Figure 4.1: A possible alignment of observations (feature vectors depicted as rectangular boxes) with a hierarchical and non-deterministic coarse and fine level unit sequence.

Another difference between the ASWU-based-system and the phone-unit-based system is the dependency of the temporal resolution of the system on the contextual resolution of the system. As described in chapter 3, temporal resolution is lost if neighboring segments, found distinct in an acoustic segmentation, are found equal after quantization as the different segments are then represented by the same model of stationarity. The dependency is due to the fact that segments merged at a coarse level of quantization may not have been merged at a finer level of quantization. When explicitly modeling context, the temporal resolution is fixed to that of the context-independent base unit inventory (the fine-level unit inventory). This is in contrast to the implicit context modeling approach (the high complexity systems described in chapter 3) as there, temporal adjustment could be made at refinement steps. When explicitly modeling context, the dictionary is fixed in terms of the

*number* of (fine-level) units. The design of the explicit context system will derive a set of distributions shared by the context-dependent fine-level unit models (i.e. will adjust the contextual resolution of the system) but will maintain the temporal resolution as defined by the base (context-independent) fine-level unit inventory. As it was found that a lack of temporal resolution can lead to decreased performance, the context-independent system that is chosen as the basis of the context-dependent models should provide sufficient temporal resolution to accompany the contextual resolution refinement obtained by the explicit modeling of context.

### 4.3 Learning Context Classes

In contrast to phonetic units where class definitions can be obtained from the knowledge of phonetics, no such information is available if an automatically designed unit inventory is used. In the algorithm proposed here, the unit groups providing the desired generalization are learned from data in a separate parallel clustering step. The shared distributions for the context-dependent ASWUs are then obtained in a divisive clustering step that uses the groups learned in the separate clustering step. The distributions for the context dependent unit models can therefore be seen as derived by a four stage clustering process. The first clustering step defines the coarse level unit inventory. The second clustering step derives the fine-level unit inventory that is hierarchically related to the coarse level unit inventory. The groups of coarse-level units that provide equivalent contextual cues are learned in the third clustering step. Then, the distributions for the fine-level units in coarse unit contexts are derived by a divisive clustering step in which the unit groups, learned in the third clustering step, are used.

The unit groups are learned from data by running several clustering processes in parallel. One K-means clustering process operates on all the observed context-dependent units from the same fine-level center unit and divides those observed con-

texts in groups, with the constraint that the groupings must be the same for all K-means clustering processes (i.e. for all fine-level center units). The collection of all parallel K-means clustering processes divide the pool of all observed contexts into groups. In other words, if there are  $Q$  unique units in the base fine-level unit inventory,  $Q$  K-means clustering processes will be run in parallel. The parallel K-means clustering processes are constrained to have the same number of clusters and the same partition of unique contexts over the cluster inventories at all time. For example, if base unit  $A$  as well as  $B$  was observed in context  $X$ , the constraint imposes that these data are assigned to the corresponding clusters in the parallel clustering processes of  $A$  and  $B$ . Each corresponding cluster in the parallel K-means clustering processes will therefore contain the same group of contexts. Note that units quite possibly are not observed in all contexts but due to the constraints between the parallel K-means clustering processes, the unobserved contexts can still be assigned to one of the clusters. The links between the parallel K-means clustering processes provide the ability to learn general groups of units, since knowing that a unit falls into one of these groups is an indicator that its contextual effect on various center units is similar to that of others in the group.

Another way of viewing the parallel clustering step is to consider it a single K-means clustering process where each cluster is represented by a  $Q$ -dimensional vector of  $d$ -dimensional vector model parameters. Each element of a vector cluster representative points to the shared model for a center unit in the contexts that are grouped in that cluster. The data that is partitioned in this vectorized K-means clustering process is also represented by vectors, where each vector represents the observations (via sufficient statistics) of one unique context. Each of the  $Q$  elements of a datum vector corresponds to the sufficient statistics of the data associated with one unique center unit in that one context. If the inventory of all observed contexts (regardless of the center unit) is denoted as  $U$ , then the training data can be seen as a sparse matrix of dimensions  $Q$  by  $U$ . The  $(i, k)$ -th element of that matrix is the data of the

$i$ -th unit in the  $k$ -th unique context. The data matrix is sparse because not all units were observed in all contexts. The learning of context groups is based on K-means clustering, which involves iterating between partitioning and re-estimation steps. The partitioning step assigns each column of that matrix (representing the data from a unique context) to the minimum distance cluster (representing the context groups) using the negative log-likelihood distance as in chapter 3 and is described next. The unobserved contexts can still be assigned to a context group on the basis of other center units that *were* observed in that context. The re-estimation step estimates distribution parameters independently for each of the  $Q$  units. An illustration of the vector clustering of all unique contexts into two context groups is depicted in figure 4.2.

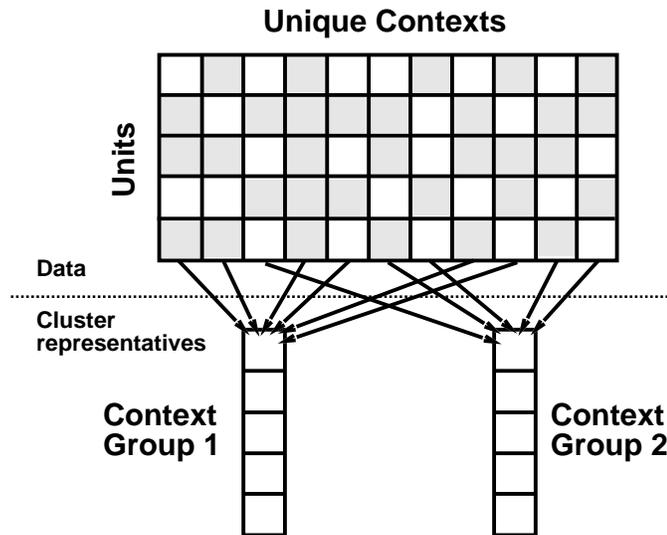


Figure 4.2: Clustering unique contexts in two context groups. Shaded squares correspond to observed contexts.

As for any K-means clustering procedure, we iterate between a partitioning and a re-estimation step. For partitioning, we compute the distance of a datum with respect to a cluster representative by assuming the  $Q$  elements of the vector are independent. In other words, we sum the negative log likelihoods of the center-unit

(context-dependent) data in each element of the vector with respect to the corresponding center-unit (context-dependent) unit model in the vector cluster representative. Let the data from context  $C \in \{1, 2, \dots, U\}$  be denoted as  $\mathbf{Y}^C = [\mathcal{Y}_1^C, \mathcal{Y}_2^C, \dots, \mathcal{Y}_Q^C]$  where  $\mathcal{Y}_x^C$  denotes the sufficient statistic for the observations of unit  $x$  in context  $C$  and possibly  $\mathcal{Y}_x^C = \emptyset$ . Let the cluster representative of cluster  $M$  be denoted as  $\mathcal{M} = [\{\mu_1^M, \Sigma_1^M\}, \{\mu_2^M, \Sigma_2^M\}, \dots, \{\mu_Q^M, \Sigma_Q^M\}]$  where  $\{\mu_p^M, \Sigma_p^M\}$  denote the mean and covariance of the model of the  $p$ -th unit in the  $Q$ -dimensional vector of models. The distance of the data with respect to the cluster representative is then computed as

$$\mathcal{D}(\mathbf{Y}^C | \mathcal{M}) = - \sum_{\substack{i=1 \\ \mathcal{Y}_i^C \neq \emptyset}}^Q \mathcal{L}(\mathcal{Y}_i^C | \mu_i^M, \Sigma_i^M), \quad (4.1)$$

where  $\mathcal{L}$  is defined as in equation 3.5. Given a partition of the data, the ML parameter estimates of the models in a cluster representative can be computed from the sufficient statistics vectors assigned to that cluster using equations 3.6 and 3.7.

Progressively smaller context groups can be derived by alternating binary divisive and K-means clustering. It is desirable to obtain different sized groups as this is analogous to the groups used in context-dependent phonetic unit clustering (e.g. specific phones vs. phone classes) and there may be different factors to capture (e.g. manner vs. place of articulation). In the divisive stages, an approach similar to the one used in the automatic unit design algorithms described in chapter 3 is used. Let the data  $\mathbf{Y}^S$  be assigned to the cluster that is to be split and let that cluster be represented by model vector  $\mathcal{M}$ . The selection of the cluster that is to be split is based on the average likelihood per frame of the data assigned to a cluster. Initially, two new cluster representatives  $\mathcal{M}_1$  and  $\mathcal{M}_2$  are derived from  $\mathcal{M}$  by perturbing the means of the unit models that represent the elements of  $\mathcal{M}$ . The data  $\mathbf{Y}^S$  is then partitioned in  $\mathbf{Y}_1^S$  and  $\mathbf{Y}_2^S$  ( $\mathbf{Y}^S = \mathbf{Y}_1^S \cup \mathbf{Y}_2^S$  and  $\mathbf{Y}_1^S \cap \mathbf{Y}_2^S = \emptyset$ ) by iteratively assigning each datum to either  $\mathcal{M}_1$  or  $\mathcal{M}_2$  based on the distance defined above and re-estimating  $\mathcal{M}_1$  and  $\mathcal{M}_2$  by ML estimation from the data partition. After completion of one or more divisive stages, K-means iterations are run to obtain the

context groups. Iteration of the divisive and K-means stages allows the derivation of progressively smaller groups.

Data sparsity complicates the “vector clustering” process. In the repartitioning steps it is required to compute for a datum  $\mathbf{Y}^n$

$$c_n = \underset{x}{\operatorname{argmin}} \mathcal{D}(\mathbf{Y}^n | \mathcal{M}_x) \quad (4.2)$$

If  $\mathbf{Y}^n$  has an element  $\mathbf{Y}_j^n = \emptyset$ , then that element will not contribute to any of the computed distances and will not affect the assignment of that datum to a cluster. After completion of the data partitioning step, in the re-estimation step, that element of that datum will not contribute to the new cluster representative  $\mathcal{M}_{c_n}$ . It is possible that there is no or very few data points for the  $j$ -th element of  $\mathcal{M}_{c_n}$  in the data set assigned to that cluster. In that case, the model of the cluster representative of the parent group is used in the divisive stages or the context-independent model is used in the K-means stages.

## 4.4 Experimental Results

To investigate the effects of modeling context explicitly in the ASWU framework, experiments using the different base unit inventories were conducted. The base inventories were those designed as described in section 3.3.2. All experiments were conducted on the Resource Management corpus, testing system performance on the February 1989 test set unless otherwise stated. The experiments are divided in three series, each investigating different aspects of modeling context explicitly. In the first series, described in section 4.4.1, the effects of modeling local context of the ASWUs in a word-internal system are described. Then, in section 4.4.2, the effects of modeling a more distant context are investigated, again limiting the scope to word-internal contextual effects. The performance of systems using a decision tree-based distribution predictor, allowing the generalization required to model contexts across word boundaries, is described in section 4.4.3.

### 4.4.1 Local Context Experiments

Four experiments were conducted to investigate the effect of modeling local context: the performance of a system modeling local context explicitly, the effect of constraining sharing scenarios to be limited to within the same center unit, the effect of starting from different sized base unit inventories (i.e. testing the effect of the temporal resolution of the system) and the effect of increasing the context span. In addition, because these variations are associated with different numbers of atomic units, it is also possible to look at the effect of increasing the degrees of freedom in clustering. In all the experiments here, complexity is increased using successive stages of data-driven divisive and K-means clustering.

The first experiment used the 124 unit inventory, as the base unit inventory. An explicit context of the neighboring left and right unit was incorporated and the resulting context-dependent units will be referred to as “tri-units”. Atomic group sufficient statistics were computed for these context-dependent units for use in subsequent clustering. This yielded 6.3k sufficient statistics (there were 6.3k unique observed units in context). Starting with a cluster inventory represented by the 124 context-independent units, the number of shared distributions for the tri-units was increased in progressive refinement steps analogous to the implicit context modeling approach (as described in section 3.2.3). Clusters with less than 100 observations were removed in the ML clustering steps. Figure 4.3 shows the recognition performance of the shared distribution systems derived at different progressive refinement stages. The atypical decrease in performance after Viterbi training the largest unit inventory seems to indicate that the system gets over-tuned to the idiosyncrasies of the training data rather than learning patterns that generalize to unseen test data. The best performing system uses 1519 distributions and achieves an accuracy of 86.3%, which is much lower than the comparable (1514 distributions) agglomeratively clustered word-internal triphone system (90.2% accuracy on the same test set).

In a second experiment, the allowable parameter sharing configurations were lim-

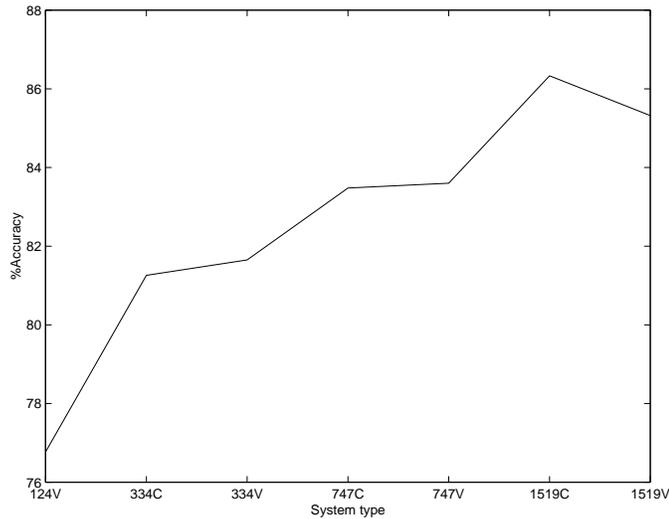


Figure 4.3: Recognition performance of the shared distribution, local word-internal, tri-unit unit inventory starting from a low complexity (124 unit) base unit inventory. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training.

ited so that distributions could only be shared among context-dependent units stemming from the same context-independent base unit, i.e. sharing was constrained to different context-dependent versions of the same unit. To allow a non-uniform allocation of free parameters across the different base units, the number of shared distributions for each of the units was determined by an average likelihood per frame threshold and minimum observation occupancy. A separate divisive clustering run was performed for every unique base unit until, for each cluster, the data was either modeled with an average likelihood per frame exceeding the set threshold (empirically determined), or the number of occupancies had fallen below the minimum (100). The final set of distributions was then derived by K-means clustering, removing clusters with occupancies below the set threshold. Again, the 124 unit inventory was used as the base unit inventory. 1262 shared distributions were derived by center-unit tied tri-unit distribution clustering. The number of distributions per base unit ranged from 1 to 34 with a median of 10. The recognition performance of the clustered unit

inventory was 84.6% accuracy which improved to 84.8% accuracy after 3 iterations of Viterbi training.

Extrapolating based on inventory size differences, the performance of the systems in these two experiments is comparable. Thus it appears that the constraint on the allowable sharing scenarios does not affect performance. The initial data partition that is imposed by the constraint reduced the computational cost of the distribution clustering step by approximately a factor of 6. Comparing the performance of these explicit context models to that of systems with comparable complexity that model context implicitly (see section 3.3.2), the implicit context modeling systems outperform the explicit context systems (91.2% accuracy vs. 86.3% accuracy). However, the explicit context systems described so far were based on a low-complexity base unit inventory resulting in two important differences in the degrees of freedom of the clustering process. First, the small unit inventory reduces the granularity of the training data representation compared to the implicit context modeling which uses a granularity defined by word-position (24k sufficient statistics for the 1385 unit, implicit context modeling system vs. 6.3k sufficient statistics for the explicit tri-unit system starting from the 124 base unit inventory). As the data is represented by many fewer sufficient statistics, sharing scenarios are more constrained than those considered in the implicit context modeling case. Second, the choice of a low complexity unit inventory as the base units results in a low temporal resolution system due to the relationship between contextual and temporal resolution.

To investigate the impact of these limitations, in a third experiment, the 635 unit ASWU system was used for the base unit inventory providing more temporal resolution (this unit inventory was derived by a binary temporal adjustment step) and a larger number of tri-unit sufficient statistics (13k). The performance as function of progressive refinement steps are depicted in figure 4.4. Again, increasing and refining the unit inventory initially leads to an accuracy increase but results in over-tuning to the training data when increased further. The best performance of this system

(90.0% accuracy) is slightly worse than the best performance of the implicit context modeling system (91.2% accuracy).

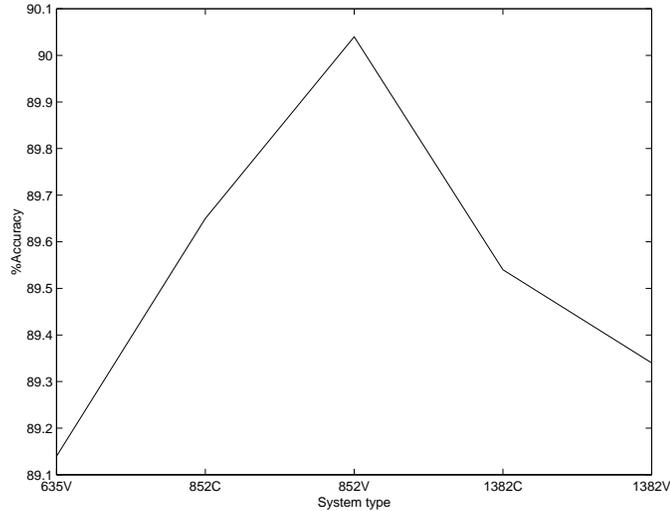


Figure 4.4: Recognition performance of the shared distribution, local word-internal tri-unit inventory starting from a high complexity (635 unit) base unit inventory. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training.

Whether performance could be improved further by increasing the context span granularity of the data representation, is explored in a fourth experiment. The context window was expanded to include the two neighboring units to the right and left; these context-dependent units will be referred to as “quin-units”. Again, the 635 unit inventory was chosen as the base units and the granularity of the data representation in terms of unique quin-unit sufficient statistics was 20k, still a smaller number of sufficient statistics than for the word-position dependent system but larger than for the tri-unit system. Using progressive refinement, the number of shared distributions was increased and recognition accuracy as function of the stage in the progressive refinement process is depicted in figure 4.5.

The increased performance of this system in comparison to the tri-unit-based system indicates that additional modeling improvements can be gained, either from

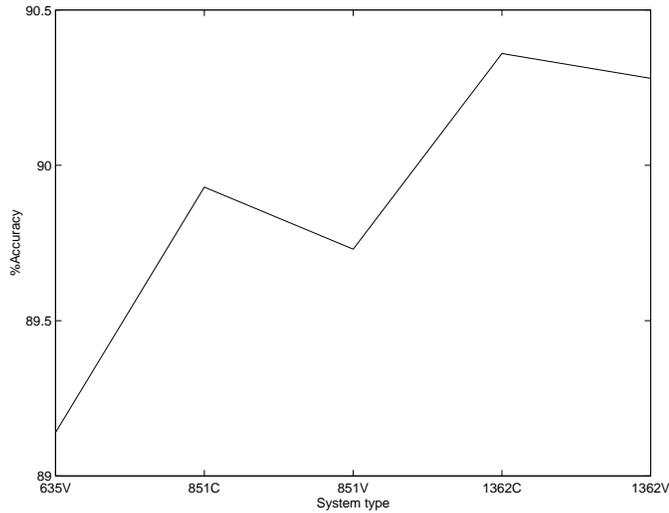


Figure 4.5: Recognition performance of the shared distribution, local word-internal, quin-unit inventory starting from a high complexity (635 unit) base unit inventory. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training.

considering a more distant context or by increasing the granularity of the training data representation presented to the clustering step. The effect of modeling a more distant context is explored in section 4.4.2.

An overview of the best case local context results are given in table 4.1. The results seem to indicate that increasing the number of sufficient statistics to represent the data leads to increased performance up to a point (for this task roughly 20k), and that a further increase beyond that point does not seem to yield improved performance.

#### 4.4.2 Distant Context Experiments

Analogous to the type of explicit context modeling used in phone-based systems and as suggested by the experiments using explicit modeling of local context, it may be beneficial to consider a more distant ASWU context rather than representing context

# base units	context window	constraints	# unique CD units	# shared distributions	recognition performance (% acc.)
124	3	none	6.3k	1519	86.3
124	3	center unit	6.3k	1262	84.8
635	3	none	13k	852	90.0
635	5	none	20k	1362	90.4
635	word	none	30k	1385	90.1

Table 4.1: Best case results for the explicit local context systems, which can be compared to 90.1% accuracy for the implicit context modeling with 1385 distributions.

in terms of the directly neighboring ASWUs. To investigate this type of context definition, the unit inventory derived by variable temporal adjustment of size 743 as described in section 3.3.2 was used as the context-independent base unit inventory. The more distant unit context was provided by the hierarchical relationship between the *fine* units in the 743 unit inventory and the *coarse* units in a 50 unit inventory derived in a previous clustering stage. The average duration of the units in the 50 unit system was approximately that of phone-units. Each unique fine unit was modeled in explicit context defined as the coarse unit it descended from (the center coarse unit) as well as the coarse units neighboring to the left and right. The scope of the context was more distant than that described in section 4.4.1 as the context is not necessarily defined as the directly neighboring state-like units but was still limited to within the word (word-internal). Given this context definition, 18k unique context-dependent units were found for which shared distributions were derived by means of progressive refinement. The recognition performance of these shared unit inventories is depicted in figure 4.6 as a function of the progressive refinement stages.

Even though the number of sufficient statistics is smaller than that of the best case (“quin-unit”) system using the local ASWU context (18k sufficient statistics vs.

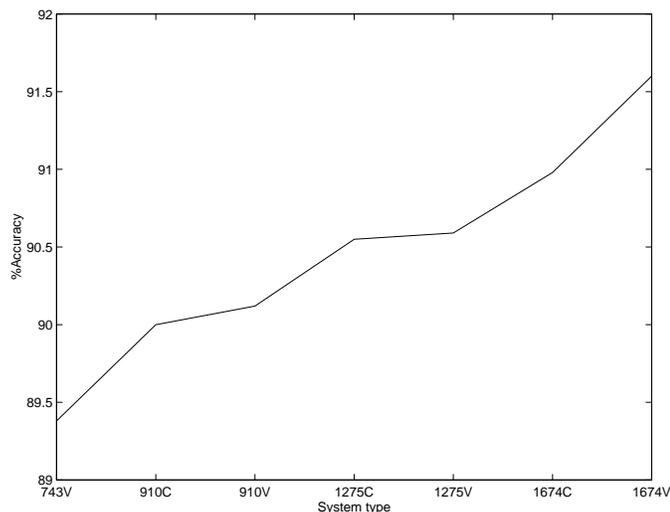


Figure 4.6: Recognition performance of the shared distribution, distant word-internal context inventory based on a high complexity (743 unit) unit inventory. System type indicates the number of units and is appended with either a C for the system after clustering or a V for the system after Viterbi training.

20k sufficient statistics), the definition of a more distant context seems to result in improved recognition accuracy (91.6% accuracy vs. 90.4% accuracy). Note though that this could also be due to better modeling of the temporal resolution of the base unit inventory (variable temporal adjustment vs. binary temporal adjustment used as the basis of the quin-unit system).

### 4.4.3 Cross-Word Context Modeling

As the first step towards widening the scope of the model to incorporate contextual effects across word-boundaries, equivalent context classes were learned from data using the algorithm described in section 4.3. The same 743 base unit inventory and distant context definition (considering contextual effects across word boundaries) as in section 4.4.2 was used. Three sets of equivalent coarse-level context classes were derived for the center, left and right context position by means of 50 stages of

incrementing the number of context classes by divisive clustering and re-estimating those context classes by K-means clustering. At every stage, the context class with the lowest average likelihood was split and the contexts held in that class were divided over the two new classes. Then all context partitions were considered by execution of the K-means algorithm using the distance measure and re-estimation formulae as described in section 4.3. The number of different context classes (groups as well as individual coarse level units) found during this process were 88, 100 and 98 for the center, left and right context positions respectively.

Using the context class definitions, two sets of decision-tree-based distribution predictors were designed (a separate decision tree was created for each unique fine-level center unit). One set of decision trees for predicting distributions of a word-internal system and one set for predicting distributions of a cross-word system. The decision trees were designed using the data represented by word-internal or cross-word sufficient statistics. In both cases, the decision trees were grown using a greedy algorithm and an ML objective function. Up to 1583 distributions were obtained at the tree leafs. The recognition accuracy of the word-internal system using the clustered distributions was 91.2% accuracy and improved to 91.6% accuracy after 3 iterations of Viterbi training. This is equivalent to the performance of the 1499 distribution agglomeratively clustered system described in section 4.4.2. The recognition accuracy of the cross-word system using the decision-tree-based clustered distributions designed on the cross-word sufficient statistics obtained an accuracy of 92.1% accuracy which did not improved by Viterbi training.

## 4.5 Summary and Conclusions

In summary, several options for explicit modeling of context within the ASWU framework exist. One option is to explicitly model context in terms of the directly neighboring ASWU units (modeling local context). It was found that the temporal resolution

of the context-independent base unit inventory has a large effect on the accuracy of the resulting context-dependent system. It was also found that constraining parameter sharing to be limited to context-dependent versions of the same context-independent base units does not affect recognition performance significantly. Experiments also indicate that modeling a more distant context leads to more separable models. Within the local context modeling approach, performance improved when a longer context window was considered. Further performance gains were obtained by considering a more distant context, which was defined by taking advantage of the hierarchical relationship between units in the inventories derived at different stages of the progressive refinement. Using this type of context definition, groups of base-units that exhibit equivalent effects when appearing in the context of another unit were automatically derived by means of a vector clustering process. When using these unit groups in the design of a decision tree distribution predictor, equal recognition accuracy was obtained for the decision tree and agglomerative clustered systems. When the decision trees were used to generalize to unseen contexts and the system was used in a cross-word setting, a 6% reduction in the word error rate was observed.

As in the implicit context modeling case, the temporal resolution of the system is an important factor as explicit modeling of context using a low temporal resolution system leads to worse performance than explicit modeling of context starting from a temporally finer system.

The algorithm for derivation of groups of units functioning as equivalent context cues seems successful as the greedily designed decision tree distribution predictor leads to performance comparable to that of the agglomerative clustered system. Although the sharing scenarios considered in the decision tree clustered system are much fewer and constrained by the defined equivalence classes, the accuracy of the resulting systems is equivalent to that of the agglomeratively clustered system that allowed all parameter sharing scenarios. Using the decision-tree-based system in a cross-word setting, an additional performance gain was obtained indicating that the learned

equivalence classes seem to provide the desired generalization to unseen contexts.

In comparison to phone-unit-based systems, the ASWU systems modeling context explicitly outperform context-dependent phone-based systems when the scope is limited to word-internal contextual effects: 91.6% and 90.2% for the best ASWU-based and best phone-based systems, respectively. However the phone-based systems give a small gain over ASWU-based systems when cross-word contextual effects are incorporated into the model (92.7% vs. 92.1%).

# Chapter 5

## Use of ASWUs in a Large Vocabulary System

This chapter describes the design of a large vocabulary continuous speech recognition (LVCSR) system for spontaneous conversational speech and describes an approach to incorporate automatically derived units into such a system. Although the described algorithms are directly suitable for application in a large vocabulary task in the sense that they can be used to design a large unit inventory, the training data requirements for the unit and lexicon design are impractical for many applications where the training data does not cover all words in the vocabulary. Many of the words in an LVCSR system will have no or very few observations making it impossible to (reliably) design pronunciations for those entries. This is particularly true for a spontaneous speech corpus where a small vocabulary will cover most of the corpus. For example, in the Switchboard corpus used for the experiments described here, the 400 most frequent words cover 87-88% of the training corpus in terms of word tokens, and each is observed more than 100 times.

The system described in this chapter becomes suitable for application to an LVCSR task by relying on a mixture of phonetic units and automatic units. The hybrid nature of the system derives detailed acoustic models for the most frequent

words by designing automatic units and their pronunciations and relies on the phonetic units and a hand-crafted lexicon to provide the desired capability to generalize to unseen or infrequently observed words. The design of the hybrid system consists of four phases. First, the normalization technique used to reduce acoustic differences in the acoustic features due to speaker identity are described in section 5.1. The second is the design of the phonetic-unit sub-system, which will be described in section 5.2. Given this phonetic system, the location of tokens of the most frequently observed words can be estimated. Then, in the third phase, the automatic unit sub-system is designed on the tokens of the most frequently occurring words, whose locations are estimated by the phonetic-unit-based system designed in the second phase. The design of the automatic unit sub-system uses the algorithms described in chapters 3 and 4. In the final design phase, described in section 5.3, the hybrid system is built by combining the two sub-systems trained in isolation. The designed hybrid system is limited to modeling only word-internal contextual effects as in a hybrid system, the cross-word context could either be in terms of phonetic units or automatically derived units making the complicating the design design of a decision tree distribution. The recognition task and experiments using the hybrid system are described in section 5.4. A summary of the main results are provided in section 5.5.

## 5.1 Vocal Tract Length Normalization

A complication in speaker-independent automatic speech recognition compared to speaker-dependent recognition is the increased acoustic variability due to speaker differences. The increased acoustic variability due to a spontaneous speaking style over a read style is in addition to this speaker-dependent variability. Hence, for a speaker-independent spontaneous speech task, application of a technique to reduce the speaker-related acoustic variability is even more desirable than for a read speech task. Much of the acoustic variability due to speaker identity can be contributed to

the differences in the length of the vocal tract. The average length of the vocal tract among males is significantly larger than the average among females, so some speaker normalization can be achieved by the use of gender dependent models. Unfortunately, due to a significant variance in the vocal tract length within a gender group, considerable acoustic variability among speakers within a gender remains. More fine grained automatic normalization schemes were developed over the last couple of years. All these techniques incorporate vocal tract length normalization in the feature extraction phase, but they differ in the way the vocal tract length is estimated. Techniques for the estimation of the vocal tract can be divided in two groups. A “*knowledge-based*” approach was developed by BBN [22], where vocal tract length differences are estimated from the average third formant location. This approach was also investigated by others [75]. A second approach computes the likelihood of a finite set of different feature sequences, each normalized for different vocal tract lengths. The estimated vocal tract length is then obtained by determining which features generated the highest likelihood. In the initial work by Andreou *et al.* [3] as well as later work by Dragon [57] and AT&T [39], the speech recognition system itself was used in the likelihood computation. To address the computational cost problem introduced in this way, Dragon developed an algorithm in which a text-independent multivariate mixture density was used for the likelihood computation [66]. As this approach showed comparable performance (an approximately 2% to 3% drop in word error rates) at much smaller computational cost, many other sites currently use this approach [39, 42, 75].

In the work presented here, both formant and likelihood-based approaches were investigated. For the implementation of the formant-based method, the commercially available *XWaves+* package was used. In the likelihood approach, a 256 mixture segment model with time-invariant parameters was used to estimate likelihoods. The implementation of the vocal tract length normalization of the speech features is described in section 5.1.1. Then, in section 5.1.2, the formant-based vocal tract length estimation procedure is described. Section 5.1.3 describes the parameter estimation

procedure for the mixture models used in the likelihood-based vocal tract length estimation procedure. The algorithm that is used to estimate the vocal tract length of a new speaker is described in section 5.1.4.

### 5.1.1 Frequency warping

Given an estimate of the vocal tract length of a speaker relative to a mean vocal tract length, the spectral features derived from the speech waveform of that speaker are to be normalized, removing acoustic differences due to vocal tract length. To investigate the effect of the vocal tract length on the spectral features consider a simple tube model representing the vocal tract as shown in figure 5.1. The effect of changing



Figure 5.1: A simple tube model of the vocal tract.

the length  $L$  of the tube (i.e. varying the length of the vocal tract) will cause a linear shift of  $k/L$  in of the resonance frequencies with  $k \in \{1, 3, \dots\}$ . Note though that the tube model is only a reasonable model for a schwa vowel (like *a* in *about*). A reasonable model for closed vowels such as */iy/* (like *ea* in *beat*) is the Helmholtz resonator depicted in figure 5.2 which has the first resonance frequencies dependent on the vocal tract length parameter  $L$  as  $\sqrt{(V/AL)}$  while the other resonance frequency dependencies are approximately linear. These simplified models show that in order to compensate for different vocal tract lengths a frequency warping can be used. The models also lead us to the conclusion that this frequency warping should be phone dependent. In this work however, each speaker will be limited to a single phone-independent frequency warping to normalize spectral differences due to vocal tract length differences as is assumed in previous work by others, since phone dependent warpings cannot be implemented in the feature processing. The warping function used is depicted in figure 5.3. The warping parameter  $\alpha$  controls the slope of the linear

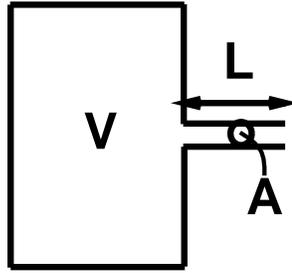


Figure 5.2: The Helmholtz resonator model of the vocal tract.

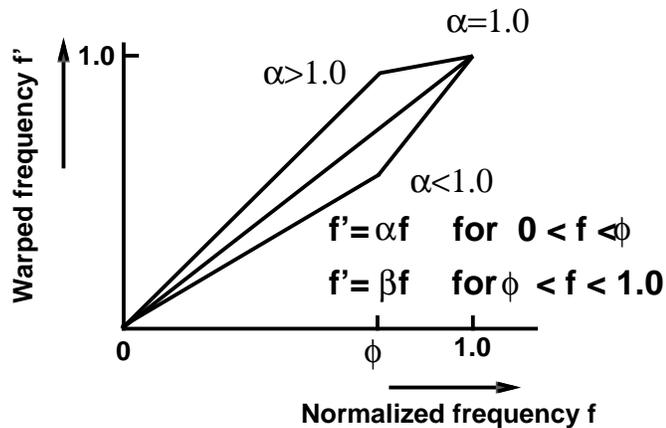


Figure 5.3: Piecewise linear frequency warping function.

warping from 0 to the fixed frequency  $\Phi$ . From that point to the Nyquist frequency, the warping is also linear so as to reach the point (1, 1). To implement this frequency warping the approach described by [66] is used, where the warped frequency axis is sampled at equally spaced intervals. For each warped frequency  $f'$  the corresponding original frequency  $f$  is computed. As the spectral representation is derived by an FFT, there is no guarantee that there is an estimate of the spectral energy at that exact frequency. To derive the spectral energy at arbitrary frequencies in between the discrete frequency estimates provided by the FFT, we use a linear interpolation of the spectrum estimated by the FFT. The warping process is depicted in figure 5.4

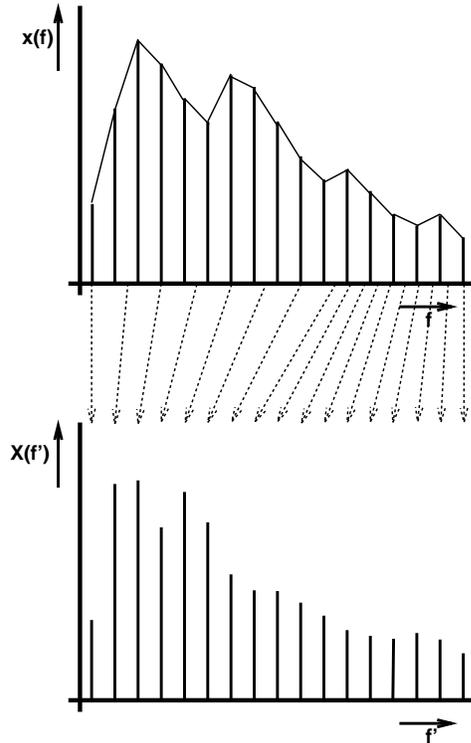


Figure 5.4: Equally spaced sampling of the warped frequency axis using spectrum estimation by linear interpolation with a warp factor  $\alpha < 1.0$ .

### 5.1.2 Formant Based Warp Estimation

To determine the warping for a particular speaker, two related problems have to be solved. First, a reference or “normal” vocal tract length has to be defined. Second, for each new speaker, the warping factor to normalize the speaker has to be determined. One approach is to use formant estimates to solve these problems [22]. The normal vocal tract can be defined by computing the median formant location over all voiced frames from a training corpus. Then by computing the median formant location for a particular speaker by only using the voiced frames from that speaker, an estimate of the warping to normalize the features from that speaker is obtained. If the corpus median formant location is denoted as  $F_c$  and the median formant location of speaker  $S$  is denoted as  $F_S$ , the warping factor for that speaker is simply  $F_S/F_c$ .

The advantage of this approach is that continuous warp estimates are obtained. A disadvantage is that this technique can be computationally expensive when a very robust formant estimation algorithm is used. Another disadvantages of this technique are that the warping factors are sub-optimal in the maximum likelihood sense. As the linear warping is an approximate model, even the “correct” warping estimate may not be the optimal ML match for the recognition model used. In addition, if a computationally inexpensive formant estimation algorithm is used and/or if the technique is applied to short utterances, it will be more likely to get erroneous formant estimates and therefore erroneous warp estimates.

### 5.1.3 Maximum Likelihood Warp Estimation

To solve the related problems of defining the “normal vocal tract” and to have an algorithm to automatically determine how to frequency-warp the data from a speaker towards the normal vocal tract, a likelihood-based approach can be used. Here an approach similar to [66] was used except that a segment-based mixture model was used rather than a frame-based one. The segment models used a single time-independent Gaussian distribution to model the observations within a segment. A text-independent segment-based multivariate mixture model is trained for speech from “the normal vocal tract”. To train this model and simultaneously define the normal vocal tract, the following training algorithm was used:

1. **Initialize:** Estimate a multivariate mixture model  $\Lambda_0$  on un-warped features. Set  $i = 0$ ;
2. **Likelihood computation:** Compute the likelihood of the features of each training speaker  $m = \{1, 2, \dots, M\}$  warped at different warp factors  $\alpha \in A = \{\alpha(1), \alpha(2), \dots, \alpha(N)\}$  given the last model  $\Lambda_i$ .
3. **Estimate warps:** Find the most likely warp factor  $\alpha^m \in A$  for each speaker  $m$  by determining which warped features were found most likely given  $\Lambda_i$ . Let

$\alpha_i^m$  denote the most likely warping for speaker  $m$  at iteration  $i$ .

4. **Training set definition:** Define a new training set using for each speaker the features warped using  $\alpha_i^m$ .
5. **Retrain:** Retrain the mixture model using the newly defined training set.
6. **Iterate:** Set  $i = i + 1$ , go to 2.

A pictorial representation of this training algorithm is given in figure 5.5.

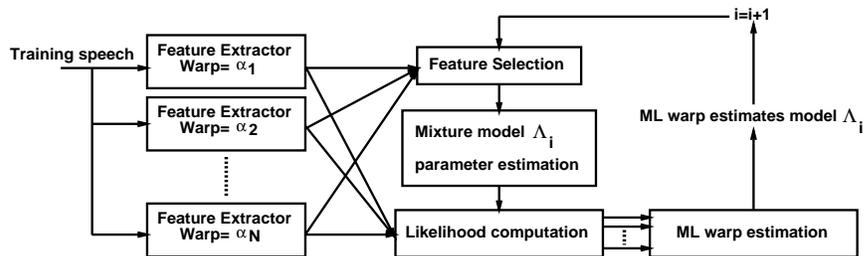


Figure 5.5: Warp mixture model training overview.

The mixture model was trained using a divisive clustering approach in which each cluster is represented by a segment model with a constant mean and covariance. The distance measure used in clustering is the negative log-likelihood of data with respect to a cluster model.

As the mixture model is implemented using segment models, a segmentation has to be derived before this iterative training scheme can be executed. The acoustic segmentation algorithm described in 3.2.1 was used to derive this segmentation. It is important to prevent introducing a bias for certain warpings by allowing the segmentation for one warp factor to have a larger number of segments (i.e. a larger amount of freedom) than the segmentation of the features at another warp factor. The segmentation is therefore derived in two steps. First the un-warped features are acoustically segmented. Then the features at other warpings are segmented under the constraint that the number of segments per utterance is equal for each warping. Graphically, the segmentation is derived as shown in figure 5.6.

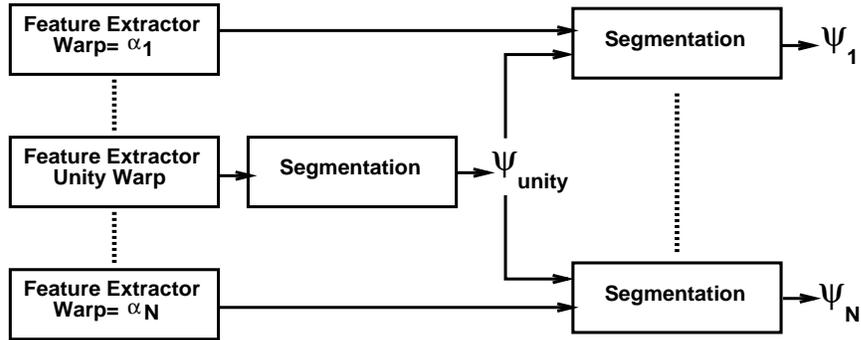


Figure 5.6: Acoustic segmentation of the data for the purpose of training the likelihood-based warp estimation model.

### 5.1.4 Normalizing Test Speakers

To normalize the features of a test speaker (i.e. a speaker not included in the training set), some or all of the speech available from that new speaker is used to estimate the appropriate warping factor for that speaker. Typically 30 to 60 seconds of speech is used to estimate the warp factor of a speaker. After estimating the speaker-dependent warp factor, all the features derived from the speech of that speaker are warped using this warping factor.

For the formant-based approach to warp factor estimation, the median formant location of the test speaker is estimated using the formant tracker also used in the training process. Given the estimated formant location of the test speaker, the frequency warping factor for that speaker is determined as  $F_s/F_c$ , where  $F_s$  is the speaker median formant location and  $F_c$  is the median formant location estimated in the training process.

In the likelihood-based approach, features are computed for the amount of speech used in warp factor estimation at all allowable warps. The un-warped features are then acoustically segmented first. Subsequently, the features warped at the other factors are acoustically segmented under the constraint that the resulting segmentation should have the same number of segments as the un-warped segmentation does. The

likelihood of these features given the trained likelihood model is then computed for all features at all warps. The speaker-dependent warp factor is then determined according to which warped features generated the highest likelihood. This warp factor estimation process is depicted in figure 5.7.

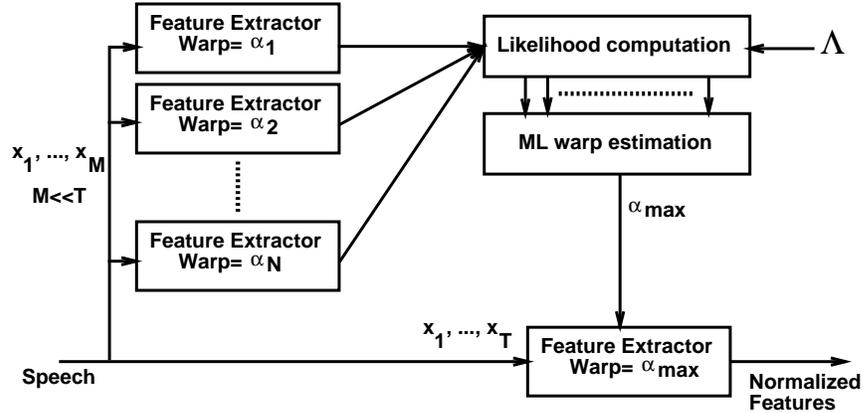


Figure 5.7: Test speaker warp factor estimation using a likelihood approach.

## 5.2 Phone-Based Sub-System Design

The first phase of the hybrid system consists of the design of a phone-unit-based sub-system. In the work described here, two types of phonetic sub-systems were used; both modeled context explicitly but one considered only word-internal contexts, the other considered contextual effects across word boundaries. In both systems, a hand-crafted phonetic unit inventory (54 phones) and lexicon was used for the phonetic part of the system. Both systems had equal complexity in the sense that an equal number of distributions were used in both systems. Both also used the same *bigram* language model, (i.e. using a first order Markov assumption or restricting the memory of the model to only the last word).

The word-internal system used phonetic unit left-to-right 3-state HMMs with a topology allowing the center state to be skipped. The cross-word system also used

left-to-right 3-state HMMs but did not allow state skipping. The transition probabilities of both HMMs were kept uniform, i.e. the transition probabilities were not re-estimated in training. As the acoustic likelihood is mainly determined by the emission probabilities, ignoring the transition probabilities will only have a small effect on the results. The acoustic models of both systems used phone units in an explicit left and right context (triphones). The state emission probabilities of these triphone models were modeled by mixture Gaussian distributions, with shared distributions derived by a clustering step. Parameters for these models were estimated from data by gradually increasing the system complexity, starting from single Gaussian distribution models. First the word-internal system was trained starting from context-independent phonetic units as described below. Given the phone-state boundary estimates provided by that system, the cross-word unit system was designed.

Initial context-independent phone model parameters were estimated from a phone-level segmentation provided by another speech recognition system [47]. The parameters of these models were then refined using Viterbi training with the constraint of fixed word boundaries. The model inventory was then increased by explicit modeling of context at the state level (i.e. each state was modeled with an explicit context of the left and right state rather than the left and right phone). These explicit state context units are referred to as “tri-state” units. Single Gaussian emission probabilities were estimated for all unique tri-state units by EM training. The system complexity was then further increased by explicitly modeling context at the phone level. The tri-state system was used to re-align the data, and sufficient statistics were computed for states in all unique left and right phone contexts (triphone sufficient statistics were computed) but ignoring contexts across word boundaries (word-internal system). These sufficient statistics were then clustered using combined divisive and K-means clustering with likelihood as the objective function, similar to the ASWU system. No structure was imposed on clustering, so units are allowed to share across center phone identity and state position within the phone models. The cluster in-

ventory was initialized with the tri-state model inventory derived previously by the EM re-estimation. The shared triphone state distributions derived by the clustering process were then refined by 3 iterations of EM training. The complexity of the clustered triphone models was then increased by estimating mixture distributions, using the incremental mixture-splitting technique described in [73]. In this approach, the number of mixture components of the emission distributions were increased in stages and the parameters of the intermediate mixture distributions were re-estimated by EM training. Given a re-estimated intermediate mixture distribution, the number of mixture components are increased by perturbing the mean(s) of the mixture component(s) with the largest variance(s) to provide the initialization for subsequent EM re-estimation steps. This gradual increase of parameters with re-estimation at intermediate stages attempts to prevent the system from converging to a local optimum. As found in [73], more accurate mixture distributions usually result when avoiding dramatic changes in the number of free parameters during the system design. This approach relies on the conjecture that a well estimated initialization point for a system with an increased number of mixtures can be derived from the careful estimate of a system with a fewer number of mixtures.

Provided with the word-internal system, a new alignment of the training data was found by the Viterbi algorithm. Using those phone-state alignments, sufficient statistics were computed for each unique triphone state including contextual effects across word-boundaries. These sufficient statistics were then clustered using decision tree clustering. As in the word-internal system, the shared distribution were re-estimated by EM training, and mixture distributions were derived for the emission probabilities.

### 5.3 Hybrid System Design

The hybrid system integrates the two types of units. One option for system building is to simply merge the sub-systems designed independently. In such a merged system, the unit inventory is defined as the union of the automatic unit and phonetic unit inventories, and the lexicon uses automatic units for the most frequent words and phonetic units for the remaining entries. Although this approach to the hybrid system design is simple, it has several problems. First, the automatic units might provide easier separable models in comparison to the phonetic unit word models for some of the most frequent words but possibly not all of them. To solve this problem, a criterion is needed to decide whether to include the phonetic or automatic unit pronunciation in the lexicon. Second, as the most frequent words are now modeled by the automatic units, the phone-based units do not need to cover the full space of triphones and the parameters can be re-estimated. However, eliminating a large portion of the training data could make the models less general and therefore less accurate on unseen data. Third, as the automatic units were trained using isolated tokens with fixed word boundaries, it is likely that embedded training (allowing the word boundaries to shift) will result in more accurate model parameter estimates.

As an alternative to simply merging the independently-trained automatic and phonetic unit systems, a parameter re-estimation step is performed on a parallel version of the hybrid system. In other words, the most frequent words are represented with multiple pronunciations: the automatic unit pronunciation and the phonetic unit pronunciation. The estimation step of the EM algorithm is then used to compute two probabilities: the standard “state occupancy” probability as described in equation 2.21 (the probability of being in a particular mixture component of a state at a particular time given the whole of the observation sequence), and the “word-initial state transition probability” (the probability of transitioning into either the first automatic unit or phonetic unit states of a multiple pronunciation word at any time given the whole observation sequence). This “word-initial state transition prob-

ability” can be computed from the state transition probabilities  $\xi_t(i, j)$  described in equation 2.22 and can be used to estimate the probability of each possible pronunciation. The estimated probability of pronunciation variant  $v$  of word  $w$  (either the automatic unit or phonetic unit pronunciation of a word) given the HMMs and the training data can be computed as

$$P(v | w, \mathbf{Y}, \lambda) = \frac{\sum_{t=1}^T \sum_{i \in F(s_v^w)} \xi_t(i, s_v^w)}{\sum_{t=1}^T \sum_{i \in F(s_v^w)} \sum_{k=1}^M \gamma_t(i, k)}, \quad (5.1)$$

where  $T$  denotes the total number of observations in the training data,  $s_v^w$  denotes the initial state of pronunciation variant  $v$  of word  $w$  and the function  $F(\cdot)$  describes the set of all states that can proceed the state given as its argument except for the state itself (i.e. the final states of proceeding words from which a transition to the initial state can be made). This estimated probability of each possible pronunciation can be used to weight or to prune the different pronunciation alternatives for each word. The state occupancy probabilities can be used to re-estimate the model parameters of either or both types of unit models. In the work described here, only the ASWU models were re-estimated to avoid possible problems associated with triphone model re-estimation from a biased data sample for unseen models. In addition, the implementation uses pruning rather than weighting, in which case pronunciations are removed from the lexicon. As the ASWUs need to cover fewer words than during the independent ASWU sub-system design, it is useful to do a second parameter re-estimation pass to obtain emission distribution estimates, accurately describing the data corresponding to the words that they cover.

## 5.4 Experiments

First in section 5.4.1, the results of a pilot experiment using different approaches to normalizing the features for speaker differences are described. Then in section 5.4.2, the Switchboard corpus and the feature extraction process for the LVCSR experiments is described. The performance of the phonetic baseline system is described in

section 5.4.3. The hybrid system experiments conducted on this corpus are described in section 5.4.4.

### 5.4.1 Feature Normalization Experiments

To investigate the formant and likelihood-based warp factor estimation techniques, a pilot experiment was conducted on the TIMIT corpus. This corpus consists of a training set of 462 speakers and a test set of 168 speakers with approximately 25 seconds of read English speech per speaker. The speech is read in a recording studio, digitized at a 16kHz sampling rate and quantized using 16 bits per sample. Features were computed using a 25 ms Hamming window at a rate of 100 frames per second. Mel-scale cepstral coefficients of dimensionality 14 were computed from a 24 channel filter-bank using triangular filters. The usefulness of the features for the purpose of speech recognition were evaluated by a phone segment classification experiment. The TIMIT corpus is particularly appropriate for this type of experiment as the speech was manually segmented at the phone level. A segment model was estimated for each of the 48 phone labels used in the experiment. The segment models represented the feature trajectories of the phones by a 3 region piecewise stationarity and represented each stationarity by a single mixture Gaussian distribution with a full covariance. The model parameters were estimated on the normalized features of all of the training data. Testing was done on the complete test set consisting of 50754 phone segments.

First, using the formant-based approach to warp factor estimation, the median locations of the formants were estimated for the complete training corpus. Using the formant tracking algorithm implemented in the *XWaves+* software package, the median locations of the first, second and third formant were estimated. The median formant locations are given in table 5.1.

Then using the formant estimates from the data of each speaker, warp factors were computed using either the first, second or third formant locations. A histogram of the third formant-based warp estimates for the speakers in the training and test

Formant	Median (Hz)
F1	557
F2	1534
F3	2597

Table 5.1: Median formant locations of the first, second and third formants in the training part of the TIMIT corpus.

sets of the corpus are depicted in figure 5.8. Separate histograms are depicted for the male and female speakers which shows a clear separation of the warp factors based on gender but also shows a significant variance in warp estimates among speakers from the same gender.

The classification performance by use of speaker normalized features using the formant-based approach are summarized in table 5.2. The baseline classification rate was obtained by using features without normalization. The classification rates for normalized features used normalization in both the training and testing phases.

Formant	Classification rate	Improvement
baseline	43.6%	-
F1	45.3%	1.7%
F2	45.2%	1.6%
F3	45.6%	2.0%

Table 5.2: Classification improvements due to speaker normalization using a formant-based approach.

In the likelihood-based TIMIT experiments, a gender balanced training set of 272 speakers was constructed, randomly selected from the available 462. A 256 mixture segment model was estimated for warp factor estimation. The set of allowable warp factors was  $\alpha \in \{0.80, 0.82, \dots, 1.20\}$ , based on warp estimates obtained from the third formant experiment. To obtain the final mixture model, 5 iterations of the

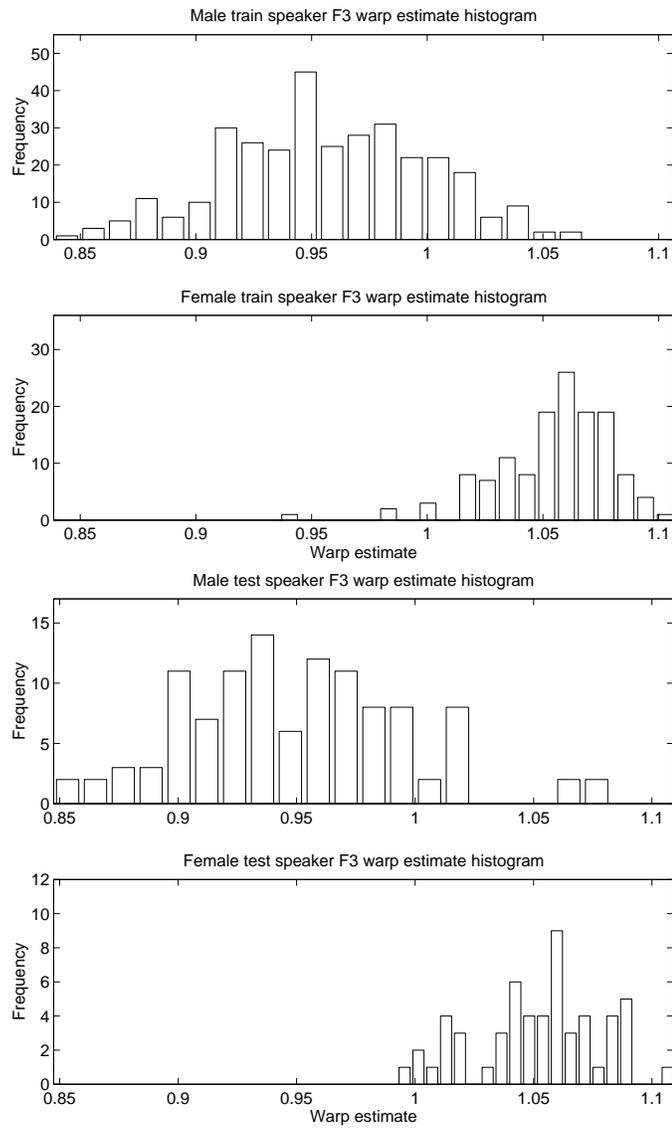


Figure 5.8: F3-based warp factor histograms for training and test sets.

algorithm described in 5.1.3 were run. A histogram of the warp factor estimates for the speakers in the training and test sets are depicted in figure 5.9. As in the formant-based approach, the warps for male and female speakers are clearly separated and a significant variance exists among speakers from the same gender. The training data likelihood throughout the iterative training process and the classification rate obtained by estimating warp factors for the test speakers using the mixture models obtained after each of the iterations is depicted in figure 5.10. Note that the likelihood increase of the training data going from the fourth to the fifth iteration is very small and that the likelihood of the test data decreases at this iteration. The classification result shows a similar trend.

As the results for the likelihood-based warp factor estimation procedure are slightly better than those using a formant-based approach (45.6% vs. 45.8%), the results seem to indicate that the advantage of estimating the best warp in the likelihood sense outweighs the disadvantage of having warp estimates limited to a discrete set of allowable warpings. Therefore, the ML approach was used in experiments on the Switchboard corpus.

The performance of speaker normalization could be improved by making the warp factor phone-dependent rather than speaker-dependent. It is questionable however if the current likelihood-based approach is suitable as is in such a framework, since the current approach requires much more data than the average duration of a phone to make a reliable estimate of the warping factor as illustrated in figure 5.11 for a few speaker. As shown, most speakers required at least 10 seconds of data before a “converged” warp factor estimate was found.

## 5.4.2 The Switchboard Corpus

For the experiments on a large vocabulary spontaneous human-to-human dialogue task, the Switchboard corpus was used. This corpus consists of approximately 160 hours of telephone quality speech (digitized at 8kHz, quantized using 8bit mu-law

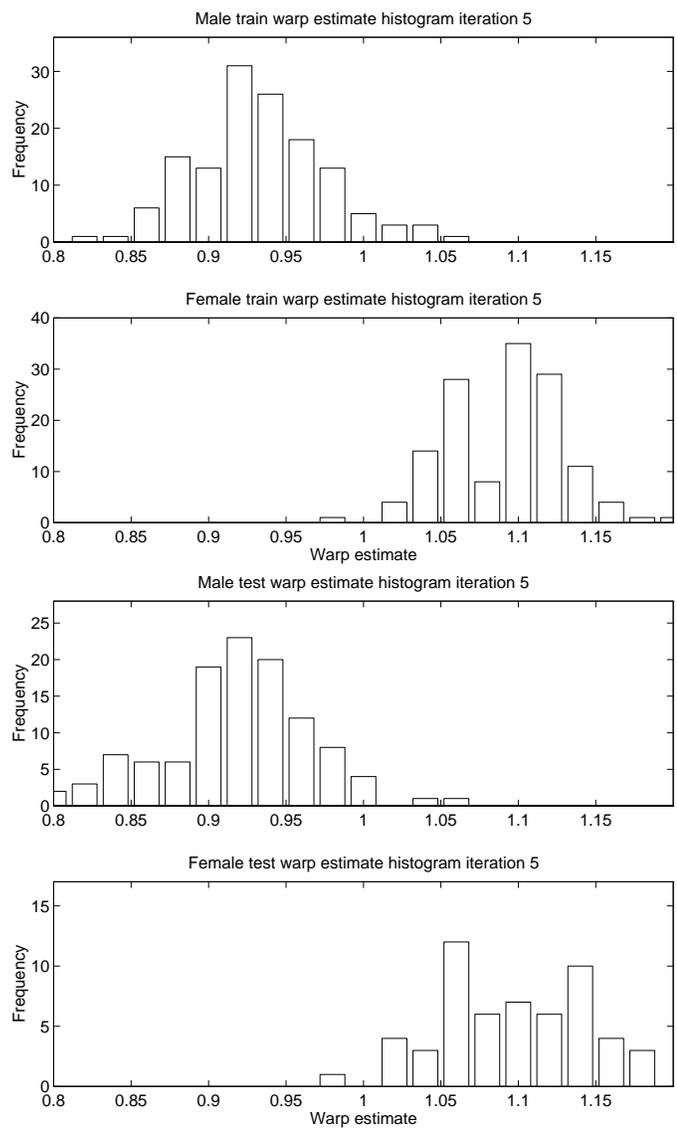


Figure 5.9: Warp factor histogram of training and test speakers using the mixture model estimated after 5 iterations.

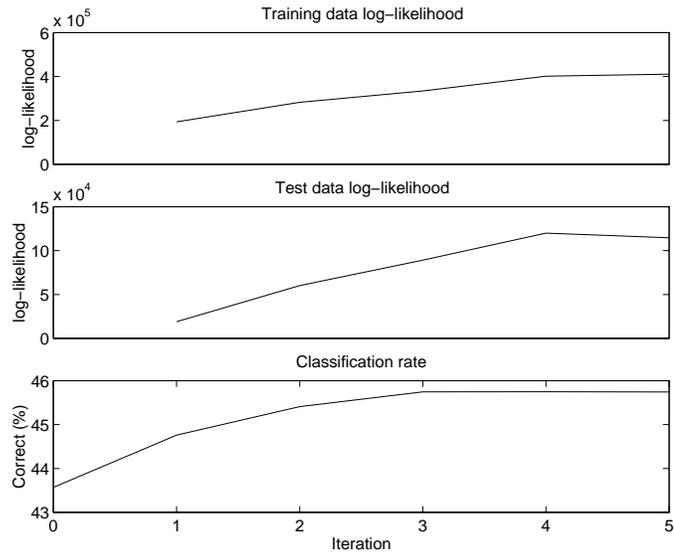


Figure 5.10: Classification scores and data likelihoods using the different mixture models estimated after each training iteration step.

coded samples). The data consists of spontaneous conversational speech recorded over long-distance telephone lines. The speakers generally do not know each other and are asked to converse on a pre-determined topic (chosen from a set of topics). The conversations are approximately 5 minutes in length.

All recognition experiments used approximately 120 hours of speech from about 2500 conversations of the Switchboard corpus as training material. Some experiments used the mel-scale cepstral coefficients with vocal-tract-length normalization applied, others used the mel-scale cepstral coefficients without normalization. A test set of approximately 30 minutes of speech from 7 conversations was defined. Gender detection was performed using the likelihoods of the vocal-tract-length normalization model. The test lexicon contained 20500 entries for 19557 unique words (i.e. 939 entries had multiple pronunciations). The parameters of the bigram language model used in decoding were estimated on the approximately 3 million words training text available for the corpus. The out-of-vocabulary rate (percentage of word tokens not corresponding to any word in the lexicon) for the test set using this lexicon was 0.7%.

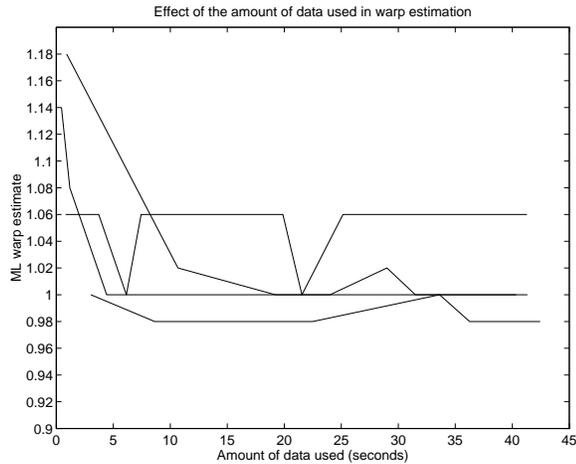


Figure 5.11: ML estimate of the warping factor for 4 different speakers as a function of the amount of data used in estimation.

Speaker normalization for this corpus was performed using the likelihood-based approach and a mixture model for the purpose of warp factor estimation was computed on a gender balanced training set taken from 240 conversation sides, randomly selected from the available 2559. A total of 4 hours of speech was used for training, with approximately 60 seconds per speaker. Approximately 40 seconds of speech was used for the warp estimation of test speakers.

From the waveforms, features were extracted at a rate of 100 frames per second using a 25 ms Hamming window. Mel-scale cepstral coefficient vectors of dimensionality 14 were derived from a 24 channel filter-bank of triangular filters equally spaced along the mel-scale. As the hybrid system used for this task was gender-dependent (i.e. a separate system was designed for the male and female speakers) a gender dependent speaker normalization scheme was applied. For each gender, a 256 mixture segment model was used for warp factor estimation. The allowable warping parameters used in these experiments were  $\alpha \in \{0.80, 0.82, \dots, 1.36\}$ .

The warp factor distributions of the speakers in the training set (data from these speakers were used to design the mixture model) after 5 iterations of training of the male model are depicted in figure 5.12. The warp factor distributions of the speakers

in the training set after 5 iterations of training of the female model are depicted in figure 5.13. The distribution of warp factors for all the 2559 speakers in the corpus using the gender dependent models are depicted in figures 5.14 and 5.15 for males and females respectively. The data likelihood given the models at different iteration steps is given in figure 5.16.

### 5.4.3 Phone-based System Results

The word-internal phone-based system was trained using the method described in Section 5.2, resulting in clustered triphone inventory sizes of 5265 and 5244 for the male and female system, respectively. The features used for this system were 14-dimensional VTL normalized features and derivatives. The triphone state emission probability distributions were refined further to 12-mixture distributions by mixture splitting and EM training. The average likelihood per frame during the process is depicted in figures 5.17 and 5.18 for the male and female parts of the system respectively. The mixture increase pattern and the number of EM iterations used to train the mixture distributions were chosen in correspondence with the methods reported for the training of other large vocabulary systems [72]. Note however, that the likelihood after the EM training iterations has not reached an optimum yet. The performance of the 12 mixture word-internal triphone system was 40.0% accuracy (42.5% accuracy for the male sub-system, 36.7% for the female sub-system).

The 12 mixture model inventory was then used to re-segment the training data, allowing both phone state as well as word boundaries to move as much as  $\pm 400$  frames. In addition, at each word boundary, an optional silence word was allowed to be inserted. The obtained word level segmentation contained explicit information as to which pronunciation variant was used for those words having multiple pronunciations.

Starting from the phone-state alignment provided by the word-internal phone-based system, a cross-word phone-based system was designed. This system was de-

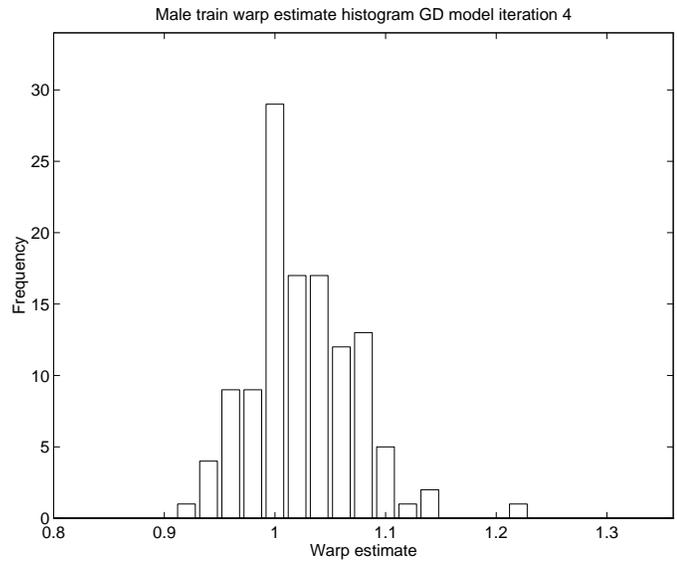


Figure 5.12: Warp factor histogram of training data using the male gender dependent mixture model estimated after iteration 4.

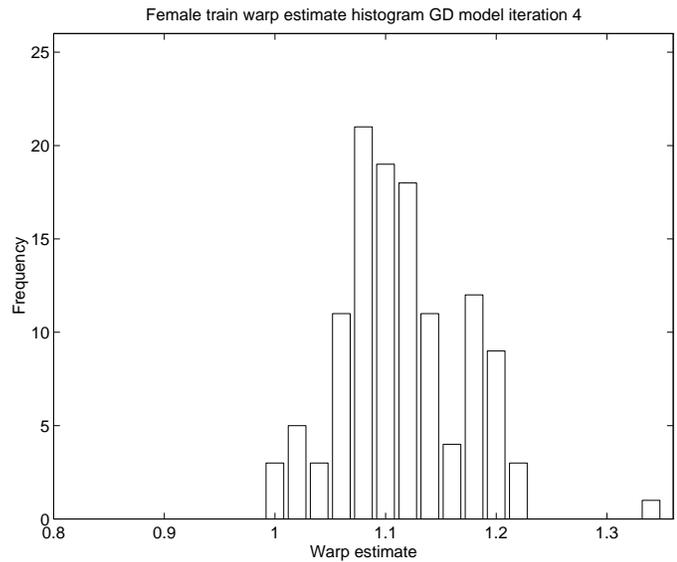


Figure 5.13: Warp factor histogram of training data using the female gender dependent mixture model estimated after iteration 4.

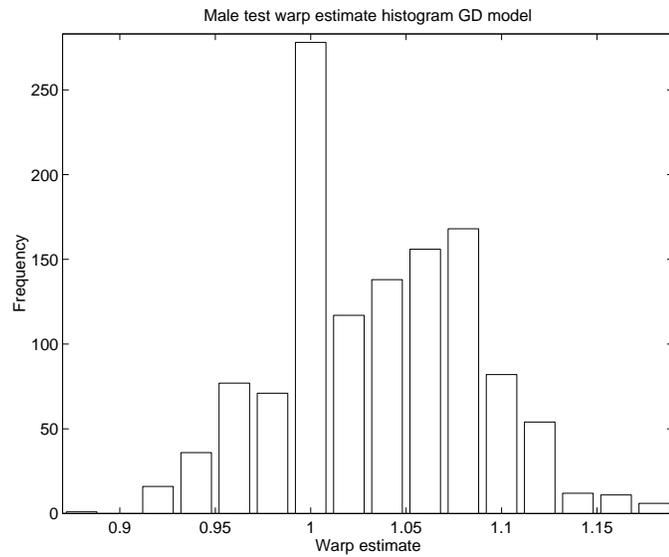


Figure 5.14: Warp factor histogram of all data using the male gender dependent mixture model estimated after iteration 4.

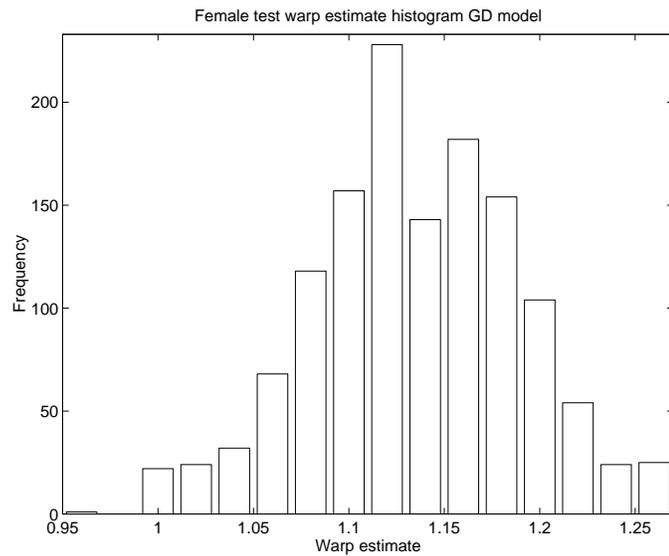


Figure 5.15: Warp factor histogram of all data using the female gender dependent mixture model estimated after iteration 4.

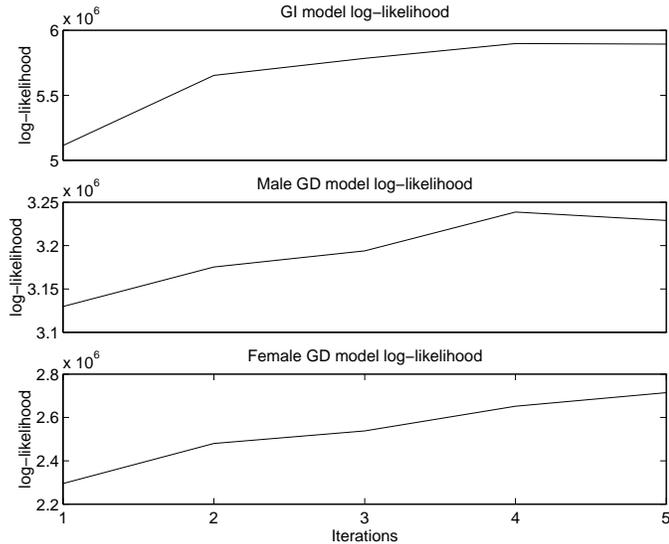


Figure 5.16: Data likelihoods of the training data given the model estimated after the different training iterations.

signed on features that were not normalized for the vocal tract length of the different speakers (i.e. features were mel-scale cepstral coefficients). The features used in the system were 12 dimensional cepstral coefficients, normalized energy and their first and second derivatives (39 dimensional vectors). Sufficient statistics were computed for every unique state observed in the alignment when considering an explicit left and right phone context (triphones). These sufficient statistics were then clustered using decision tree clustering, designing a tree for each unique phone-state. In other words, sharing only among different contexts of the same center state were allowed. Using a greedy tree growing algorithm, 5265 and 5244 leaf distributions were estimated for the male and female system respectively. These distributions were subsequently refined to 12 mixture distributions using the mixture splitting and EM re-estimation algorithms. The average likelihood per frame of training data for the male and female part of the system during this process are depicted in figures 5.19 and 5.20 respectively. As the accuracy of the word-internal system was below state-of-the-art performance, a slower increase in the number of mixtures and a larger number of EM

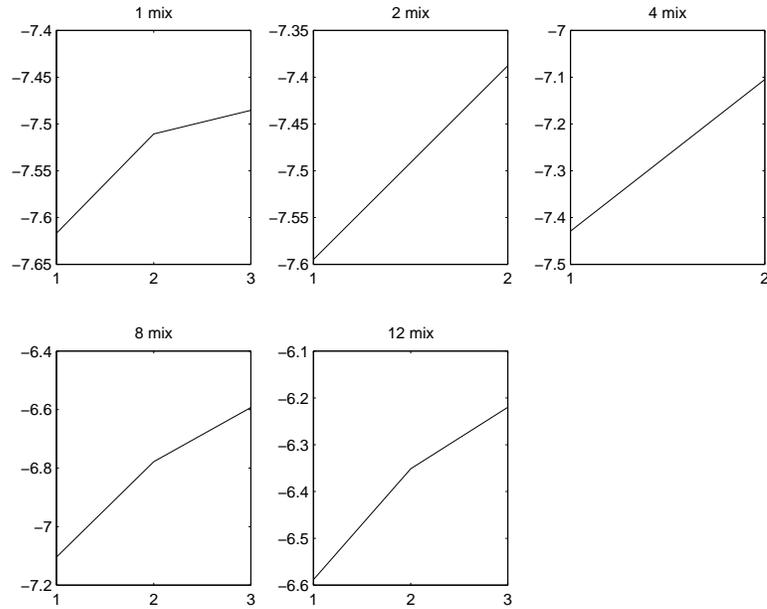


Figure 5.17: Average likelihood per frame of training data for the male word-internal phonetic-unit system during the mixture splitting and EM re-estimation process.

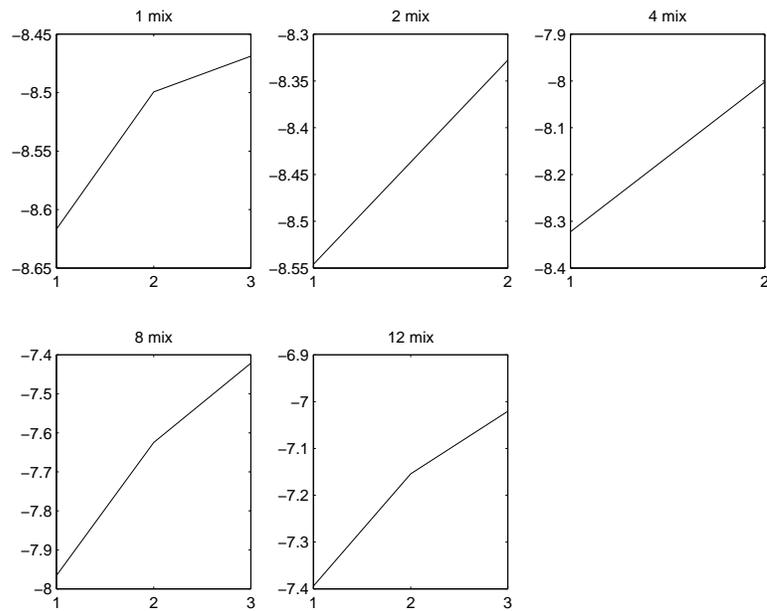


Figure 5.18: Average likelihood per frame of training data for the female word-internal phonetic-unit system during the mixture splitting and EM re-estimation process.

iterations were chosen to allow a higher training likelihood and improved recognition performance.

The recognition performance of the cross-word system using the un-normalized features, resulted in an overall test accuracy of 50.9% (50.0% on the male sub-set, 52.1% on the female sub-set). These results are comparable to state-of-the-art triphone-based, cross-word systems [51]. The improved performance of this system over the word-internal system (50.9% accuracy vs. 40.0% accuracy) is partly due to the incorporation of cross-word contextual effects but also due to the slower mixture increase and larger number of EM iterations performed during the training of the system as the gain of modeling cross-word effects in comparison to modeling with-words effects alone are generally on the order of 2% to 4% [51].

#### 5.4.4 Hybrid System Experiments

The word-internal **ASWU sub-system** was trained using the algorithm described in chapter 3 (no temporal refinement, using a corpus based covariance estimation and average likelihood per frame thresholding mechanism for the acoustic segmentation step), with an initial acoustic segmentation tuned so that there were on average 3.4 acoustic segments per phone segment. A total of 3000 automatic unit models/gender were estimated through constrained clustering. The number was chosen arbitrarily to be a large fraction of the phone-based system number, since the models covered a small percentage of the lexicon entries but a large percentage of the data that the phone-based models covered. The parameters were estimated from the word tokens of the 400 most frequent words, keeping word boundaries fixed to the times provided by the word-internal phone-based system. Again, 12 mixture distributions were estimated for the automatic unit models by mixture splitting and EM training. The average likelihood per frame of the training data during the mixture splitting and EM re-estimation process are depicted in figures 5.21 and 5.22 for the male and female parts of the system respectively. As in the phone-based sub-system, the data

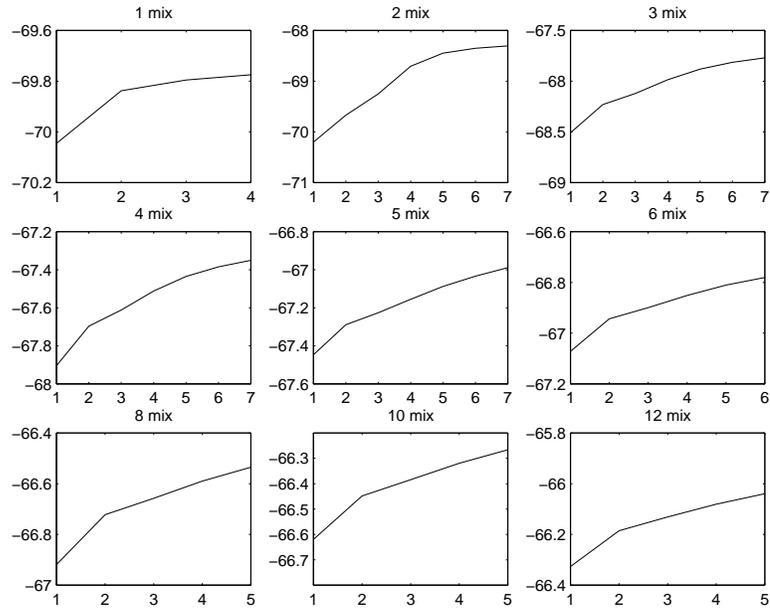


Figure 5.19: Average likelihood per frame of training data for the male phonetic-unit system during the mixture splitting and EM re-estimation process.

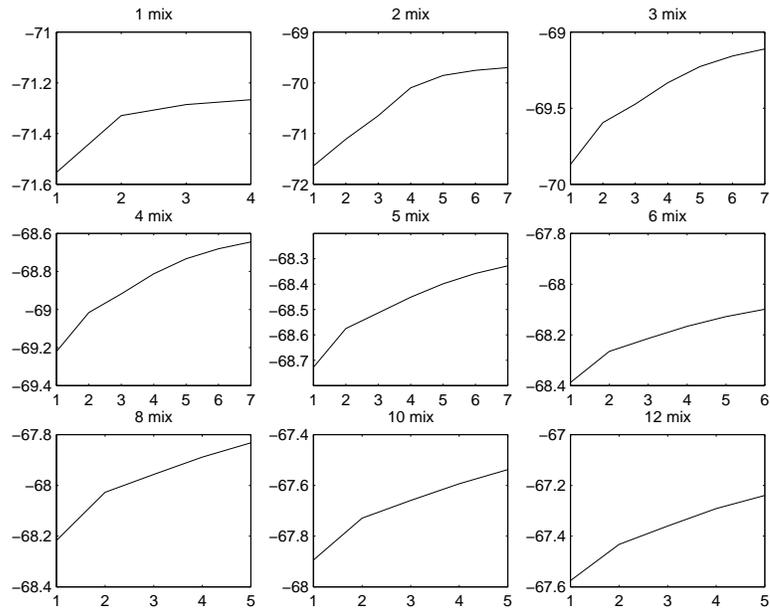


Figure 5.20: Average likelihood per frame of training data for the female phonetic-unit system during the mixture splitting and EM re-estimation process.

likelihood does not converge to an optimum within the number of EM iterations run.

For the design of a word-internal **hybrid system**, one EM re-estimation pass was performed. A histogram of the pronunciation probabilities for the automatic unit pronunciations of the most frequent lexicon entries is given in figure 5.23, showing that the automatic unit pronunciation is more likely (i.e. a better fit to the data) for most but not all words (for cases where there are multiple phone-based pronunciations per word, there are the same number of automatic unit pronunciations and these probabilities are summed in the figure). A pruned lexicon was then obtained by using the automatic unit pronunciations when they had a combined probability larger than 0.5, and the phone-based pronunciations otherwise. The automatic unit model parameters were then re-estimated by another EM re-estimation pass, but the phonetic model parameters were held fixed.

Recognition performance on the test set using the phonetic units alone and the hybrid system are given in table 5.3. The application of the automatically derived units

System	Male (% acc.)	Female (% acc.)	Overall (% acc.)
Phonetic	42.5	36.7	40.0
Hybrid (no re-est.)	43.0	38.7	40.5
Hybrid (re-est.)	43.9	39.2	41.9

Table 5.3: Recognition accuracy of the word-internal system on the Switchboard test set using either phonetic units alone or a hybrid system. The features are vocal tract length normalized mel-scale cepstral coefficients.

without parameter re-estimation leads to an 0.5% absolute accuracy improvement which increases to 1.9% accuracy after one EM parameter re-estimation step.

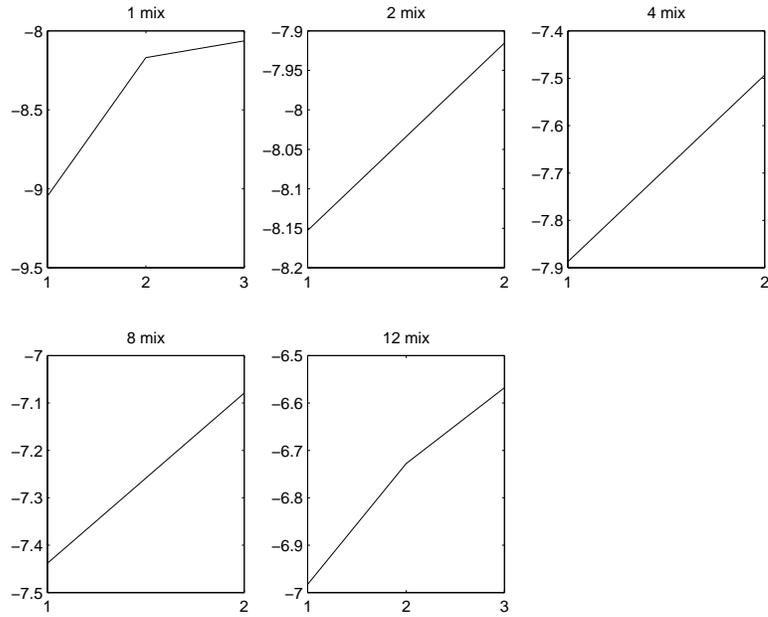


Figure 5.21: Average likelihood per frame of training data for the male word-internal ASWU system during the mixture splitting and EM re-estimation process.

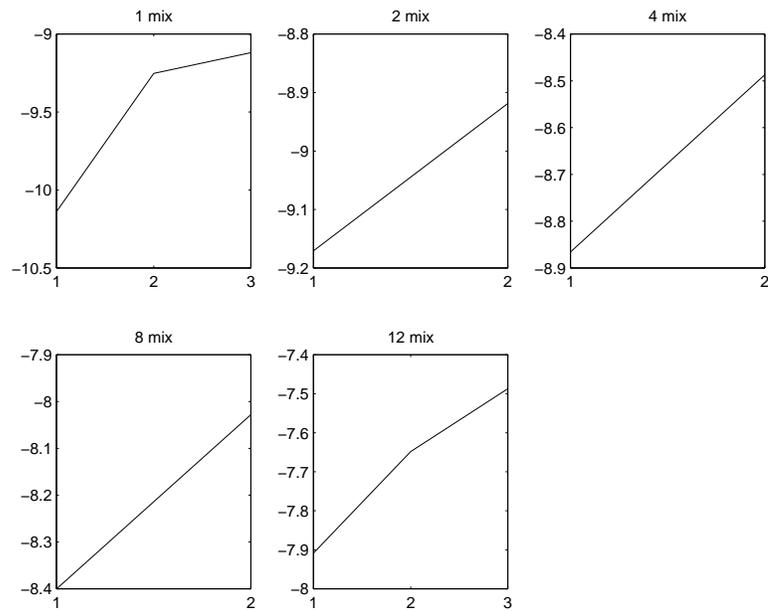


Figure 5.22: Average likelihood per frame of training data for the female word-internal ASWU system during the mixture splitting and EM re-estimation process.

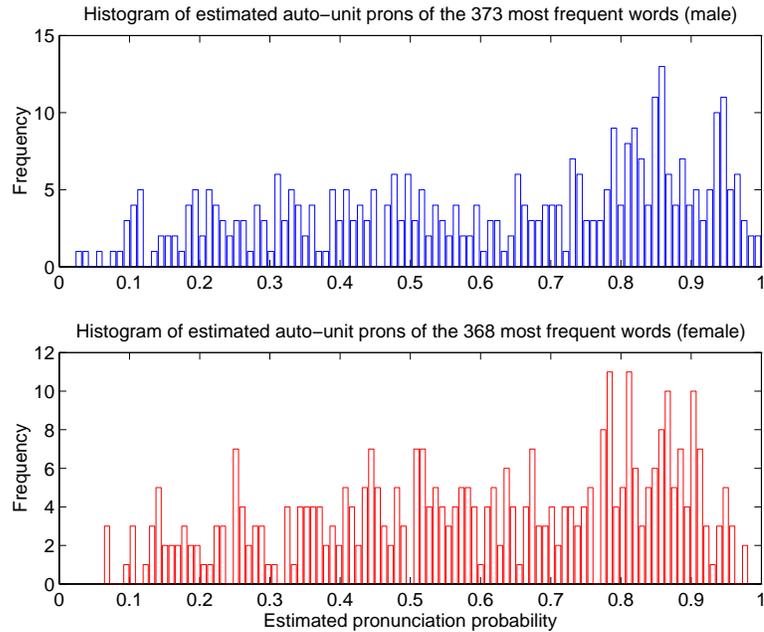


Figure 5.23: Histogram of estimated probabilities of the automatic unit pronunciations.

## 5.5 Summary and Conclusions

Some normalization for the features used in a speaker-independent speech recognition system can be achieved by applying frequency warping in the feature computation step. The warping appropriate for a particular speaker can be estimated using a formant estimation procedure or by evaluating the likelihood of the differently warped features. In a pilot study on the TIMIT corpus, evaluating the effectiveness of the normalization procedure by means of a phone classification experiment showed classification performance improvements using either method. The warp factors estimated by either method showed a clear separation of the distributions of warp factors of males and females, but also a considerable spread around the gender dependent mean warp factor.

Using automatic units for large vocabulary speech recognition becomes feasible when incorporated in a hybrid system that combines the advantages of automatically-

derived acoustic units for high frequency words with the advantages of generalizable phone-based units for infrequently observed words. Experimental results on the Switchboard corpus show that the automatically-derived units and associated pronunciations do indeed give a better fit to the data than phone-based units, in terms of higher training likelihood and improved recognition accuracy. The experimental results are on a first-pass decoding paradigm (within-word triphones and bigram language model), and further system development and experiments are needed to demonstrate improvement in a more complex, multi-pass decoding system. In our experiments, the phone-based models are based on the full data set, and are not re-trained to reflect the fact that they are not used for the most frequent words. One unresolved question is whether there is a performance gain to be had from retraining or adapting the phone-based models.

# Chapter 6

## Conclusions

In the currently popular approach to speech recognition, statistics are used to pose the recognition problem as one of finding the most likely sequence of words given the acoustic observations. The likelihood of a hypothesized word sequence is computed from the contributions of two model components, the language model describing the likelihood of observing a particular sequence of words and the acoustic model describing the likelihood of acoustic observations corresponding to a word-sequence. The focus of this thesis is on the acoustic model. Large vocabulary speech recognition systems typically use a unit inventory as the basic building block for the acoustic model. The acoustic model is therefore defined by a unit model inventory describing the acoustics and a lexicon describing how word-models can be constructed from the unit models. Most systems use phonetically motivated units as the basic building blocks. The advantage of using such a unit is that it allows the use of phonetic knowledge to be used in construction of the acoustic model. This knowledge allows the model to generalize, as it can be used to synthesize models for unseen events both at the word-level (providing a pronunciation for words infrequently observed in the training data) and at the unit level (providing a distribution for units in infrequently observed contexts). A problem of using phonetic units as the basic building blocks is that they are suboptimal in terms of the maximum likelihood objective function used

in the system. Where the parameters of the language and acoustic model are estimated from data using a maximum likelihood objective function, the unit inventory and lexicon are generally hand-crafted. This thesis provides techniques that allow the use of automatically derived units (and lexicons) in state-of-the-art systems. The provided algorithms are particularly well suited for the design of units meant for large vocabulary speech recognition as they address the weaknesses of automatic units in comparison to phonetically based units. The extensions proposed in this thesis create a framework that provides solutions to the generalization problems – the problem of providing a way to synthesize models for unseen events at the word and local context level – and retain the desirable qualities of automatic units. In this chapter, the key algorithmic and experimental contributions are summarized in section 6.1, and possible extensions of this work are discussed in section 6.2.

## 6.1 Summary and Contributions

Although the use of automatically derived units was proposed before, the developed techniques had three key limitations that prevented application in a large vocabulary state-of-the-art speech recognition system. First, large vocabulary recognition requires the use of a large unit inventory but previous automatic unit and lexicon design algorithms become computationally too expensive to be practical for the case where there is a large inventory. The algorithm summarized in section 6.1.1 provides a computationally efficient solution to the design of a large unit inventory jointly with the lexicon. The best word-internal ASWU system achieved a 14% relative error reduction in comparison with the best performing state-of-the-art triphone based system at a comparable number of free parameters. The proposed algorithm is particularly well suited for large training corpora as storage and computational requirements are (for the most part) proportional to the vocabulary size of the system rather than the size of the training corpus itself. Second, in order to allow explicit modeling of con-

textual effects across word boundaries, a mechanism to synthesize models for units in unseen contexts needs to be provided. In contrast to automatic units, phonetic units can provide such generalization by use of the grouping of phones provided by phonetic knowledge. The algorithm summarized in section 6.1.2 can be used to derive unit classes that in turn can be used to build a distribution predictor capable of synthesizing distributions for unseen contexts using the algorithms developed for context-dependent phonetic units. Third, unlike for phonetic units, ASWU pronunciations cannot be synthesized for words infrequently observed in the training data. The proposed algorithm, summarized in section 6.1.3, can be used to design a hybrid system using phonetic units to obtain the desired ability to generalize to infrequently observed words and automatic units to provide more accurate acoustic models for frequent words.

### 6.1.1 Large Automatically Derived Unit Inventories

In chapter 3, a *joint* design algorithm for unit inventory, unit models and lexicon is proposed. The proposed algorithm is similar to previously developed unit design algorithm but differs in three respects:

- The unit design is constrained to fit a limited complexity pronunciation model.
- The unit design is performed in iterative progressive refinement steps rather than a single step design.
- A likelihood criterion is consistently used throughout the system design.

The first step of the algorithm bootstraps the design, then in iterative refinement steps, the final unit inventory, unit models and corresponding lexicon are obtained. The bootstrap phase itself is a two step process equivalent to previous unit design algorithms. The key difference of the algorithm described here in comparison to previous work is that pronunciation constraints are introduced in the design. A limitation

of the current approach is that the constraint only allows design of a single linear pronunciation per word, but extensions to allow multiple pronunciations are straightforward, as discussed in section 6.2. Due to the use of pronunciation constraints, the unit inventory and lexicon are designed jointly which guarantees a matched condition between the two. The constraints also make the algorithm more suitable for the design of unit inventories on large corpora, because the storage and computational requirements of the algorithm are now proportional to the vocabulary size rather than the corpus size.

The first step of the bootstrap algorithm segments the training data and attempts to find stationary regions in the speech signal, as appropriate for the model assuming piecewise Gaussian stationarity (HMMs). A second clustering step provides a quantized representation of the acoustic space spanned by the acoustic segments. The acoustic segmentation uses, in contrast to previous work, a likelihood objective as segments hypothesized in the search are evaluated using the segment likelihood. To avoid possible data sparsity problems, a fixed covariance was used in the segmentation step. Both covariances estimated on a per-utterance basis as well as a grand variance approach (i.e. estimating the covariance from the complete training corpus) were investigated. Furthermore, to obtain a non-trivial segmentation from the acoustic segmentation step a thresholding approach is required. Thresholding based on average likelihood per frame as well as an information theoretical measure (MDL) were investigated.

The constraints imposed between the segmentation and clustering stages can be decomposed in two types of constraints. First after acoustic segmentation, a **pronunciation length** constraint is imposed that guarantees that all observations of a lexical item in the training data are segmented in an equal number of segments. The pronunciation lengths of the lexical items are derived by aligning an acoustic segmentation without constraints with a word-level segmentation. The median number of segments seen across examples of a lexical item is then used to (heuristically)

define its pronunciation length. A second acoustic segmentation operates under the imposed constraints and ensures that segment boundaries coincide with word boundaries and that all examples of a word are segmented using a consistent number of segments. The second **pronunciation consistency** constraint pre-groups the data originating from the same word-position to prevent such groups of data from division over clusters (unit models) in the clustering process. This constraint guarantees that all data from a particular word-position will be assigned to a single cluster (unit model), implicitly defining the pronunciation of that word-position. In contrast to previous work, the clustering of the pre-grouped data therefore **jointly** designs the unit inventory by means of the data partitioning, the unit models by estimation of the cluster centers and the lexicon by the distribution of the data groups over the cluster inventory. Using the unit inventory and lexicon derived by the clustering step, the bootstrap process is completed by Viterbi training of the clustered unit models; iteratively re-adjusting segment boundaries on the basis of the last unit models and then re-estimating unit models on the basis of the last segmentation.

Several options were investigated for the iterative refinement process. All increased the system complexity in stages, allowing segmentation adjustments to match the unit inventory derived by the last clustering step. Experimental results showed the importance of appropriately adjusting the temporal resolution of the system throughout the system refinement procedure. In a binary temporal adjustment approach, the temporal resolution was refined by means of splitting segments by acoustic segmentation at some refinement stages but not all. The variable temporal adjustment approach adjusts the temporal resolution at each refinement stage by executing the constrained acoustic segmentation algorithm but letting the last Viterbi segmentation function as the word-level segmentation. New median lengths are now computed for each unique unit in the last Viterbi segmentation instead of for each unique word. The clustering is still performed on the basis of the word-position to maintain a fine grained representation of the training data. Using this approach, a hierarchical relationship exists

between the units in the inventories derived at different iterations of the refinement process.

The temporal resolution of the system was found to have a large effect on the system performance. The systems that did not allow temporal resolution adjustment performed worse than systems that did, even with varied starting points providing a high temporal resolution to compensate for the loss of resolution incurred in the iterative refinement process (performance differed with more than 2%). Among the systems that did allow temporal resolution adjustments during the iterative refinements, the best performance was obtained by the system that adjusted the temporal resolution at every refinement step by variable temporal refinement (i.e. re-aligned the last Viterbi segmentation with an acoustic segmentation at every refinement stage).

The automatic unit systems outperformed the phonetic unit systems both at low and high complexity. For comparable systems at low complexity (a context-independent phone unit versus a 150 automatic unit based system) error rates were reduced by 19% (24.4% error for the phonetic unit based system vs. 19.7% error for the automatic unit based system). At high complexity an error rate reduction of 5% (11.9% error for the phonetic unit based system vs. 10.4% error for the automatic unit based system) was obtained. The difference between the low complexity systems is significant with confidence of 95% but the difference between the high complexity systems is not significant. The better performance shows that the use of automatically derived units result in a better designed set of units than those designed manually. The difference at low complexity is particularly important for constrained resource (e.g. hand-held) systems.

The computational efficiency was obtained by imposing the pronunciation length consistency constraint. This heuristically determined constraint is a significant difference between the algorithm proposed here and previous work. The results of the low complexity systems are comparable to the systems based on automatic units designed without the heuristically motivated pronunciation length constraint which

indicates that no severe performance degradation due to this heuristic is introduced. The improved performance of the high complexity system shows the effectiveness of the proposed algorithm for the design of large unit inventories. The larger gain of automatic units over phonetic units at low complexity compared to high complexity shows that, at high complexity, the distributions for the sub-optimally defined unit models can capture the acoustic variability even though inflated due to the sub-optimal choice of units. However, for spontaneous speech the increased complexity may come at an unnecessarily high cost.

Experiments showed little performance differences when the acoustic segmentation parameters were varied. Thresholding using an information theoretical thresholding approach (weighted MDL based) seemed to perform slightly worse than an average likelihood per frame threshold (75.6% vs. 76.8% accuracy). Also the use of a per utterance estimated covariance compared to a grand covariance approach resulted in small performance differences ( $< 1\%$  accuracy). However, the small differences could be an artifact of the corpus, as the recording conditions within the Resource Management corpus are well controlled.

### 6.1.2 Explicit Context Modeling

In order to demonstrate performance competitive with state-of-the-art phone-based systems, we explored several approaches to explicit modeling of context, in chapter 4. The approaches are similar to the implicit context modeling approach described in chapter 3 in the sense that sufficient statistics are computed and shared distributions for context-dependent units are derived by clustering and Viterbi training. The main difference between the implicit and explicit context models is the atomic group definition that sufficient statistics are computed for. In the implicit modeling case, the atomic groups were defined according to word-position; whereas, in the explicit modeling case, atomic groups are defined by unique ASWU contexts. An important motivation for modeling context explicitly is that it allows cross-word contextual

effects to be included in the model which cannot be included in the implicit (word-position-based) context model.

Several aspects that impact the performance of a system that models context explicitly were investigated. First, the context definition determines the number of unique context-dependent units and therefore determines the number of sufficient statistics that are used to represent the training data. Second, the temporal resolution of the context-dependent system is determined by the unit inventory on which the context-dependent units are based. The definition of an explicit context for the base units provides a contextual refinement but will not allow the temporal resolution of the system to change from that of the base unit inventory. Third context can be defined as either local, using the directly neighboring ASWUs or more distant using ASWUs further away (analogous to the context definition used in phone-based systems). The distant context definition can be derived within the ASWU framework by taking advantage of the hierarchical relationship between unit inventories derived at different stages of the design of the base unit inventory. Fourth, out of computational and storage considerations, the clustering stage can be constrained to allow only a limited set of sharing scenarios rather than all possible sharing scenarios. Most phone based system for example, allow sharing only among different context-dependent versions of the same center unit. Fifth, the explicit modeling of context allows an extension of the context model to provide distributions for context-dependent units that were not seen in training. This generalization allows for the modeling of contextual effect across word-boundaries which cannot be achieved within the implicit context modeling approach. The proposed algorithm learns groups of equivalent conditioning factors from data. Since the learned groups are derived by considering the conditioning effect of units, they are well suited for use in the design of a subsequent predictor model that uses these groups to gain the ability to synthesize distributions for unobserved contexts.

Experimental results showed that the granularity of the representation of the

training data (number of atomic units) and temporal resolution of the base unit inventory have a large effect on the performance of a context-dependent system. Comparing two systems that model the local context to the left and right (“tri-unit” systems), one based on the 124 unit ASWU system and the other based on the 635 unit ASWU system, the system based on the 635 unit base inventory performed significantly better (90.4% vs. 86.3%). The system based on the 635 unit inventory used 13k sufficient statistics in comparison to 6.3k sufficient statistics for the tri-unit system based on the 124 unit inventory.

Experiments also showed improved performance of a system that uses a more distant context in comparison to a system based on a local context. Comparing the 635 ASWU based context-dependent system that uses a local context window of size five (considering two units to the left and right of the center unit), with the 743 unit based system that uses a more distant context, the distant context system outperforms the local context system (91.6% accuracy vs. 90.4% accuracy). The base unit inventories have approximately the same temporal resolution and the number of atomic units for the distant context system was smaller than that of the local context system (18k vs. 20k) indicating that the distant context is more useful for the contextual conditioning of distributions.

Constraining sharing scenarios to allow only sharing among different context-dependent units of the same center unit did not affect performance much (less than 0.5% accuracy) indicating that sharing of context-dependent units from different center units is not crucial. The computational requirement of the constrained clustering process was approximately six time lower than that of the unconstrained process.

Experiments using the context classes that were learned from data by a parallel clustering step show the effectiveness of the proposed algorithm. The word-internal system using decision tree clustered distributions achieved equivalent performance in comparison to the system that used clustering without constraints (91.2% accuracy for the implicit context system vs. 91.6% for the decision tree clustered system).

This indicates that the defined classes did not exclude essential sharing scenarios even though much fewer sharing scenarios were considered in comparison to the implicit context system that did not constrain sharing scenarios. The additional gain obtained by modeling cross-word effects (92.1% accuracy) showed that the learned classes provide useful information about the correlation between units when they appear in the context of others. Allowing the modeling of cross-word contextual effects in this way provides a novel extension to the ASWU framework. In addition, the learning of groups of equivalent conditioning factors provides a mechanism to extend pattern recognition techniques where, unlike phonetic modeling in speech, there is no knowledge of the relation between conditioning classes available.

### **6.1.3 Extension to Large Vocabulary Systems**

The final area of focus was to apply the automatically derived units in a large vocabulary setting. Since direct application of automatic units for this type of task is infeasible due to the training data requirements, an algorithm was developed (described in detail in chapter 5) for the design of a hybrid system. In this hybrid system, phonetic units are used for infrequently observed units taking advantage of their ability to generalize to unseen events. Automatic units were used for the most frequent entries that provided sufficient examples for reliable pronunciation design, providing a detailed acoustic model for those entries. The approach first uses the training data for the unit model design (both phonetic as well as automatic), and then uses the training data to select which model type is most likely for the most frequent lexical entries. In the pronunciation selecting step, the most frequent lexical entries are represented in parallel by their phonetic and automatic unit models and the likelihood of using one or the other model is computed in an estimation step. The effectiveness of the automatically derived units together with this selection process is illustrated by the accuracy improvement obtained. Using the most likely model resulted in a 0.5% accuracy gain over the phonetic unit baseline. Re-estimating the parameters of

the automatically derived units in an additional step resulted in an additional 1.4% accuracy improvement. The estimation step also confirmed that for most words, the automatically derived units and lexicon are a better fit to the data in comparison to phonetic units as most lexical items were estimated as more likely when modeled by automatic units.

## 6.2 Future Work

The work presented in this thesis can be extended in several ways. A limitation of the developed algorithms is that they focus on the design of linear single pronunciations. In spontaneous speech, phone segments can be modified dramatically and are frequently dropped completely [25], so that a single linear pronunciation is likely to be inadequate for representing many words. The impact of the linear pronunciation constraint was limited by considering lexical items with differing phonetic pronunciations as unique lexical items but an extension of the algorithms to automatically learn multiple pronunciations where appropriate will yield an additional accuracy improvement of the acoustic model as also demonstrated in [29]. To extend the described algorithm, contextual splitting of the data groups, now formed for each unique word-position, would allow multiple pronunciations to be learned. The biggest problem extending this approach is in the initial definition of the atomic units where decisions need to be made on which of the contextually split data groups a segment of a word token belongs to. Where in the single pronunciation case, segments from different tokens but the same word-position all belong to one group, in the multiple pronunciation case, multiple groups exist corresponding to potential pronunciation variants.

The hybrid approach proposed in this thesis can be extended to incorporate cross-word contextual effects. This requires that the decision tree predictors are to be extended to allow the union of both phonetic as well as automatically derived units to appear in the context of units. In addition, the approach can be extended to an

alternative approach for modeling of cross-word contextual effects that uses multi-word lexical entries with different pronunciations (e.g. [46]). Given an algorithm for learning multiple ASWU pronunciations, it is straightforward to combine this with multi-word lexical entries and can be combined with an acoustically motivated definition of the multi-word set [4].

The segmentation and clustering algorithms can be implemented for polynomial mean trajectory segment models in general [4, 34], but for simplicity the experiments and equations given in this thesis correspond to the special case of a hidden Markov modeling, i.e. a constant mean trajectory. Combining higher order models with the progressive refinement of the unit inventories which provides a hierarchical relationship between units derived at different stages of refinement, the units can be used in a multi-pass search approach. In such an approach, coarser (and lower order) models can be used for an initial pass to limit the search space of a subsequent pass that uses finer (and higher order) units. As the multi-pass search paradigm reduces the computational cost of the later passes, it becomes computationally feasible to use these higher order models.

Another possible extension of the proposed work is to change the design criterion. By definition, phonetic units are focussed on discriminatory qualities rather than data likelihood. To achieve analogous results in an automatic approach, a discriminant criterion function can be used in the unit design process.

In conclusion, the work presented in this thesis allows application of automatically derived units in a larger number of tasks than previously developed algorithms. The strong points of the ASWU approach, which itself is not new, include more detailed acoustic models especially at low complexity and less manual (and possibly error-prone) effort. The described algorithm particularly addresses the weaknesses of the automatic unit framework providing methods that are capable of deriving large unit inventories and that allow generalization to unseen events which were previously not addressed in other work on automatically derived units. As a result, many ap-

plications can now directly benefit by use of the described algorithms (for example, telephone based stock quote information systems and automatic handwriting recognition systems).

# Bibliography

- [1] F. Alleva, X. Huang and M. Hwang, “An improved Search Algorithm Using Incremental Knowledge for Continuous Speech Recognition,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 307-310, 1993.
- [2] T. Anderson, *An Introduction to Multivariate Statistical Analysis*, J. Wiley & Sons, New York, 1984.
- [3] T. Kamm, G. Andreou and J. Cohen, “Vocal Tract Normalization in Speech Recognition: Compensating for Systematic Speaker Variability,” *Proc. of the 15th Annual Speech Research Symposium*, pp. 161-167, CLSP, Johns Hopkins University, Baltimore, MD, June 1995.
- [4] M. Bacchiani, M. Ostendorf, Y. Sagisaka, and K. Paliwal, “Design of a speech recognition system based on non-uniform segmental units,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 443-446, 1996.
- [5] L. Bahl, P. Brown, P. de Souza, R. Mercer and M. Picheny, “Acoustic Markov models used in the Tangora speech recognition system,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 497-500, 1988.

- [6] L. Bahl, P. de Souza, P. Gopalakrishnan, D. Nahamoo and M. Picheny, "Decision trees for phonological rules in continuous speech," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 185-188, 1991.
- [7] L. Bahl *et al.*, "Automatic phonetic baseform determination," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 173-176, 1991.
- [8] L. Bahl, P. Brown, P. de Souza, R. Mercer and M. Picheny, "A method for the construction of acoustic Markov models for words," *IEEE Transactions on Speech and Audio Processing*, Vol. 1, no. 4, pp. 443-452, 1993.
- [9] L. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Annals of Mathematical Statistics*, Vol. 37, 1554-1563, 1966.
- [10] L. Baum, T. Petrie, G. Soules and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov Chains," *Annals of Mathematical Statistics*, Vol. 41, no. 1, pp. 164-171, 1970.
- [11] R. Bellman, "Dynamic Programming," Princeton University Press, Princeton, New Jersey, USA, 1957.
- [12] L. Breiman, J. Friedman, R. Olshen and C. Stone, "Classification and Regression Trees," Wadsworth & Brook/Cole, Monterey, 1984.
- [13] N. Campbell "Synthesizing Spontaneous Speech," in Y. Sagisaka, N. Campbell and N. Higuchi editors, *Computing Prosody*, Springer, New York, 1997.
- [14] P. Chou "Optimal Partitioning for Classification and Regression Trees," *IEEE Trans. PAMI*, Vol. 13, no. 4, pp. 340-354, 1991.

- [15] Y.-L. Chow and R. Schwartz, “The N-Best Algorithm: An Efficient Procedure for Finding Top N Sentence Hypotheses,” *Proceedings of the Second DARPA Workshop on Speech and Natural Language*, pp. 199–202, 1989.
- [16] M. Cohen, “Phonological structures for speech recognition,” Ph.D. thesis, University of California, Berkeley, 1989.
- [17] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 28, no. 4, pp. 357-366, 1980.
- [18] L. Deng, M. Aksmanovic, D. Sun and J. Wu, “Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states,” *IEEE Transactions on Speech and Audio Processing*, Vol. 2, no. 4, pp. 507-520, 1994.
- [19] A. Dempster, N. Laird and D. Rubin, “Maximum Likelihood Estimation from Incomplete Data,” *Journal of the Royal Statistical Society (B)*, Vol. 39, no. 1, pp. 1-38, 1977.
- [20] V. Digalakis and H. Murveit, “Genones: Optimizing the degree of tying in a large vocabulary HMM-based speech recognizer,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. I, pp. 537-540, 1994.
- [21] V. Digalakis, D. Rtischev and L. Neumeyer, “Fast speaker adaptation using constrained estimation of Gaussian mixtures,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 3, no. 5, pp. 357-366, 1995.
- [22] E. Eide and H. Gish, “A Parametric Approach to Vocal Tract Length Normalization,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 346-348 1996.

- [23] H. Gish and K. Ng, "A segmental speech model with applications to word spotting," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. II, pp. 447-450, 1993.
- [24] J. Godfrey, E. Holliman and J. McDaniel, "Switchboard: Telephone Speech Corpus for Research and Development," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 517-520, 1992.
- [25] S. Greenberg, "Speaking in shorthand – a syllable-centric perspective for understanding pronunciation variation," *Proc. ESCA Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, pp. 47-56, 1998.
- [26] H. Hattori and S. Sagayama, "Vector field smoothing principle for speaker adaptation," In *Proceedings of the International Conference on Spoken Language Processing*, pp 381-384, 1992.
- [27] H. Hermansky and S. Sharma, "TempoRAI Patterns (TRAPs) In ASR Of Noisy Speech," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 289-292, 1999.
- [28] T. Holter and T. Svendsen, "Combined optimisation of baseforms and model parameters in speech recognition based on acoustic subword units," In *Proceedings of the IEEE Workshop on Automatic Speech Recognition*, pp. 199-206, 1997.
- [29] T. Holter and T. Svendsen, "Maximum likelihood modeling of pronunciation variation," In *Proceedings of the ESCA Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, pp. 63-66, 1998.
- [30] HTK, version 2.1, Cambridge University, 1997.
- [31] J. Humphries *et al.*, "Using accent-specific pronunciation modelling for robust speech recognition" In *Proceedings of the International Conference on Spoken Language Processing*, p. 2324, 1996.

- [32] M.-Y. Hwang and X. Huang, "Subphonetic Modeling for Speech Recognition," *Proc. DARPA Workshop on Speech and Natural Language*, pp. 174-179, 1992.
- [33] A. Kannan, M. Ostendorf and J. R. Rohlicek, "Maximum Likelihood Clustering of Gaussians for Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, Vol. 2, no. 3, pp. 453-455, 1994.
- [34] A. Kannan and M. Ostendorf, "A comparison of constrained trajectory models for large vocabulary speech recognition," *IEEE Transactions on Speech and Audio Processing*, Vol. 6, no. 3, pp. 303-306, 1998.
- [35] L. Lamel, R. Kassel and S. Seneff, "Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus," *Proceedings of the DARPA Speech Recognition Workshop*, pp. 100-109, Report no. SAIC-86/1546, 1986.
- [36] C.-H. Lee, F. Soong and B.-H. Juang, "A segment model based approach to speech recognition," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 501-504, 1988.
- [37] C.-H. Lee, B.-H. Juang, F. K. Soong, and L. Rabiner, "Word recognition using whole word and subword models," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 683-686, 1989.
- [38] C.-H. Lee and J.-L. Gauvain, "Speaker adaptation based on MAP estimation of HMM parameters," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 558-561, 1993.
- [39] L. Lee and R. Rose, "Speaker Normalization using Efficient Frequency Warping Procedures," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 353-356, 1996.

- [40] C. Leggetter and P. Woodland, “Flexible speaker adaptation using maximum likelihood linear regression,” *Proc. ARPA Workshop on Spoken Language Technology*, pp. 110-115, 1995.
- [41] J. Lucassen and R. Mercer, “An information theoretic approach to the automatic determination of phonemic baseforms,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. III, pp. 42.5.1-42.5.4, 1984.
- [42] Miller *et al.*, BBN submission, LVCSR meeting, Baltimore, MD, 1997.
- [43] H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, “Large-Vocabulary Dictation Using SRI’s DECIPHER<sup>TM</sup> Speech Recognition System: Progressive Search Technique,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 319-322, 1993.
- [44] L. Nguyen, R. Schwartz, Y. Zhao and G. Zavaliagos, “Is *N*-Best Dead?,” *Proc. ARPA Workshop on Human Language Technology*, 1994.
- [45] NIST evaluation on the Switchboard and Callhome english spontaneous human to human dialogues, *LVCSR Hub 5 workshop*, Baltimore, 1996.
- [46] H. Nock and S. Young, “Detecting and correcting poor pronunciations for multiword units,” *Proc. ESCA Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, pp. 85-90, 1998.
- [47] M. Ostendorf, F. Richardson, S. Tibrewal, R. Iyer, O. Kimbal and J. Rohlicek, “Stochastic Segment Modeling for CSR: The BU WSJ Benchmark System,” *Proc. ARPA Workshop on Spoken Language Technology*, 1994.
- [48] M. Ostendorf, B. Byrne, M. Bacchiani, M. Finke, A. Gunawardana, K. Ross, S. Roweis, E. Shriberg, D. Talkin, A. Waibel, B. Wheatley and T. Zeppenfeld,

- “Modeling Systematic Variations in Pronunciation via a Language-Dependent Hidden Speaking Mode,” Boston University Technical Report no. 97-002, 1997.
- [49] M. Ostendorf and H. Singer, “HMM topology design using maximum likelihood successive state splitting,” *Computer Speech and Language*, 11, no. 1, pp. 17-42, 1997.
- [50] K. Paliwal, “Lexicon building methods for an acoustic sub-word based speech recognizer,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 729-732, 1990.
- [51] D. Pallett *et al.*, “1994 Benchmark Tests for the ARPA Spoken Language Program,” *Proc. ARPA Workshop on Spoken Language Technology*, pp. 5-38, 1995.
- [52] P. Price, W. Fisher, J. Bernstein and D. Pallett, “The DARPA 1000-Word Resource Management database for continuous speech recognition,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 651-654, 1988.
- [53] L. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proceedings of the IEEE* Vol. 77, no. 2, pp. 257-286, 1989.
- [54] L. Rabiner, J. Wilpon and F. Soong, “High Performance Connected Digit Recognition Using Hidden Markov Models,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, no. 8, pp. 1214-1225, 1989.
- [55] M. Riley, “A statistical model for generating pronunciation networks,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. II, pp. S11.1-S11.4, 1991.
- [56] M. Riley, W. Byrne, M. Finke, S. Khudanpur, A. Ljolje, J. McDonough, H. Nock, M. Saranclar, C. Wooters and G. Zavaliagos, “Stochastic Pronunciation

- Modelling from Hand-Labelled Phonetic Corpora,” *Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, pp. 7-12, 1998.
- [57] R. Roth, L. Gillick, J. Orloff, F. Scattone, G. Gao, S. Wegmann and J. Baker, “Dragon systems’ 1994 Large Vocabulary Continuous Speech Recognizer,” *Proc. ARPA Workshop on Spoken Language Technology*, pp. 116-210, 1995.
- [58] R. Schwartz and Y. Austin, “A Comparison of Several Approximate Algorithms for Finding Multiple (N-Best) Sentence Hypotheses,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp.701-704, 1990.
- [59] Y. Shiraki and M. Honda, “LPC speech coding based on variable-length segment quantization,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 36, no.9, pp. 1437-1444, 1988.
- [60] T. Sloboda and A. Waibel, “Dictionary learning for spontaneous speech recognition,” In *Proceedings of the International Conference on Spoken Language Processing*, p.2328, 1996.
- [61] S. Stevens and J. Volkman, “The relation of pitch of frequency: A revised scale,” *American Journal of Psycholinguistics*, Vol. 53, pp. 329-353, 1940.
- [62] T. Svendsen and F. Soong, “On the automatic segmentation of speech signals,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 77-80, 1987.
- [63] T. Svendsen, K. Paliwal, E. Harborg, and P.O. Husøy, “An improved subword based speech recognizer,” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 108-111, 1989.

- [64] T. Svendsen, F. Soong, and H. Purnhagen, "Optimizing baseforms for HMM-based speech recognition," In *Proceedings European Conference on Speech Communication and Technology*, Vol. 1, pp. 783-786, 1995.
- [65] J. Takami and S. Sagayama, "A successive state splitting algorithm for efficient allophone modeling," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. I, pp. 573-576, 1992.
- [66] S. Wegmann, D. McAllaster, J. Orloff and B. Peskin, "Speaker Normalization on Conversational Telephone Speech," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 339-341, 1996.
- [67] M. Weintraub, K. Taussig, K. Smith and A. Snodgrass, "Effect of Speaking Style on LVCSR Performance," In *Proceedings of the International Conference on Spoken Language Processing*, 1996.
- [68] M. Weintraub, E. Fossler, C. Galles, Y. Kao, S. Khudanpur, M. Saraclar and S. Wegmann, "Automatic Learning of Word Pronunciation from Data," WS96 project report, Johns Hopkins University, Baltimore, 1996.
- [69] C. Westendorf and J. Jelitto, "Learning pronunciation dictionary from speech data," In *Proceedings of the International Conference on Spoken Language Processing*, p.1045, 1996.
- [70] J. Wilpon, C. Lee and L. Rabiner, "Improvements in Connected Digit Recognition Using Higher Order Spectral and Energy Features," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 349-352, 1991.
- [71] C. Wooters and A. Stolcke, "Multiple-pronunciation lexical modeling in a speaker independent speech understanding system," In *Proceedings of the International Conference on Spoken Language Processing*, pp. 1363-1366, 1994.

- [72] P. Woodland and S. Young, "The HTK tied-state continuous speech recogniser," In *Proceedings European Conference on Speech Communication and Technology*, Vol. 3, pp. 2207-2210, 1993.
- [73] S. Young and P. Woodland, "The use of state tying in continuous speech recognition," In *Proceedings European Conference on Speech Communication and Technology*, Vol. 3, pp. 2203-2206, 1993.
- [74] S. Young, J. Odell and P. Woodland, "Tree-based state tying for high accuracy acoustic modeling," *Proc. ARPA Workshop on Human Language Technology*, pp. 307-312, 1994.
- [75] P. Zhan and M. Westphal, "Speaker Normalization Based on Frequency Warping," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 1039-1042, 1997.

## **Vita**

Michiel Adriaan Unico Bacchiani was born in Amsterdam, The Netherlands on October 17, 1967. In 1994 he received the “ingenieurs” (ir.) degree in Electrical Engineering from the Technische Universiteit Eindhoven (Technical University Eindhoven), Eindhoven, The Netherlands. He entered the Department of Electrical and Computer Engineering at Boston University in 1996 and obtained a doctorate in 1999.