

Clustering Wide-Contexts and HMM Topologies
for Spontaneous Speech Recognition

Izhak Shafran

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2001

Program Authorized to Offer Degree: Electrical Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Izhak Shafran

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

Mari Ostendorf

Reading Committee:

Jeff Bilmes

William Byrne

Date:

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Bell and Howell Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature_____

Date_____

University of Washington

Abstract

Clustering Wide-Contexts and HMM Topologies
for Spontaneous Speech Recognition

by Izhak Shafran

Chair of Supervisory Committee:

Professor Mari Ostendorf
Electrical Engineering

In most speech recognition systems today, all the acoustic variation associated with a phoneme is characterized in terms of the identity of its neighboring phonemes. The neighbors influence only the state observation density of a fixed Hidden Markov Model. Other sources of variation are captured implicitly by using Gaussian mixture models for the state observations. Consequently, these models can be very broad, particularly for casual spontaneous speech. In this thesis, we explore conditioning of phonemes on higher level linguistic structure, specifically syllable- and word-level structure to learn models for phonemes that are more specific to the context, reporting experimental results on a large vocabulary (35k words) conversational speech task (Switchboard).

In particular, this thesis makes three main contributions related to wide context conditioning. First, we demonstrate that syllable- and word-level structure can be incorporated into current acoustic models to improve recognition accuracy over triphones. For a fixed number of parameters, these models are computationally more efficient than pentaphones, both in training and in testing. In addition, use of syllable and word features leads to a small but significant improvement in performance. The wide-contexts used in our acoustic model can implicitly capture re-syllabification effects to a certain extent. However, we find that

explicitly modeling re-syllabification does not improve recognition further, because there are only a small number of phones that exhibit acoustic difference after re-syllabification.

The second contribution addresses the difficulties that arise when a large number of additional conditioning features are used. As the number of conditioning features increases, the training cost can increase exponentially. Moreover, a large fraction of the training labels tends to have too few examples to have reliable statistics associated with them, and this could potentially cause decision trees to learn bad clusters. A new method has been developed for clustering with multiple stages, where each stage clusters a different subset of features, and also has a choice of using the partitions learned in the previous stages. Apart from reducing the risk of unreliable statistics, it is designed to ameliorate data fragmentation problem and is computationally less expensive. This method was successfully demonstrated with pentaphones, resulting in equivalent performance at a lower cost.

Finally, a new algorithm is described to design context-specific HMMs. The idea is to model reduction of a phone for certain contexts, and to learn a more constrained topology. Using contextual information, the algorithm clusters HMM paths where each path has a different number of states. An HMM distance measure has been formulated to prune out the paths which are similar. During decoding, the paths are allocated dynamically for each sub-word unit according to their context. We investigated this algorithm to model phone topologies, finding improved characterization of speech given known word sequences but no significant improvement in word error rate.

TABLE OF CONTENTS

List of Figures	iv
List of Tables	vi
Chapter 1: Introduction	1
1.1 Modeling Contextual Variation with Syllable and Word Features	4
1.2 Modeling Temporal Variation with Flexible Topology	6
1.3 Outline of the Thesis	8
Chapter 2: An Automatic Speech Recognition System	11
2.1 Feature Extraction	13
2.2 Language Model	14
2.3 Elementary Models	15
2.3.1 HMM Parameter Estimation	17
2.3.2 Optimal HMM State Sequence	20
2.4 Acoustic Model for Word Sequences	21
2.5 Large Vocabulary Search	25
Chapter 3: Experimental Paradigm	27
3.1 Task	27
3.1.1 Training Corpus	27
3.1.2 Recognition Test	28
3.2 Recognition System	28
3.3 Baseline Performance	31

Chapter 4:	Using Syllable and Word Feature	33
4.1	Beyond Triphones	33
4.2	Using Rich Syllable Feature	37
4.2.1	Syllable Features	38
4.2.2	Results and Discussion	40
4.2.3	Observations on the Use of Syllable and Word Features	41
4.3	Re-syllabification	43
4.3.1	Scheme for re-syllabification	43
4.3.2	Experiments, Results and Discussion	45
4.4	Summary	47
Chapter 5:	Multi-stage clustering	49
5.1	Clustering with Large Feature Set	49
5.2	Multi-stage Clustering	51
5.3	Experiments and Discussion	54
5.4	Summary	56
Chapter 6:	Topology Refinement	57
6.1	Learning HMM Topologies	57
6.2	An Algorithm to Cluster HMMs	59
6.2.1	Distance Metric for HMMs	61
6.2.2	Mapping Contexts to HMMs	65
6.3	Experiments and Results	66
6.4	Summary	72
Chapter 7:	Conclusions and Future Directions	75
7.1	Contributions and Conclusions	75
7.1.1	Modeling Contextual Variation with Syllable and Word Features	75

7.1.2	Multi-stage Clustering for Incorporating Wide Contexts	76
7.1.3	Modeling Temporal Variations with Flexible Topology	77
7.2	Implications and Future Directions	77
	Bibliography	80
	Appendix A: Questions on Syllable- and Word-level Features	96
	Appendix B: Cross-entropy between single Gaussian distributions	98

LIST OF FIGURES

1.1	Estimated distribution of phone duration in conversational speech (tail omitted for better display of mode).	7
2.1	Overview of an ASR system based on a statistical framework.	12
2.2	An example left-to-right HMM used to model a sub-word unit	22
2.3	Mapping a state of a sub-word unit to a distribution with a decision tree. . .	24
3.1	Sequence of steps in training acoustic models.	29
3.2	Sharing covariances and means at different levels in decision trees.	30
4.1	Re-syllabification of “mark all” using the Sonority Dispersion Principle. . . .	44
5.1	Increase in training complexity with features - triphone through pentaphone.	50
5.2	Multi-stage clustering illustrated with two stages.	52
6.1	A set of candidate HMM paths for a context-dependent phoneme.	59
6.2	Computing $D(h_1, h_2)$ by evaluating likelihoods over all examples of h_1 and h_2 in the training data using both h_1 and h_2	62
6.3	Computing $D(h_1, h_2)$ by evaluating the likelihood of all the sequences that can be synthesized with h_1 and h_2	62
6.4	Representation of product space of two HMMs x and w for computing $p(a(w) v)$.	64
6.5	Initializing the parallel HMM paths from baseline fixed topology.	66
6.6	Average log-likelihood on the training data (male and female speakers) using all three HMM paths.	67

6.7	Average entropy of the distribution of path probabilities (male and female speakers).	68
-----	---	----

LIST OF TABLES

3.1	Performance comparison on 1998 NIST Development Task.	31
4.1	Coding of syllable- and word-level features in the dictionary.	39
4.2	Increase in size of sub-word unit inventory with features (160 hrs)	39
4.3	Comparing the average log-likelihood on the training data and on a held-out data set.	40
4.4	Word error rates using different features in state clustering.	41
4.5	Usage of syllable and word features in automatically trained decision trees.	42
4.6	Sonority rank for phonemes used in our work.	44
4.7	Examples of re-syllabifications chosen in the training data.	46
5.1	Average log-likelihood on a held-out data set using one vs. two stages of clustering.	55
5.2	Word error rates of systems trained with one vs. two stages of clustering.	55
6.1	Recognition results on mapping context to HMM paths.	70
6.2	Average log-likelihood using different multi-path HMMs.	70
6.3	Change in word errors, computed with string alignment; negative counts represent improvement with the topology modeling.	71
6.4	Recognition results from a hybrid approach.	72
A.1	Questions about position of phone (syllable) in word, asked on the phone and its immediate neighbors	96
A.2	Questions about lexical stress, asked on the phone and its immediate neighbors	96

A.3 Questions about phone in syllable 97

ACKNOWLEDGMENTS

Foremost, I would like to express my gratitude to Mari Ostendorf, my thesis supervisor, for providing me an opportunity to work in speech recognition. She introduced me to this exciting area of research with her extraordinary ability to communicate complex ideas in simple terms. She guided me throughout the thesis work. She has been very meticulous about our work and read all the manuscripts numerous times. I am also thankful for her support, her patience and for creating a congenial lab environment.

I would like to thank all the members of my committee for reading the manuscript and suggesting improvements. Katrin Kirchhoff has been generous with her time, and provided me with many rounds of feedback, from thesis proposal to dissertation. The work on re-syllabifications was largely motivated by her suggestions. Richard Wright was kind enough to give me short tutorials on various linguistic phenomena that occur in conversational speech, and to point me to numerous useful papers. Jeff Bilmes helped me think about statistical problems in a different light, particularly using graphical models. Bill Byrne gave useful suggestions for improvement, particularly in the presentation of certain chapters.

I would like to thank all the members of the lab (both SSLI lab and the former SPI lab at Boston University), for contributing to an inspiring and pleasant atmosphere. I will always remember the good times spent with my colleagues - Becky Bates, Ivan Bulyko, Ozgur Cetin, Randy Fish and David Palmer, especially after moving from Boston to Seattle. Thanks to Becky for the many installments of lab-wide ballet lessons, much of which I have not retained, unfortunately. Many pleasant memories are also associated with lunch hours discussions at Elephant Walk with Michiel Bacchiani, Cam Fordyce, Manhung Siu, Mujdat Cetin and the unforgettable Harihara Sivakumar. Chia-Ping Chen, Scott Otterson, Sonia

Parandekar and many others colleagues have made the lab a very lively place to be. I am glad that Michiel Bacchiani is a good code-writing machine. He has contributed many useful libraries and tools for this work.

I would also like to acknowledge the speech group at GTE (formerly BBN) for providing VTLN feature vectors for Switchboard corpus and N-best hypothesis for test sets. Owen Kimbal and Rukimini Iyer were particularly helpful in this regard. Michael Riley and Michiel Bacchian from AT&T helped us with FST libraries and an FST-based decoder. Thanks to Dan Ellis and Shawn Chang for help with ICSI transcripts.

I have also been fortunate to be surrounded by people who have helped me go all this way. This includes both family, friends and teachers. I am thankful to my father for his moral support and for prodding me with - “what will you be learning next?”. Last, but not least, I am indebted to Rachel, my wife. She has enriched my life as a wonderful companion in a multitude of facets including an enjoyable social life and many rejuvenating travel escapades. I am thankful for her love, support, understanding and for urging me to finish.

This work was supported by the National Science Foundation, grant no. ISI-9618926.

Chapter 1

INTRODUCTION

Fifty years ago, Arthur C. Clarke portrayed a machine, *HAL 2000* which became immensely popular for its ability to listen and speak to humans [21, 22]. In an age when ENIAC¹, the first computer, had just been invented and programming a computer required special skills, the popularity of *HAL 2000* illustrated the universal appeal of speech as an interface to machines. Today, in the year 2001, computers are ubiquitous and are managing an increasingly large number of tasks that range from mundane to critical. They are also gateways to a huge repository of information available on the internet. To provide access to these resources for the vast majority of humanity, it is necessary to develop easy, universal, and multiple modes of interaction with computers, and a necessary part of such an interface is speech.

The task of understanding human speech by a computer is typically broken down into two subtasks, namely, automatic speech recognition and natural language processing. The former transcribes the audio speech signal into words, while the latter interprets these transcriptions. This thesis will focus on the problem of automatic speech recognition.

In the last two decades considerable progress has been made in automatic speech recognition (ASR), and now complex and sophisticated systems are being deployed in successful applications. In general, the complexity of a recognition task depends on many conditions. Recognition of words in isolation is a much easier task than continuous speech recognition. A recognizer that works for any speaker is harder to build than one that is tuned to a particular speaker. Read or dictated speech can be recognized more accurately than conversational

¹Electronic Numerical Integrator and Computer, invented by John W. Mauchly and John P. Eckert, 1946.

speech. Early ASR systems required each user to train the system to their voice before use, and they had to speak with a pause ... between ... words. These discrete isolated-word speaker-dependent systems were superseded by more complex speaker-independent continuous speech recognition systems, and the vocabulary of the system also increased many-fold. Statistical pattern recognition has emerged as a successful framework to recognize speech. The acoustic observations are modeled as random variables associated with symbols such as phones and words, thus attempting to capture the variability inherent in speech. These developments have enabled a large number of applications ranging from telephone operated services such as directory assistance, airline reservations, and credit card services; professional dictation systems for lawyers and doctors; and aids for the handicapped.

A number of challenges still need to be tackled to allow widespread use of ASR technology. A system may recognize certain types of speech with high accuracy and perform badly on others, even when humans are able to understand both of them with nearly equal ease. This gap in performance is well illustrated by the results reported on the 1998 DARPA Broadcast News benchmark tests by the best research systems. The word error rates on the spontaneous speech portion of the test set (14-16%) were nearly double those on the baseline condition of planned recordings (8-9%) [100]. In this test the baseline or "F0" condition is defined as planned speech from a native speaker, over a high bandwidth channel, with no background noise. Further, those systems that also participated in 2000 DARPA Conversational Speech benchmark test, performed at error rates of roughly 30%. The degradation in performance may be due to many factors such as channel effects, variability in speaking rate and dialect of speakers, less careful pronunciation, loosely structured language, and the presence of disfluencies. For deploying ASR systems for a wide range of applications, including voice-mail transcription and speech translation, it is necessary to recognize conversational speech more accurately.

Though many factors play a role, the poor performance of current ASR systems in recognizing conversational speech is arguably due to their inability to model large acoustic variation effectively. To tease apart the contribution of acoustic variation, a study was

conducted in 1996 where a system, similar to the current systems, was used to recognize the same word sequence spoken in three different styles [124]. Spontaneous speech was recorded, and then the transcript of the same speech was read and “acted spontaneously” by the same speakers. While the system recognized spontaneous speech with a word error rate of 52.6%, the acted and the read versions were recognized at 37.4% and 28.8% error rates respectively. The differences in error rate must be due to differences in the acoustics related to speaking styles since the transcript was the same in all the three tests. Further experiments on this corpus (Multi_reg corpus) showed that the degradation with increasingly casual speaking style occurs across telephone-band and wide-band speech and under matched training and test conditions [115]. To bridge this gap in performance, the acoustic variation related to the speaking style needs to be modeled better in large vocabulary speaker-independent conversational speech recognizers, and that is the specific goal of this thesis.

In many ASR systems, the acoustic variation of words is modeled at two levels - the pronunciation model which maps a dictionary phoneme sequence (base form) to the realized phone sequence (surface form), and the acoustic model which maps the surface form to observed spectral variation in terms of multivariate distributions. To study the role of these two components, Keating compared a sample from a conversational speech corpus (Switchboard) with that of a read speech corpus (TIMIT). Both samples were previously hand-transcribed [71]. She found that the average number of surface (phonemic) forms per word in fluent speech was twice that of read speech. Similarly, the number of phones (using the TIMIT symbol inventory) for each phoneme varies twice as much. While the phonemic variation of a word may be dealt with using pronunciation models, the increased phonetic variation is probably better addressed by improving current acoustic models. Work with simulated data which was produced using an acoustic model of speech, has also pointed to pronunciation variability as one of the main problems in recognizing conversational speech [88]. So far, the work on pronunciation modeling in terms of phoneme-level substitutions, deletions and insertions has yielded small performance gains [17, 113]. Further gain could come from improvements in acoustic models, and recent research bears this out.

Saraclar et al. [115] showed that modeling pronunciation beyond the phonemic level, and allowing changes in acoustic models, brings additional benefits. Experiments by Hain and Woodland also demonstrate an advantage in using phonetic context to directly influence the temporal nature of acoustic models [54]. These studies support the notion that there is a need to represent variation of a more gradient nature, and we hypothesize that, in a similar manner, high-level context (beyond phonemic) influences the acoustic models of the phonemes as well as the pronunciation of a word.

1.1 Modeling Contextual Variation with Syllable and Word Features

Conventionally, acoustic variation of a phoneme has been captured by conditioning the acoustic models on the identity of the neighboring phonemes. Typically, in large vocabulary ASR, phonemes with immediate left- and right-neighbors (triphones) and possibly two neighbors (pentaphones) are used. Conditioning only on phonemic context does not capture the acoustic variation of conversational speech fully. In recent years, augmenting the context with position of phoneme in the word has brought additional improvements to ASR performance, and it is now widely used (e.g. [55]). This is consistent with observations in linguistic studies about word-position effects on different consonants, using electropalatography (EPG) [73]. The linguopalatal (tongue-palate) contact, which affect the strength and duration of sound produced, for word-initial consonants is significantly different from word-final. In this work, we look at conditioning acoustic models on high-level contexts, specifically syllable and word structure.

Our hypothesis is that the syllable structure is also useful in modeling the variation in conversational speech not accounted for by phoneme context. Consider the phoneme “t” (in the context “iy t er”) in “beater”, “beat Ernest” and “return”. Even though it is the same triphone, the articulation of the phone “t” in the three contexts is distinctly different - in the first it is flapped, in the second it is an unreleased closure and in the third it is a closure plus a release. These differences are closely related to syllable structure, and correspond to ambisyllabic, syllable-final, and syllable-initial contexts, respectively. For read speech,

the variation in acoustic realization of “t” has been studied using statistical techniques by Randolph [109]. Realizations of “t” in TIMIT, a corpus of read speech, were hand-labeled as released, unreleased, flapped, deleted or glottalized. Using mutual information, he found that these realizations were explained by syllable position better than the phoneme context, and the context of features such as the place of articulation, the manner of articulation, the voicing or the stress. Further, using decision trees trained with these attributes, he obtained classification accuracies of about 84%, and an entropy reduction of about 56%. These patterns are likely to be present in conversational speech, and syllable structure may explain different realizations in other phonemes.

Often word beginning and ending rules can be generalized to the syllable level and this is used to explain why certain phone sequences do not occur in English [57]. For example, extending the notion that “ng” (as in rang) and “ŷ” (as in beige) do not occur as word onsets explains the absence of word medial clusters of the form “VCngV” or “VCCngV” and so on. If word-internal (syllable-level) phonotactics mirror word-level phonotactics, then incorporating syllable-level structure into the recognition system should result in gains beyond those seen in incorporating word-level structure. However, since there are some additional phonological processes that target syllable edges (see [20] for a discussion), information about position-in-word as well as position-in-syllable is useful.

Statistical studies analyzing a conversational speech corpus (Switchboard) show systematic effects associated with components of a syllable [49, 38, 51]. In particular, they found the syllable onsets to be more stable than codas, and an analysis of errors made by state-of-the-art systems on recognizing conversational speech suggests that accurate recognition of syllable onsets is more important for word recognition than of syllable codas [50].

The focus of our work is on learning contextual variation directly in the acoustic model using both word- and syllable-level information, since they seemed promising in both previous pronunciation models and acoustic studies mentioned above. In contrast to modeling the syllable explicitly as a unit, we use a tree-based clustering mechanism to allow sharing of parameters across all contexts for robust estimation. To tackle the problems that arise

in using a large number of contextual features, which could potentially include prosody and cues about sentence structure, we have extended the decision tree based clustering to use multiple stages of clustering. Previous work using syllables can be criticized for ignoring the effects of re-syllabification. We investigated the effects of re-syllabification on recognition performance, finding no significant impact on recognition performance. In this thesis, the term syllable- and word-level features refers to attributes of syllable and word structure, and it does not represent features from the input speech.

1.2 Modeling Temporal Variation with Flexible Topology

Another limitation of current systems is the adherence to a fixed model structure for all phones irrespective of their phonetic contexts. In certain systems, a limited amount of temporal variations is captured with transitions that allow certain states of the model to be skipped.

One of the obvious outcomes of using a fixed topology is the imposition of an unrealistic constraint on the duration of the phones. Popular topologies impose a 30 millisecond constraint on the minimum duration of a phone. This happens because the phonemes are modeled by at least three states (allowing fewer states for all phonemes tends to cause insertion of spurious phones), and each state needs at least one observation frame (each frame is about 10 millisecond). However, examination of the duration of phones in conversational speech shows that about one in ten phone instances is shorter than 30 ms. Figure 1.1 shows a plot of duration distribution computed from three and a half hours of conversational speech that was hand-labeled. The data was labeled at syllable-level in the International Computer Science Institute (this subset will be referred to as ICSI transcription). The phone durations were approximated by distributing the syllable duration evenly amongst its constituent phones, thus underestimating the number of phones that are shorter than 30 ms. A fixed topology may not capture the temporal variations associated with either end of this distribution.

Temporal variation of a phoneme also dependent on its context. A short flap, “dx” as

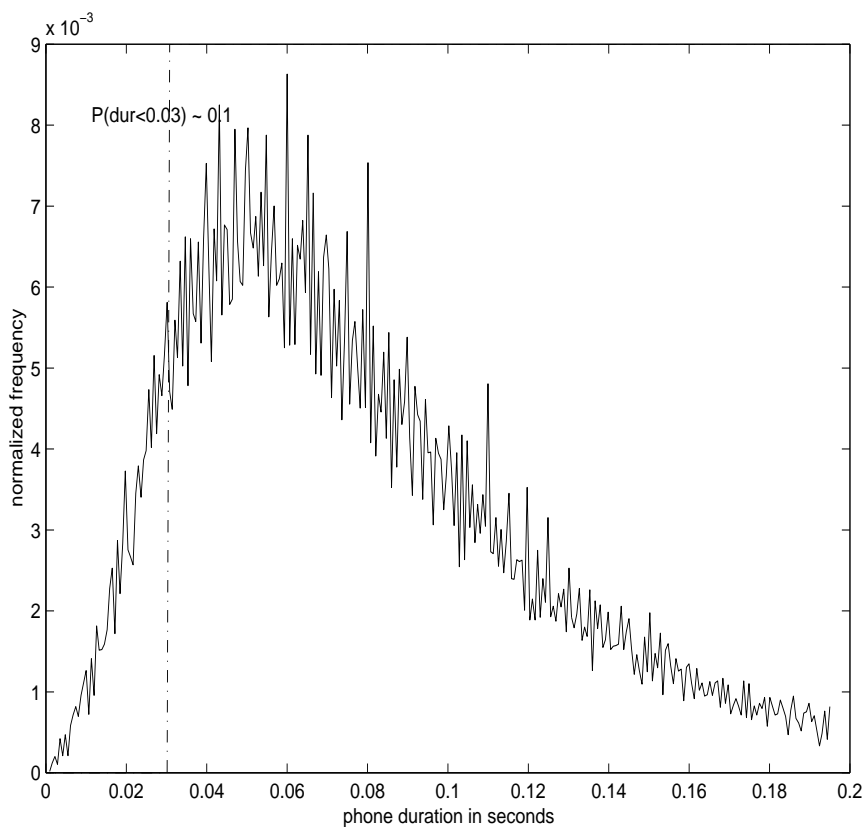


Figure 1.1: Estimated distribution of phone duration in conversational speech (tail omitted for better display of mode).

in Peter (p-iy-dx-axr) has less temporal variation than a diphthong “oy” as in avoid (ax-v-oy-d). Use of more observation distributions than necessary may lead to poor estimation of parameters, while the use of fewer may lead to less discriminatory models.

A few researchers have developed methods for learning context-dependent HMM topology [82, 121, 99, 31, 3, 53, 54, 33], most of which cannot be directly used for a large vocabulary speech recognizer for reasons described in Chapter 6.

We investigate a different approach with the idea that high level structure can be used to design HMM topologies for phonemes in a context specific manner. The general problem of designing topologies is difficult to solve. Using a few reasonable assumptions, we simplify the task, and outline an algorithm to learn them automatically. The algorithm was applied

on a large conversational speech recognition task, different settings were explored, and their impact was assessed.

1.3 Outline of the Thesis

To set the background for subsequent discussion, and to explain terminology and techniques used in our experiments, the fundamentals of automatic speech recognition systems are briefly explained in Chapter 2. There are a number of parameters that go into fine tuning a speech recognition system, not all of which can be explored in the span of a thesis. Chapter 3 explains the experimental paradigm that we adopted for this work, including the training corpus and the recognition task. The three subsequent chapters deal with the core of this thesis, and gradually develop a sub-word unit that is specific to its wider context.

In Chapter 4, we improve the current acoustic models, which have a fixed topology, by incorporating syllable- and word-level features in their state observation densities. The chapter begins with the description of related work, and enumerates the attributes of syllable and word structure that we have used in this thesis. Then, we delve into our investigation of their impact on state observation densities, including results from training and testing on a standard large vocabulary spontaneous speech recognition task. We compare the gains across different systems, and examine the automatically learned decision trees to assess the usefulness of each feature. Further, we investigate the effects of re-syllabification on the recognition performance.

The encouraging results from Chapter 4 opens the possibility of using a variety of other high-level features in future. However, a few technical difficulties need to be tackled before this can be achieved. In Chapter 5 we develop a framework to overcome these difficulties. We test our algorithm on a large task of clustering pentaphones, and verify that the state observation densities thus estimated give similar performance on a recognition task.

In Chapter 6, we pursue our aim of improving sub-word units by designing context-specific model topologies for phonemes. After a brief review of previous work, we develop a systematic framework for mapping context-dependent phonemes to HMM topologies. The

models from Chapter 4 with improved state observation densities are used as the baseline. The fixed topology is replaced by a context-sensitive topology, and then the models are re-estimated. These models are tested and their impact is analyzed on a conversational speech task.

Finally, the conclusions from and the contributions of this work, and possible future directions are summarized in Chapter 7.

Chapter 2

AN AUTOMATIC SPEECH RECOGNITION SYSTEM

An automatic speech recognition (ASR) system attempts to find a sequence of words corresponding to original utterance. A series of complex functions need to be performed to achieve this. This chapter describes the fundamental principle behind various components of such a system, explains the terminology, and provides references for many of the sub-tasks.

Currently, statistical pattern recognition is the most popular framework for recognizing speech and so most of our discussion pertains to it. Alternative paradigms using artificial neural networks [81] also exist. A hybrid strategy effectively combines the use of techniques from both these areas [90, 13, 111].

The general framework for statistical ASR system can be schematically represented by Figure 2.1 (adapted from [129]). In any ASR system, the speech signal is first converted to a sequence of vectors, $x_{1:T} = [x_1, x_2, \dots, x_T]$, as described in Section 2.1. The core task then is to compute the word sequence $w_{1:M} = [w_1, w_2, \dots, w_M]$ that minimizes a cost function. In statistical pattern recognition problems, the problem is cast in terms of a probability distribution over random variables, $X_{1:T}$ and $W_{1:M}$ here, and a solution is obtained using the Bayes decision rule for minimum risk. Making the assumption that all classification errors (incorrect sentences here) are equally bad or costly, an optimal decision rule (maximum *a posteriori* or MAP rule) is obtained as:

$$\hat{w}_{1:M} = \operatorname{argmax}_{w_{1:M}} P(w_{1:M}|x_{1:T}) \quad (2.1)$$

$$= \operatorname{argmax}_{w_{1:M}} \frac{P(x_{1:T}|w_{1:M})P(w_{1:M})}{P(x_{1:T})} \quad (2.2)$$

$$= \operatorname{argmax}_{w_{1:M}} P(x_{1:T}|w_{1:M})P(w_{1:M}) \quad (2.3)$$

Since $P(w_{1:M}|x_{1:T})$ cannot be modeled easily, the MAP rule is applied to estimated proba-

bilities $P(x_{1:T}|w_{1:M})$ and $P(w_{1:M})$, as if they were true probabilities.

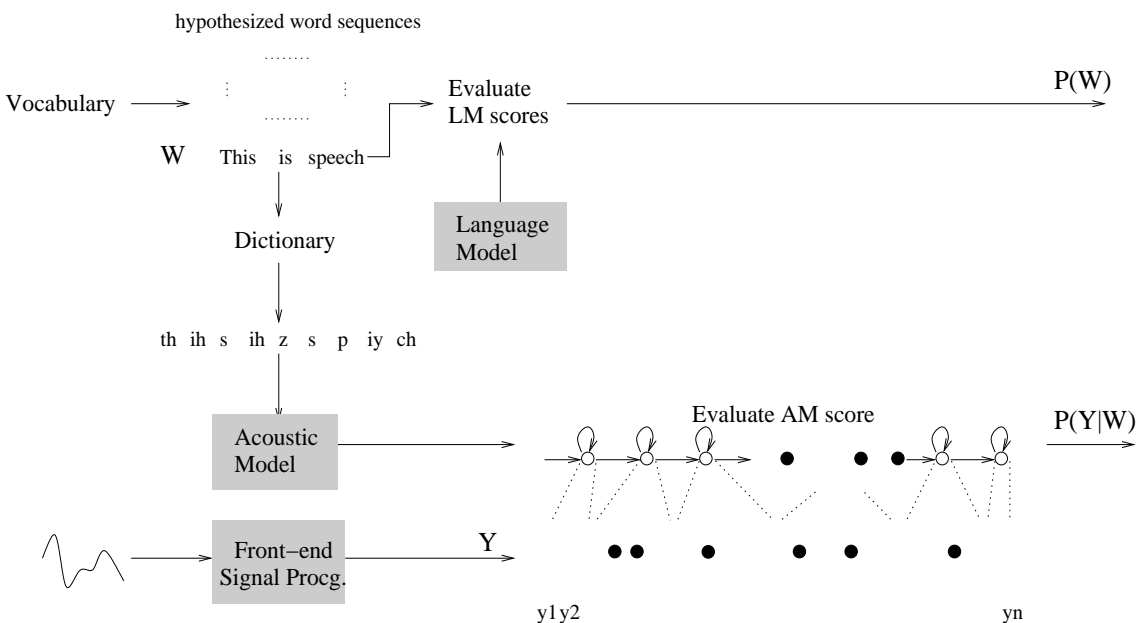


Figure 2.1: Overview of an ASR system based on a statistical framework.

The component $P(w_{1:M})$ encodes the prior knowledge about the likelihood of observing a word sequence in the language of interest, and is evaluated by a **language model**. Section 2.2 briefly describes the structure and estimation of a typical language model.

Given a hypothesized word sequence, an **acoustic model** evaluates how well it matches the input speech. As much of the proposed work will be in the area of acoustic modeling, we will examine this topic in further detail. Since separate acoustic models cannot be trained for every word sequence in a large vocabulary system, the task is typically broken down into modeling smaller sub-word units which are then strung together to produce a model for the word sequence. The models for the **sub-word units** are chosen in a manner that enables estimation of their parameters automatically from representative training data. Hence, there are two main issues in acoustic modeling: choice of an elementary model which includes issues related to estimating their parameters and using them to decode a symbol sequence; and composing an acoustic model for a hypothesized word sequence. These are

described in Section 2.3 and Section 2.4 respectively.

The implementation of Equation 2.3 is referred to as the search algorithm. A few approximations and a number of computational tricks are used to implement the search over the huge space of all possible word sequences at a computational cost of $O(T)$, which are briefly outlined in Section 2.5.

2.1 Feature Extraction

For accurate word recognition, extraneous variation in the input to the ASR system needs to be minimized. These variation may be due to differences in pitch between speakers, differences in length of the vocal tract across speakers or the nature of the channel through which speech is delivered to the ASR system. A series of operations are performed on speech to minimize the impact of these variations.

Voiced speech can be modeled as the output of a linear filter $h(n)$ driven by periodic glottal pulses $g(n)$. The characteristics of this filter are related to the shape of the vocal tract and vary less across examples of units of speech, while the glottal component varies widely with pitch. By transforming the input speech signal as shown below it is possible to separate the contributions of the two components, assuming non-zero spectrum.

$$\begin{aligned}
 FT(h(n) * g(n)) &= H(\Omega)G(\Omega) \\
 \log(|FT(h(n) * g(n))|) &= \log(|H(\Omega)|) + \log(|G(\Omega)|) \\
 IFT(\log(|FT(h(n) * g(n))|)) &= IFT(\log(|H(\Omega)|)) + IFT(\log(|G(\Omega)|)) \quad (2.4)
 \end{aligned}$$

Here, FT and IFT denote the Fourier transform and its inverse [28].

The glottal pulse has harmonics related to the pitch of the speaker and appears much higher in the cepstral domain compared to the response of the vocal tract. By computing only the lower (12-14) order **cepstral coefficients**, the variation due to pitch is largely left out while retaining most of the essential information for word recognition. Before processing speech, short segments are extracted using overlapping windows (such as Hamming window 20-30 ms).

Motivated by the robustness of human auditory system, certain transformations are applied in the frequency domain before computing the cepstral coefficients. In most systems, the frequencies are warped to resemble human auditory processing, using scales such as Mel-scale - almost constant Q-filters. Additionally, certain systems modify the frequency response to mimic the human response to loudness, and suppress constant or slowly varying components in each frequency channel using non-linear operations [59].

The effect of a linear channel appears as an additive term in the cepstral domain, analogous to the glottal source component shown in Equation 2.4. To reduce the effect of slowly-varying channels such as telecommunications media, many systems normalize the feature vector by subtracting a long term cepstral average.

Depending on the general shape and the length of a vocal tract, the spectral characteristics of similar phonetic sounds generated by different individuals may differ. Vocal tract length normalization is used to map the general frequency characteristics of a speaker to a normalized representation. The parameters of the warp are estimated from the acoustics of the speaker by maximizing likelihood of the warped speech using a Gaussian mixture model to characterize each warp [123, 106, 79].

After applying these techniques the speech is represented typically in the form of 12-14 cepstral coefficients (Mel-frequency cepstral coefficients, MFCC or Perceptual Linear Predictive Coding, PLP). These cepstral features, their derivatives and the log of local energy are concatenated together to form a cepstral vector. A sequence of these vectors form the input to the recognition system at a rate of about 100 per second.

2.2 Language Model

The language model evaluates how likely a word sequence is in the language of interest. Since the space of sequences is infinite, the likelihood of a random word sequence $w_{1:M}$ is computed as a product of local probabilities over equivalent neighborhood or history, h_i .

$$P(w_{1:M}) = P(w_1) \prod_{i=2}^M P(w_i | w_{1:i-1}) \quad (2.5)$$

$$= P(w_1) \prod_{i=2}^M P(w_i | h_i) \tag{2.6}$$

In the most popular approach $h_i \approx [w_{i-1} \ w_{i-2}]$, resulting in a 2nd-order Markov model which is often called **trigram**.

Straightforward use of trigram probabilities creates some difficulties. Being categorical variables these probabilities are discrete and are usually estimated from their frequency in training data. For a system with vocabulary $|V|$, this would mean counting $|V|^3$ types of tokens. In a finite amount of training data, valid three word sequence that are less probable may not occur. At the same time, setting those probabilities to zero in a product could make it impossible to search over non-zero but less probable word sequences. Techniques such as discounting, back-off or interpolation are used to avoid assigning zero probability to such rarely observed trigrams [18, 63, 93]. Clarkson and Rosenfeld’s statistical language modeling toolkit incorporates a suite of these techniques [23].

2.3 Elementary Models

The most popular approach to model sub-word units uses the **Hidden Markov model (HMM)** [108]. This model derives its power from its simplicity, the idea of capturing a random process with the simplest memory, a first order Markov dependence. The parameters of this model include a set of discrete scalar states $\{q_j; j = 1 : N\}$, a conditional **observation density** given a state $\{b_j(x) = P(x|q_j); x \in \mathcal{R}^d, j = 1 : N\}$, state transition probabilities $\{a_{ij} = P(q_j|q_i); i, j = 1 : N\}$, and initial probabilities $\{\pi_j = P(q_j); j = 1 : N\}$.

Various results needed to use HMMs are easier to see by defining HMMs in terms of their conditional independence properties. Using the notations from [9], if $x_{1:T}$ is sequence of observation and $Q_{1:T}$ is an associated hidden random state sequence with values in $\{q_j\}$, then

$$X_t \perp\!\!\!\perp \{Q_{i \neq t} X_{i \neq t}\} | Q_t \tag{2.7}$$

$$\{X_{1:t-1} Q_{1:t-2}\} \perp\!\!\!\perp \{X_{t:T} Q_{t:T}\} | Q_{t-1} \tag{2.8}$$

where \perp and $|$ denote independence and conditioning, respectively. The first property (2.7) states that the observations at time t is independent of everything else given the state at that time, Q_t (i.e., conditional independence of observation given state). The second property (2.8) implies that the state sequence is first order Markov.

Together, the independence properties make it possible to compute global quantities such as $P(x_{1:T}, q_{1:T})$ and $P(x_{1:T})$ in terms of local distributions, as can be seen from the following equations.

$$P(x_{1:T}, q_{1:T}) = P(x_1, q_1) \prod_{t=2}^T P(x_t, q_t | x_{1:t-1}, q_{1:t-1}) \quad (2.9)$$

$$= P(x_1 | q_1) P(q_1) \prod_{t=2}^T P(x_t | x_{1:t-1}, q_{1:t}) P(q_t | x_{1:t-1}, q_{1:t-1}) \quad (2.10)$$

$$= P(x_1 | q_1) P(q_1) \prod_{t=2}^T P(x_t | q_t) P(q_t | q_{t-1}) \quad (2.11)$$

$$\begin{aligned} P(x_{1:T}) &= \sum_{q_{1:T}} P(x_{1:T}, q_{1:T}) \\ &= \sum_{q_{1:T}} P(q_1) P(x_1 | q_1) \prod_{t=2}^T P(x_t | q_t) P(q_t | q_{t-1}) \end{aligned} \quad (2.12)$$

Equations 2.9 and Equation 2.10 result from merely applying the chain rule. Equation 2.11 is obtained using corollaries from HMM conditional independence assumptions Equation 2.7, and Equation 2.8 (namely, $X_t \perp \{X_{1:t-1}, Q_{1:t-1}\} | Q_t$ and $Q_t \perp \{X_{1:t-1}, Q_{1:t-2}\} | Q_{t-1}$). Further, the two independence properties enable estimation of the parameters of HMM through *alpha*- and *beta*-recursions, which are described later in Section 2.3.1. By incorporating cepstral derivatives in the feature vector, the conditional independence of observation X_t given the state Q_t is violated. However, in speech recognition with cepstral vectors at typical frame rates, the conditional independence assumption is violated even without the derivatives, due to overlapping windows and the slow variation of many sounds in speech relative to the frame rate. Incorporating derivatives provide an inexpensive alternative to representing dependence that works well in practice.

A number of variants and enhancements have been developed for HMMs, of which Factorial HMMs, Buried Markov Models (BMM) and Coupled HMMs are more recent. Factorial

HMMs aim to model observations that are produced from the interaction of loosely coupled processes [42, 94]. It allows parallel state sequences with independent transition probabilities to jointly determine the observation probability (e.g., in the form of linear combination of their means). Buried Markov Models add non-linear dependencies to the observation density from the neighboring feature vectors [8]. A cost function based on mutual information is used to select features automatically from training data in a manner that encourages discrimination between states. In Coupled HMMs, each observation stream is modeled by a hidden state sequence and the state transitions are coupled between all the state sequences [116]. All these models can also be viewed as different instances of a class of models called graphical models, which are being explored as another framework for ASR [10, 135].

Another alternative acoustic model is the Stochastic Segment Model or trajectory model (SSM) [43, 98]. While each hidden state in the Markov chain of an HMM describes only one observation, a hidden state in a segment model describes a sequence of observations. In a recent variation on this theme, called a Markov Process on a Curve (MPC), the posterior probability of a segmentation is directly modeled [117]. The probability of remaining in a state decays exponentially with the length of the arc traversed in the observation space of the multivariate random process, where the arc length is measured using a state-dependent metric.

Among all these models, HMMs with continuous observation densities continue to be the most widely used model for large vocabulary ASR task, and so we will use HMMs in this thesis.

2.3.1 HMM Parameter Estimation

In ASR, it is important to learn the parameters of the model automatically from a large collection of speech which is transcribed at the word level. To learn all the parameters of an HMM automatically is a difficult problem. However, given the structure of an HMM in terms of the number of states and the transitions between them, it is possible to learn the parameters using a few different techniques.

Ideally, the parameters should be trained to minimize classification error. However, it is hard to minimize the word error criterion [66]. Another cost function that could indirectly minimize error is the maximum mutual information estimation (MMIE) criterion. Following a result from Baum-Eagon [6], Gopalkrishnan *et al.* came up with a closed form expression for training discrete HMM using MMIE criterion [45], which was extended to the continuous case by Normandin [95]. This has been further developed in [122, 118, 104], and still remains computationally expensive.

The maximum likelihood criterion is an alternative approach that is popular mainly due to its mathematical tractability and computational simplicity. It is also optimal when the form of the model is correct. Although there is no closed form solution to estimate HMM parameters $\theta = \{a_{ij}, b_j(x), \pi_j \mid i, j = 1 : N\}$ from a training set \mathcal{X} , we can choose an initial θ and iteratively improve the estimate of θ using the Baum-Welch algorithm (which is an instance of Expectation-Maximization or **EM algorithm**) [108, 7, 29]. The EM algorithm iteratively maximizes a particular cost function $L(\theta|\theta_{old})$ which guarantees to increase the likelihood of the data given a model $P(x|\theta)$. The cost function $L(\theta|\theta_{old})$ is defined in terms of a hidden random variable Y .

$$L(\theta|\theta_{old}) = E[\log P(x, Y|\theta)|x, \theta_{old}] \quad (2.13)$$

$$\begin{aligned} E[\log P(x, Y|\theta)|x, \theta_{old}] &= \sum_y P(y|x, \theta_{old}) \log P(x, y|\theta) \\ L(\theta_{new}|\theta_{old}) \geq L(\theta_{old}|\theta_{old}) &\implies P(x|\theta_{new}) \geq P(x|\theta_{old}) \end{aligned} \quad (2.14)$$

The unobserved random variable Y is carefully chosen for each application so that the computation of $L(\theta|\theta_{old})$ is simplified. At each iteration, a new set of parameters θ are computed that maximizes $L(\theta|\theta_{old})$. In general, the iterations converge to a local maxima. EM has also been described in other minimization frameworks, based on which a few variants have also been developed (e.g., [26, 16]).

In the case of an HMM, the computation of $L(\theta|\theta_{old})$ is simplified by considering the hidden state sequence $Q_{1:T}$ as the variable Y in Equation 2.13. Then, using the independence assumptions of HMM (2.7, 2.8), the expected joint log probability over hid-

den sequence $E[\log P(x_{1:T}, q_{1:T}|x_{1:T}, \theta_{old})]$ can be computed in terms of $P(q_t|x_{1:T}, \theta_{old})$ and $P(q_{t-1}, q_t|x_{1:T}, \theta_{old})$ through two recursions, since

$$\begin{aligned}
E[\log P(x_{1:T}, q_{1:T}|\theta)|x_{1:T}, \theta_{old}] &= \sum_{q_{1:T} \in Q_{1:T}} P(q_{1:T}|x_{1:T}, \theta_{old}) \log P(x_{1:T}, q_{1:T}|\theta) \quad (2.15) \\
&= \sum_j P(q_1 = j|x_{1:T}, \theta_{old}) \log \pi_j \\
&+ \sum_t \sum_j \sum_k P(q_{t-1} = j, q_t = k|x_{1:T}, \theta_{old}) \log a_{jk} \\
&+ \sum_t \sum_j P(q_t = j|x_{1:T}, \theta_{old}) \log b_j(x_t) \quad (2.16)
\end{aligned}$$

using Equation 2.11 to rewrite the expectation in terms of the HMM parameters $b_j(x)$, a_{ij} and π_j of θ_{old} . The posterior probability of being in a state $P(q_t = j|x_{1:T}, \theta_{old})$ and the transition $P(q_{t-1} = j, q_t = k|x_{1:T}, \theta_{old})$ can be computed efficiently using forward-backward algorithm as shown below.

$$\gamma_j(t) \triangleq P(q_t = j|x_{1:T}, \theta_{old}) = \frac{P(q_t = j, x_{1:T}|\theta_{old})}{\sum_j P(q_t = j, x_{1:T}|\theta_{old})} \quad (2.17)$$

$$P(q_t = j, x_{1:T}|\theta_{old}) = P(q_t = j, x_{1:t}|\theta_{old})P(x_{t+1:T}|q_t = j, \theta_{old}) \quad (2.18)$$

So, the key quantities to compute the expectation step in the EM algorithm are:

$$\alpha_i(t) \triangleq P(q_t = i, x_{1:t}|\theta_{old}) = b_i(t) \sum_{j=1}^N \alpha_j(t-1) a_{ji} \quad (2.19)$$

$$\beta_i(t) \triangleq P(x_{t+1:T}|q_t = i, \theta_{old}) = \sum_{j=1}^N \beta_j(t+1) b_j(t+1) a_{ij} \quad (2.20)$$

$$\gamma_t(i) = \frac{\alpha_i(t) \beta_i(t)}{\sum_j \alpha_j(t) \beta_j(t)} \quad (2.21)$$

$$\xi_{ij}(t) \triangleq P(q_{t-1} = i, q_t = j|x_{1:T}, \theta_{old}) = \frac{\gamma_j(t) a_{ij} b_j(t+1) \beta_j(t+1)}{\beta_i(t)} \quad (2.22)$$

The *alpha*- and *beta*- recursions (Equation 2.19, 2.20) are computed in the forward and the backward directions (initialized with $\alpha_i(1) = \beta_i(T) = 1$), respectively, at a computational cost of $O(N^2T)$. In each iteration, the HMM parameters are updated using sufficient statistics gathered in that iteration.

A few cautions have to be observed in the actual implementation of these steps. Since the likelihood under a Gaussian density decreases exponentially with increasing distance from its mean, the value of $P(x_t|q_i)$ will often be very small, and the product of the terms in the recursions will be still smaller. To keep $\alpha_j(t)$ and $\beta_j(t)$ within the dynamic range of a machine representation, they need to be normalized as the recursion progresses in time, and the recursion is frequently carried out in the log domain.

When observation densities are modeled by a mixture of Gaussians, the mixture weights will also be unobserved variables. When the number of components are given, an analytical expression can be derived to estimate the new value of the parameters θ . Usually, the components are gradually increased till the desired number is achieved. A simplification of this algorithm replaces the sum over all state sequences in Equation 2.15 by the most likely state sequence $P(x_{1:T}, q_{1:T}|\theta)$ and is called **segmental K-means** or **Viterbi-style estimation** for its resemblance to the k-means and Viterbi algorithms, respectively [67].

2.3.2 Optimal HMM State Sequence

Given an observation sequence, the most likely symbol sequence is usually obtained from the most likely hidden state sequence.

$$\operatorname{argmax}_{q_{1:T}} P(q_{1:T}|x_{1:T})$$

The **Viterbi algorithm**, an algorithm based on dynamic programming, computes this in a highly efficient manner [108]. The HMM independence assumptions (Equation 2.7 and Equation 2.8) make it possible to compute the globally optimal path by extending locally optimal paths. The search uses an optimization rule, also known as Bellman's principle of optimality, which is best illustrated with a simple example. If the shortest Seattle-Portland route passes through Tacoma and the shortest Vancouver-Portland route passes through Seattle, then the shortest Vancouver-Portland necessarily has to pass through Tacoma. The search could be carried out in a breadth-first (also known as time synchronous manner) [92] or in a depth-first manner (e.g., [101]). Using the Viterbi algorithm in a time synchronous

search, this would mean extending the set of best paths frame by frame, each best path is a path ending in a particular allowable state at that frame. At each frame, the best path ending in a state could only come from the extension of N previous best paths, thus incurring only $O(N^2)$ computation for updating the best paths per frame. The memory used is $O(N^2T)$ which can be traded-off for computation and reduced to $O(N^2 \log T)$ [134].

These elementary units are strung together, and the most likely state sequence is computed over a huge state space in a large vocabulary decoder as outlined briefly in Section 2.5.

2.4 Acoustic Model for Word Sequences

In a large vocabulary ASR systems, how a word sequence is modeled also depends on the lexicon, the sub-word unit chosen, and the process of mapping the units to an elementary model such as HMM.

The vocabulary of a speech recognition system is defined by a **lexicon** (dictionary), with which the ASR system hypothesizes word sequences. The lexicon provides a mapping from a word to a set of predefined units, usually phonemes. Lexical information such as position of syllable in a word, position of phone in a syllable, position of phone in a word, and lexical stress can be encoded in the dictionary, an example of which is described further in Section 4.2. The mapping in the lexicon is often defined manually by experts, although sometimes text-to-phoneme rules are used. The recognizer uses the lexicon to map the word sequence into a sequence of lexical units. This sequence is sometimes modified (to phones) by a **pronunciation model** that takes into account the context of the neighboring words [112, 17, 113].

Acoustic manifestation of phones vary significantly with context, and the variants of a phone are called allophones. A typical ASR system takes this into account by modeling words as a sequence of context sensitive phones. **Triphones** are popular sub-word units, where each triphone is defined as a phone with a unique pair of left and right neighbors. For example, suppose the notation $x-y+z$ represents a phone y occurring after an x and before a z . The phrase “speak out” would be represented by the sequence, “ $\#-s+p$ $s-p+iy$ $p-iy+k$ ”

$iy-k+aw \ k-aw+t \ aw-t+\#\prime$, where $\#$ denotes utterance boundaries. Note that the triphone contexts can span word boundaries. In some contemporary systems, the context includes two pairs of left and right neighbors, to form units called **quinphones** or **pentaphones**. Additionally, some systems also use word boundaries as a conditioning factor. A typical ASR system for English uses about 46-50 phonemes and the number of triphones and pentaphones encountered in a sample of 60 hours of speech may be as many as 120K and 2.3M respectively.

It is difficult to estimate distinct models for each acoustic unit. Triphones, for example, are large in number and a few units cover a disproportionate amount of data, leaving a majority with too few examples to train their parameters from. A widely accepted solution to this problem is to assume a fixed left-to-right model structure for all sub-word units (e.g., a 5 state model as in Figure 2.2) and allow **parameter tying**, where some parameters of the observation density are shared across states of different sub-word units. As mentioned earlier, the conditional probability of the observation given a state $P(x_t|q_t = j)$ for a multivariate Gaussian distribution with large dimension decreases rapidly with increasing distance from the mean. Consequently, while decoding the best path, these terms play a much greater role than the transition probabilities. In some implementations, the probabilities corresponding to allowed and disallowed path are set to one and zero, respectively.

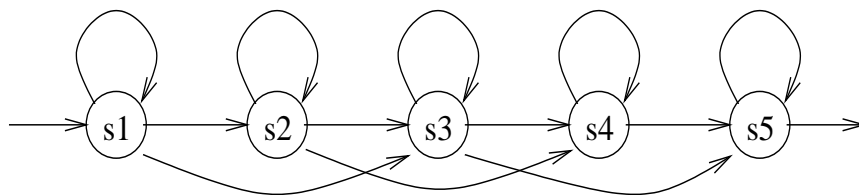


Figure 2.2: An example left-to-right HMM used to model a sub-word unit

Parameters can be tied at a state-level heuristically or can be determined automatically using the linguistic class of the associated labels in decision trees, as described shortly. In some work strictly distance-based techniques are used (e.g., k-means clustering), but they

have difficulty in handling labels that are infrequent or are unobserved in the training data. Parameters can also be tied for the covariance matrix of the distribution using algorithms such as semi-tied covariance [40, 11, 46, 41].

Decision trees have proved to be useful in classification, description and generalization of data in a variety of applications (*e.g.* [91, 14, 15]). Decision trees were first used in ASR for tying HMMs where state labels were approximated using a Poisson process [5]. It became popular when they were used to cluster observation densities of a state in a maximum likelihood framework [70, 130].

Decision tree clustering is particularly attractive for ASR, since it provides a mechanism to map those sub-word units that are not seen in the training data to a model which may closely resemble it. For ease of description let us assume the sub-word units to be triphones. To train a tree, all the triphones observed in the training data are pooled at the root node of the tree. A set of predefined questions, typically about the phonetic context of the triphones, is used to define candidate partitions (typically binary partitions) of a node in the tree. The likelihood of each partition is measured in terms of how well the training data in it can be modeled by two distributions. The question corresponding to the partition that maximizes the likelihood of the data in a node is chosen as a candidate for the next split. From amongst these candidates, the node with the best likelihood gain is split. The best partitions of the new clusters resulting from this split are added to the list of the candidate splits, and thus the tree is grown until some stopping criterion is met (*e.g.*, limit on the number of leaves or terminal nodes).

By making two assumptions, the decision tree can be used to train HMM observation densities. The observation densities corresponding to the states of the HMM are assumed to be adequately modeled by multivariate single Gaussian distributions. Likelihood evaluated by a single Gaussian on the same data on which it was estimated is only a function of its estimated covariance and the number of training samples [1]. So, the Gaussian assumption makes it possible to compute likelihood in terms of Gaussian sufficient statistics. (After the tree is grown, sometimes more complex models such as Gaussian mixtures are associated

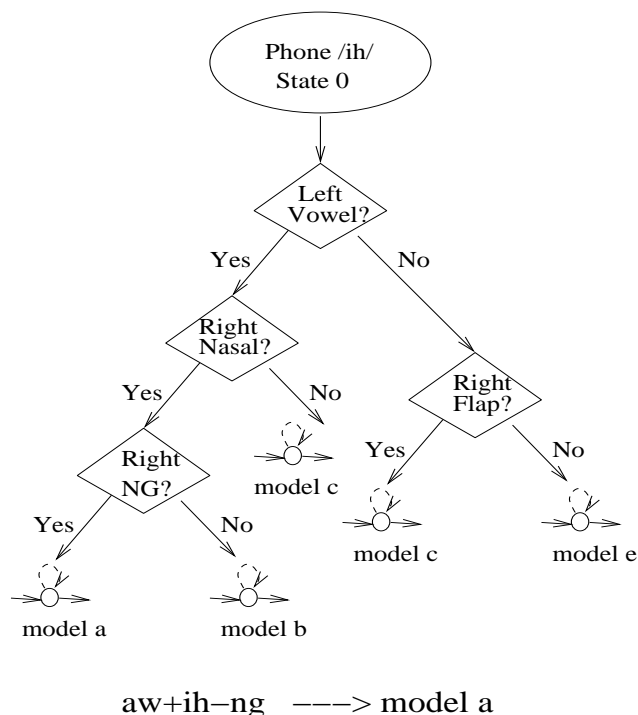


Figure 2.3: Mapping a state of a sub-word unit to a distribution with a decision tree.

with the leaf nodes.) The second assumption is that the posterior probability of an HMM state (in Equation 2.17) or equivalently the segmentation in “Viterbi-style” estimation does not change after a split. This avoids the need to re-estimate the counts belonging to each observation density after every split. In practice, these assumptions appear to work well (produce robust decision trees). There exist different schemes of tying mixtures within the decision tree framework, and a few of the popular ones are compared in [114, 83].

In building a word model for decoding, a particular context-specific phoneme is dropped down the tree and is guided by the linguistic questions at the branches. As illustrated in Figure 2.3, the acoustic model of the leaf that it lands in, is associated with the phoneme. Typically, a fixed topology of 3-5 five states is associated with each phoneme, and separate trees are used for each state of a phoneme.

In summary, word sequences are mapped to phoneme sequences using a dictionary. The

phoneme sequences may be mapped to their surface form using pronunciation models. All phonemes are assumed to have the same HMM topology, and the state observation density for a phoneme in a particular context is located using decision trees.

2.5 Large Vocabulary Search

The components described in the above sections can be put together in a few different ways to perform the task of large vocabulary automatic speech recognition, essentially the search in Equation 2.3. In all implementations, the language model, the lexicon, the sub-word units and their HMMs form a composite model. To make the search over this composite model possible, repeated evaluations of different speech segments are minimized using a variety of techniques such as shrinking the language model, arranging the lexicon in a tree form, and caching scores for the observation density (e.g., [92]).

In one implementation, all the components are represented in the form of a finite state transducer (FST). The component FSTs are composed *a priori* or on-the-fly using a lazy composition method to create a single compact model for all hypotheses. This composite FST can be searched using the Viterbi algorithm to recognize speech [89]. In virtually all implementations, the search space is also reduced by pruning away the unlikely paths at each frame or word boundaries.

The complexity of the search depends on the state space of the composite HMM, which in turn depends on the size of the lexicon, the value of n in the n -gram language model, the number of sub-word units, the number of clustered observation densities, and whether cross-word effects are modeled. For example, a pentaphone system with a trigram language model requires a lot more computational resources than a triphone system with a bigram language model. Often a first pass decoding is performed with a low complexity model to obtain a word graph or a lattice of a large number of possible hypothesis. Then, a more sophisticated model is used to re-score the lattice and pick the best hypothesis.

Instead of minimizing the sentence error in Equation 2.3, an alternative scheme which minimizes the word error rate also exists [120, 84, 134]. The model framework remains the

same; only the algorithm used in picking the best hypothesis changes.

State-of-the-art systems also use additional techniques to enhance performance. The impact of differences in acoustics from different speakers for the same sub-word unit is reduced at training using speaker-adapted training, and at testing using speaker adaptation (e.g., [30, 106]). Speaker adaptation is incorporated in a multi-pass search framework, often with more than one stage of adaptation. Combining different systems using a majority voting criterion (ROVER) also improves recognition accuracy [35]. These additional techniques have not been used in our experiments, as they are beyond the scope of this work.

Chapter 3

EXPERIMENTAL PARADIGM

To enable meaningful comparison of our results and conclusions with that of other work, all the experiments in this thesis are performed on a standard training corpus and a standard recognition task, both are described in Section 3.1. The experimental framework used for this thesis is outlined in Section 3.2. The observation densities of our acoustic models are tied differently from other ASR systems. This choice has certain trade-offs, which is also elaborated in Section 3.2. Finally, the performance of our baseline is compared with the state-of-the-art in Section 3.3.

3.1 Task

3.1.1 Training Corpus

In line with our objective to improve large vocabulary conversational speech recognition, the acoustic models in this work are trained on the Switchboard and the Callhome corpora [44]. These two corpora comprise of a collection of spontaneous telephone conversations between pairs of callers in American English, each conversation about 5 minutes long. Switchboard pertains to calls between strangers on a predefined topic, while Callhome is a more casual variant with calls between friends on any topic. Our work entails the use of a large number of contextual features (as explained in Section 4.2). To reduce the potential risks of data sparsity, we used all the training data which was available in the beginning of this work – about 140 hours of speech, incurring significant computational cost and time in training the acoustic models. About 80% of this data belonged to Switchboard, and a majority (about 60%) to female speakers.

3.1.2 Recognition Test

To draw reliable conclusions without incurring long experiment turn around time, we chose a subset of 1998 NIST Hub-5 development test set that was defined by BBN. The test set contains about 12.5k words in approximately an hour of speech from 28 speakers, taken from both Switchboard and Callhome corpora that were excluded from the training set. About 53% of the test speech is from female speakers. The output of the recognizer was evaluated using the standard criterion of word error rate ($= (C - I)/R$ where the number of words correct is C , inserted is I and the total number of words in the NIST reference transcript is R). The scoring was performed using “sclite” software from NIST that additionally takes certain spoken word equivalences into account (e.g., family’s, family is, family has).

3.2 Recognition System

The speech data was preprocessed, as explained in Section 2.1, to produce 14 dimensional vocal-tract-length-normalized MFCC vectors, at a rate of 100 vectors per second [131]. Each vector was augmented with its first order derivatives to serve as acoustic input to the recognition system. To reduce experimentation time, we used a re-scoring paradigm for decoding. For each utterance, the search was performed over a word lattice of possible hypotheses. A 35k vocabulary was used to generate these hypotheses. The lattices were derived from the 100 best hypotheses provided by BBN in 1998¹. The error rate of randomly selected hypotheses in the lattice is 55.4%, and the oracle rate is 29%. To further reduce the search time, hypothesized word times were used to constrain the lattice search space in re-scoring, i.e., the re-scored times were forced to occur within a window of $\pm M$ frames ($M = 40$) of the hypotheses. This constraint did not cost any performance loss on our baseline triphone system, and achieved the same results as decoding the word graph (or lattice) using an FSM decoder with no word time constraints. The language model used

¹It is often useful in re-scoring to combine scores from different acoustic models, but since our focus was on understanding the behavior of the syllable features, the BBN acoustic model scores were not used in the results reported in this thesis.

in all the experiments was a part-of-speech smoothed trigram trained with Broadcast news data as well as the Switchboard and Callhome data [62].

The experiments in this thesis are mainly concerned with improving acoustic models, so only that part of our system changed across experiments. A typical experiment consisted of training a new acoustic model, and evaluating its effectiveness on the test set. The decoder settings were fine-tuned on a baseline system, and then those parameters were left unchanged across all experiments in a series. Acoustic models were trained from a state-level Viterbi-alignments as shown in Figure 3.1 with techniques described in Section 2.4 and Section 2.3.1. Sufficient statistics for a single Gaussian full covariance distribution were collected for each unique acoustic unit, e.g. each state in each triphone or pentaphone. Then, and state-level decision trees were designed to cluster the different contexts associated with a specific position in a particular phone. The state-level distributions so obtained were re-estimated using word-level transcripts with EM algorithm. In experiments with variable topology, training acoustic models involved a few additional steps which are explained in Chapter 6. A large part of the experimental time was spent on training the acoustic model.

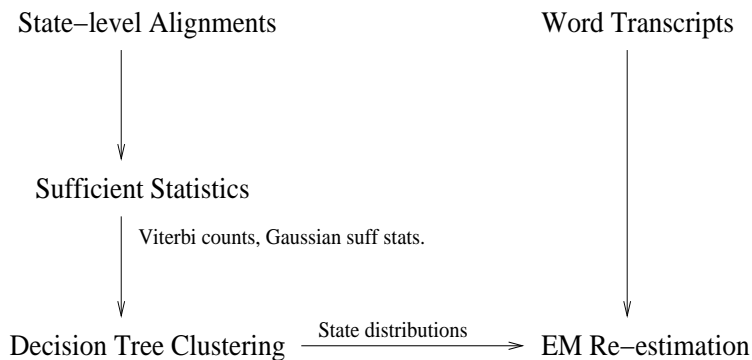


Figure 3.1: Sequence of steps in training acoustic models.

In our work, a simple variant of the standard decision tree clustering is used, where a covariance is shared over a subtree (node with all its descendants) [70]. This scheme

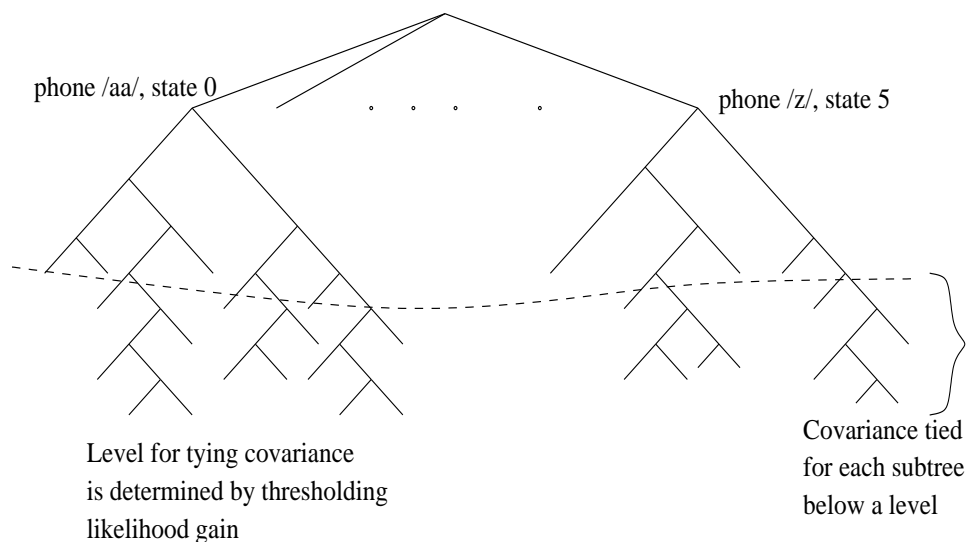


Figure 3.2: Sharing covariances and means at different levels in decision trees.

has two levels of tying in the decision tree – a shallow level to share the covariance, and a deeper level to share the means, as shown in Figure 3.2. We chose to cluster means instead of keeping the means of sub-word units as in [70]. In a few experiments, Gaussian mixture models with diagonal covariances were also used. We find that using a single Gaussian with a full covariance for an observation density with the two-level tying scheme (Baseline-ShCov) gives the same performance as an 8-component Gaussian mixture model with diagonal covariances (8-mix Gaussian). The size of the decision tree used in the single Gaussian system was about four times larger than the mixture Gaussian system, and used 10K covariances and 50K means. The mixture Gaussian system used 12K state clusters. The parameters of the shared covariance system (Baseline-ShCov) were found to converge with a far fewer number of iterations of the EM algorithm than in the case of Gaussian mixture models. We use 10K shared full covariance and 50K means in all experiments, unless stated otherwise.

This thesis has two kinds of experiments: fixed topology and variable topology HMMs. Our experiments with fixed topology uses a standard left-to-right five-state HMM topology, with skips as shown in Figure 2.2, requiring a minimum of three observation frames in a

phone.

3.3 Baseline Performance

The performance of our baseline triphone system on a similar test condition is equivalent to that reported by a leading research system in November 2000 (BBN) [25], without the multi-pass search refinements such as speaker adapted training, speaker adaptation and ROVER combination. These techniques usually give consistent additive gains on all systems. There are a few differences between the BBN system referred to in Table 3.1 and ours; enough to adequately explain the 0.9% gap. The BBN system uses normalized feature vectors, second derivatives of MFCC, and a different tying mechanism (state clustered tied mixture, with 3000 Gaussian clusters, 25000 mixture weight clusters and 80 Gaussians per mixture). With the exception of the FSM tools, our baseline system was built using software that was completely developed in-house [2].

Table 3.1: Performance comparison on 1998 NIST Development Task.

System	WER
BBN	43.4
Baseline-ShCov System	44.5
8-mix Gaussian System	44.5

Chapter 4

USING SYLLABLE AND WORD FEATURE

High level context can explain a certain amount of variation observed in conversational speech, and its incorporation into different levels of an automatic speech recognition system could improve performance. In the work described in this chapter, we investigated the role of syllable- and word-level features in clustering acoustic observation densities. Unlike previous work, we used a rich set of syllable- and word-level features and found an improvement in recognition accuracy. The new models are compared with pentaphones and the utility of various features are studied. Further, we studied the effects of re-syllabification on recognition accuracy. The models trained in these experiments used a fixed topology for all phones, which is subsequently replaced by a variable-length and/or multi-path topology in Chapter 6.

This chapter begins in Section 4.1 with a discussion of related work that explores different strategies to incorporate non-phonemic features. Section 4.2 describes our experiments, and reports our results. We also include a discussion about the usefulness of the additional features by examining the resulting decision tree. Our experiments and results on re-syllabification are explained in Section 4.3.

4.1 Beyond Triphones

Motivations such as those described in Section 1.1 have led numerous researchers to explore possibilities beyond using triphones as sub-word units for speech recognition. How can we improve upon triphones? Some researchers turned to units that were automatically derived from data to improve recognition [3, 27]. Automatic units seemed to work well only when sufficient examples were present. Alternatively, knowledge driven approaches

have been investigated. It has been hypothesized that human perception depends on a non-linear interaction between hierarchical layers of information, including syllables, words, utterance and topic (e.g. [48] among others in linguistic and psychology). Of these, word and syllable features have been prime candidates for investigation. The use of syllable features can be motivated by results from psychoacoustic studies. A good survey of the literature describing the experiments is in [126]. These studies test the capacity of human listeners to identify manipulated speech examples, and indicate that syllables may be units of perception. Replacing the nucleus with noise or silence does not seem to affect perception, while altering the onset or the coda does. Reaction time experiments, which ask the subjects to recall perceived units as quickly as possible, also show that the time scales of memory organization correspond to that of the syllable. Massaro describes an interesting collection of early experiments that attempt to understand the characteristics of the perceptual unit for speech [86]. Experiments where segments of speech were substituted with silence, or switched across ears, show that human perception of speech is at its worst when these interruptions correspond to syllable time-scale. Massaro points out that when units such as “di” and “da” are successively truncated the perception changes categorically from consonant-vowel cluster to non-speech, without hearing the phoneme “d” at any intermediate stage. It has been argued that if syllables are units of perception, a large portion of the co-articulation in fluent speech must occur within the syllable [126]. While a hierarchical structure that includes units larger than the syllable implies domains of co-articulations beyond the syllable, syllable-level structure must at least be a factor in acoustic variability.

Acoustic features that are closely related to syllables have been used in speech recognizers. Wu *et al.* [128] detected syllable boundaries using a neural network, and scored phone sequences at syllable boundaries using a syllable grammar. A potential problem with this approach is that detection of the syllable boundaries is unlikely to be robust in spontaneous speech. This approach has the general advantage, theoretically, of using longer units but does not take advantage of linguistic knowledge associated with syllable structure.

Separate acoustic features at multiple time or frequency scales have been used to cap-

ture the robustness seen in human listeners. Kirchhoff [75] integrated parallel sub-segmental phonetic units like phonation, manner, place, roundness, and centrality of articulation at the syllable level. Dupont and Boulard [32] combined phones and syllables scores computed from different frequency sub-bands at syllable boundaries. Wu *et al.* [127] combined semi-syllables (syllables split in the middle) recognized from modulation spectrogram with phones, which were recognized using a cepstral-based system, at the utterance level in an n-best re-scoring pass. In all three cases, small gains were reported for small tasks. The behavior for large vocabulary tasks is yet to be seen. In the two latter systems, the phone level acoustic models do not use syllable or higher information.

A number of researchers attempted to model the acoustics of frequent contexts exclusively by using long units, often in addition to the triphones. Matsumura and Matsunga [87], and Pfuu *et al.* [103] derived the long units from the text using entropy reduction criteria and maximization of coverage respectively. Hu *et al.* [60] grouped certain stop and vowel or semi-vowel pairs into long units. Deligne and Bimbot [27] derived long sub-word units from the data using a quantized intermediate representation. Many of these attempts showed small gains. Again, these approaches have the general advantage, theoretically, of using larger units but do not take the advantage of linguistic knowledge associated with syllable structure.

Syllables or syllable-like units have been investigated as an alternative to triphones [64, 85, 56, 31]. The role of syllable depends on the language. Experiments in other languages such as Chinese [80], Japanese [58] and Spanish [12] have found syllables to be beneficial sub-word units in recognition systems. However, the complex structure of the syllable in English makes it difficult to use them in a straightforward manner. Among the work in English, only [31] examined their use in large vocabulary conversational speech. They modeled the most frequent syllables as separate sub-word units, and the rest with triphones. This approach in conjunction with improvement in temporal structure of the acoustic model gave about 0.7% reduction (absolute) in error rate on their test set over a word-internal triphone system with bigram language model (over a baseline of 49.8%). A disproportionate

number of errors were found to be in words recognized by the triphone, and may be due to poor sharing of parameters between the triphones and syllables.

Syllables have also been used as an intermediate symbolic layer in a hierarchical representation with stochastic dependency between the layers (e.g. [77, 19]). In [19], Chung used syllable constraints to improve recognition on *Jupiter* (a small vocabulary weather-information task) as well as to hypothesize out-of-vocabulary (OOV) words. However, the acoustic models used in this work were not influenced by syllable structure.

In another approach, a few researchers began investigating the use of syllable features in improving the triphones. In a study conducted at a summer workshop in JHU [97], syllable- and word-level features were used in a decision tree framework for conversational speech. Triphones in a subset of standard training data for conversational speech were coded with these features, and clustered using decision trees. It was found that questions regarding them were asked early, i.e., near the top of the tree. They concluded that this may lead to finding better equivalence classes, and thus improve acoustic models. To make this study possible, the computational cost was reduced by ignoring the triphones that span word boundaries, i.e. evaluating performance with a word-internal triphone system. Thus, the usefulness of these features was not demonstrated conclusively.

The use of word position (initial, medial, final) as a context-conditioning feature has been shown to be useful in several studies, for both conversational speech [34, 52] and read speech [110], and is used in many research systems. Word type (function vs. content) [78] was found to be useful in experiments recognizing read speech. The use of syllable position alone has not so far proved to be useful [102, 52], though Paul reports a small gain when syllable position is used in combination with lexical stress tags. Paul's results for lexical stress are also mixed, with gains depending on the dictionary used. The mixed results on read speech could possibly be due to a variety of reasons, including inadequate levels for coding features or the sensitivity to the alignment of the training data used.

4.2 *Using Rich Syllable Feature*

Our motivation to re-evaluate syllable features in a clustering framework mainly stems from recent statistical analysis of conversational speech corpora, and a desire to overcome the shortcomings of previous work.

In a statistical study, Greenberg and Fosler found systematic variation with respect to the constituents of the syllable, namely onset, nucleus and coda [49, 38, 51]. They used a subset of the Switchboard corpus (for conversational speech) and a subset of TIMIT corpus (for read speech), both of which were annotated by linguists at the word, phone and syllable level. The perceived phones were compared with the lexical expansion of the spoken words. Both corpora show that the onset of a syllable maintains its canonical identity (sequence of phones) at most times (85-91%) regardless of the speaking style, and more often in the presence of consonant clusters. In general, the nucleus is prone to substitution by a wide range of vowels. The coda is less often realized in canonical form (phone sequence) in conversational speech (63%) than in read speech (81%). The coda is prone to deletion, but the absence of a coda does not impact canonical realization of the nucleus.

An analysis of errors made by a variety of state-of-the-art recognizers showed a strong correlation between successful recognition of a word and correct identification of its syllable onsets [50]. Estimation of tighter distributions for the onsets may improve recognition accuracy. These results together support the notion of syllable-initial strengthening, which has been observed as a more gradient phenomena in EPG studies [73]. The EPG studies by Keating also suggest that the amount of strengthening may be equal to that in word-initial position. While categorical phonetic changes can be accommodated by a larger phone inventory and a good pronunciation model, phone substitutions and deletions fail to capture more gradient aspects of variation such as strength of a stop release. This motivates us to investigate automatic training of observation distributions in acoustic models using syllable- and word-level features.

4.2.1 Syllable Features

In this work, we used a rich set of symbols to represent syllable structure, which includes consonant cluster and ambisyllabicity. The lexicon and syllable coding system used in this work was developed at the 1996 JHU workshop, and later extended for new words. This dictionary was obtained by automatic syllabification of a stress marked dictionary called the Pronlex, using Fisher’s implementation of Kahn’s principles for English syllabification [36]. This syllabification scheme does not take morphology into account. The lexical expansion of words are coded at phoneme level as illustrated in Table 4.1 (e.g., “arc aa:0:3:2:1 r:1:3:4:1 k:2:3:5:1”). Note that the position of the phone in the syllable distinguishes between onset consonants which are and are not in clusters, and marks consonants as onset even if they are not syllable initial, unlike previous work on syllable position. The actual stress in the conversational speech may differ from canonical patterns of stress in a word. Unlike previous work, stress is marked using three levels (as in Pronlex and most dictionaries): primary, secondary and unstressed. The Pronlex convention tends to mark all monosyllabic words with primary lexical stress. This was modified at JHU so that one-syllable function words receive a secondary stress, representing the possibility that their vowel may be either full or short and the associated distributions may be broader.

Linguistically motivated questions were defined on these features for a phone and its immediate neighbors. The list of these additional questions can be found in Appendix A, and comprise: 6 questions each on the position of the phone in the word and the position of syllable in the word, 2 on stress, and 21 on the position of the phone in the syllable.

The state-level segmentation of the training data from a triphone system for each phoneme was encoded with the syllable- and word-level features from the corresponding lexical expansion of the word in the lexicon. As a result, the number of states in the sub-word units increased as shown in Table 4.2. For comparison, we also trained a pentaphone system. Notice that the number of unclustered states in a pentaphone system is more than 3 times that of a system with all the syllable- and word-level features.

From these alignments, single Gaussian sufficient statistics were computed for each

Table 4.1: Coding of syllable- and word-level features in the dictionary.

Digit	Phone posn. in word	Syllable posn. in word	Phone posn. in syllable	Stress
0	first	first	onset initial	stress-less
1	middle	middle	onset other	primary
2	last	last	nucleus	secondary
3	only	only	coda only	
4			coda initial	
5			coda other	
6			ambisyllabic	

Table 4.2: Increase in size of sub-word unit inventory with features (160 hrs)

Sub-word Unit	Male	Female
a) Triphone	141K	151K
b) Pentaphone	2261K	2660K
c) Triphone + Word-posn.	254K	276K
d) Triphone + Word-posn. + Syllable-feats.	643K	709K

context-specific phoneme. Robust models were then trained by clustering the sufficient statistics using decision trees with two-level tying (as explained in Section 3.2). Separate trees were clustered for each phone state, with augmented questions about the new features. In all cases, the trees were pruned to the same size to obtain the same number of means and covariance parameters. The model parameters were then refined using a few iterations of the EM algorithm.

4.2.2 Results and Discussion

With the addition of more features, the likelihood of the models (after EM iterations) improves as shown in Table 4.3. This is not surprising, since the number of partitions that the decision tree explored also has increased. The pentaphone system showed the highest likelihood on the training data. On the held out set, the log-likelihood of the system with syllable- and word-level features were better than triphone, and slightly better than the pentaphone system. Cross-validation with a maximum likelihood objective is often used to compare the performance of a set of competing models and to evaluate their ability to generalize to unseen data (e.g. in the model selection literature [37, 133]). In a similar vein, our comparison indicate that these new models may have a capacity to generalize better than triphones or pentaphones.

Table 4.3: Comparing the average log-likelihood on the training data and on a held-out data set.

System	Training	Held-Out
a) Triphone	-12.85	-20.06
b) Pentaphone	-12.57	-19.93
c) Triphone + Word-posn.+ Syllable-feats.	-12.58	-19.84

Using the set of new models, recognition was performed and the results reported in Table 4.4 show a small gain with syllable- and word-level features. While the difference in performance of the triphone and pentaphone systems is not statistically significant¹, the system based on syllable- and word-level features is significantly better than triphone system, according to NIST statistical significance tests (.05 Wilcoxon Signed Rank Test on speaker word error Rate (%); .01 McNemar Test on sentence error). This result has also been confirmed in subsequent experiments with a Gaussian mixture system (described

¹Other systems showing improved performance with pentaphones appear to have increased numbers of parameters in the pentaphone system, whereas here the number is constrained to be roughly the same.

in Section 3.2 trained from the same state-level alignments) on the same task, showing an improvement from 44.6% on triphone system to 44.1% with syllable- and word-level features. Contrary to other reported results, this gain is not simply due to use of word position features, which accounts for roughly half of the improvement from the triphone system to the system using syllable features. We discuss the results in relation to previous work in Section 4.4.

Table 4.4: Word error rates using different features in state clustering.

System	WER
a) Triphone	44.56 %
b) Pentaphone	44.37 %
c) Triphone + Word-posn.	44.31 %
d) Triphone + Word-posn.+ Syllable-feats.	44.05 %

The computational cost for decoding and training were also significantly lower for the system with syllable- and word-level features. In training, the pentaphone system required testing 350 potential partitions for clustering up to 2.5M sub-word units, while the system with syllable features required testing only 200 potential partitions for 700k sub-word units (more than factor of four savings). In decoding, the system with syllable features was 20% faster. In addition, unlike pentaphones, which incur extra computational expense and software flexibility to span the two forward contexts, the coded triphones only look ahead as much as a single phone and could be incorporated in a standard first pass triphone decoder.

4.2.3 Observations on the Use of Syllable and Word Features

To study how the syllable features were utilized by the acoustic models, we analyzed the questions that were chosen in the decision tree for clustering states of sub-word units. The number of questions asked about a feature (the number of partitions tested) in the decision

trees and the percentage of total data affected by a feature can be used to understand how useful a feature has been in training acoustic models.

The results are summarized in Table 4.5. Even though questions about phone identity allow five times as many degrees of freedom in partitioning the training data as all other features combined, the decision trees chose the latter about one third of the time. This share did not come at the expense of questions about word position, since only 18% of the questions in a word position only system were non-phonemic. Among all the features, the fewest questions are asked about position of the phone in the syllable. However, it affected a disproportionately larger amount of data. This feature is likely to be used higher up in the tree, suggesting that questions about it generalize over a large fraction of the data. Questions about lexical stress and position of the syllable in the word tend to be asked at the bottom of the tree. It was also observed that the questions about the position of the center and the right phone in the syllable are significantly more important (4-6 times the fraction of data affected) than that of the left phone. Whether the center phone was in a monosyllabic word was among the top questions about the position of syllable in word, as one might expect from the gains observed in modeling monosyllabic words [31], but the amount of data affected was not high so it did not stand out as a particularly important feature. In general, the pattern of usage of the features across gender is similar.

Table 4.5: Usage of syllable and word features in automatically trained decision trees.

Feature	Questions in the tree	Data affected	Degrees of freedom
Triphone	66 %	76 %	180
Phone in Word	10 %	9 %	6
Syllable in Word	11 %	4 %	6
Phone in Syllable	3 %	6 %	21
Stress	11 %	5 %	4

4.3 *Re-syllabification*

Re-syllabification complicates the role of the syllable in English. Previous work can be criticized for ignoring its effects on recognition performance. In the work described above, the syllable- and word-level features were obtained from the lexicon for each word separately. The canonical syllabification of certain words (as coded in the lexicon) may undergo transformation under certain conditions. So, the tags need to be changed accordingly. We investigated the influence of re-syllabification on the clustered models.

4.3.1 *Scheme for re-syllabification*

Syllabification may vary systematically at word boundaries, depending on the neighboring word. For example, “choirs and” may be syllabified as “[k w ay r][z ax n]” instead of “[k w ay r z][ax n]” and “it’s a” as “[ih t][s ax]” instead of “[ih t s][ax]”, where the latter forms are obtained by stringing together syllables of each word. A complete description of the process of re-syllabification in English is relatively complex. However, the process of re-syllabification can be explained to a large extent by an empirical rule – the Sonority Dispersion Rule [24, 74]. The simplest syllabification is the one with maximal and most evenly distributed rise in sonority² at the beginning and the minimal drop in sonority at the end. Based on this principle, the Sonority Dispersion Rule moves the syllable boundary amongst the consonants to maximize the slope of sonority in the onset-nucleus demi-syllable, and minimizes it in nucleus-coda demi-syllable. Sonority ranks can be assigned to groups of phonemes based on their phonetic properties as shown in Table 4.6 [24]. For our application, we have ignored the extra complexity associated with assigning a sonority rank to “s” and have followed the convention in [24](see [24] for discussion).

Further, empirical observations about the shift of syllable boundary across words can be quantified using a measure proposed by Clements called dispersion. Dispersion in a

²The degree to which a speech sound is like a vowel.

Table 4.6: Sonority rank for phonemes used in our work.

Phoneme groups	Obstruents	Nasals	Liquids	Glides	Vowels
syllabic	-	-	-	-	+
vocoid	-	-	-	+	+
approximant	-	-	+	+	+
sonorant	-	+	+	+	+
Rank	1	2	3	4	5

demi-syllable³ D is a function of distance between consecutive phonemes d_i , $D = \sum_{i=1}^m \frac{1}{d_i^2}$. The distance d_i is the difference in sonority rank between the i th adjacent phoneme pair, and D is computed over all pairs m in a demi-syllable. This principle appears to explain most re-syllabifications seen in English [24]. Additionally, single vowel onsets are dispreferred. Figure 4.1 illustrates an example of re-syllabification across the words “mark all”. A majority of re-syllabifications is known to occur at words that begin with vowels.

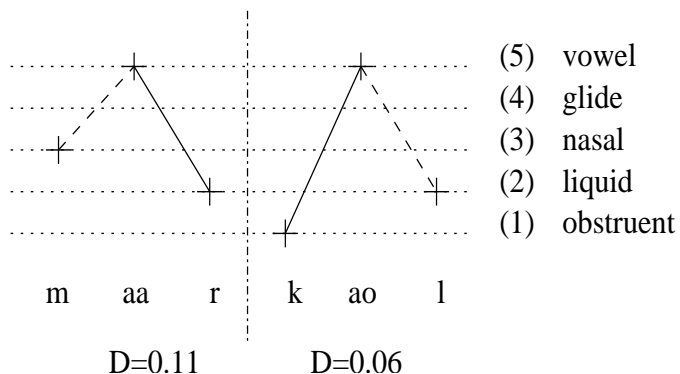


Figure 4.1: Re-syllabification of “mark all” using the Sonority Dispersion Principle.

Since a small number of rules capture a large fraction of the cases of re-syllabifications, it is possible to incorporate re-syllabification into the model of a word by allowing alternate

³Onset-nucleus or nucleus-coda pair.

“pronunciations”. The specific method used in our work is as follows:

1. Candidates for re-syllabifications include only open-vowel syllables that are word initial and do not follow a pause or a vowel (based on forced alignments in training and N-best hypotheses in testing). Our analysis of ICSI transcriptions shows that 73% of the re-syllabifications in speech occurs in open-vowel word-initial syllables.
2. If the syllable is preceded by a single consonant, mark that consonant as optionally ambisyllabic.
3. If the syllable is preceded by more than one consonant, apply the Sonority Dispersion Rule to obtain the alternate syllable boundary.

These rules were applied to augment the sequence of coded phonemes that were obtained from the lexicon, with an alternate re-syllabified path. The decoder was allowed to choose either of the two path, as described in the experiments below.

4.3.2 Experiments, Results and Discussion

First, we studied the effect of re-syllabification on the training data. Word-level transcripts were expanded into sequences of phonemes coded with syllable- and word-level features, using the lexicon described in 4.2. The rules described above were applied to generate alternate syllabifications across word boundaries for all vowel-initial words in the training set. This produced about 13% coded phonemes in the alternate re-syllabified paths, and constituted only 50 new types of coded phonemes (about 3% of the total number of uniquely coded phones in the lexicon) distributed across 15 phonemes. Next, using the acoustic models developed in Section 4.2, we let the decoder choose the best path from the lattice of possible paths. A few examples of re-syllabification that were chosen are listed in Table 4.7.

Though the examples were encouraging, the number of alternate coded phonemes that the decoder chose was only about 7% of those hypothesized. This was about 1% of the total

Table 4.7: Examples of re-syllabifications chosen in the training data.

Partial word sequence	Re-syllabifications picked by decoder
bridge across	[b r ih [jh] ax] ...
choirs and	[k w ay r][z ax n] ...
commercials and	[k ax][m er][sh ax l][z ax n]
did it	[d ih [d] ih t]
drugs out there	[d r ah g][z aw t] ...
gets a little	[g eh t][s ax] ...
got into	[g aa [t] ih n] ...
its an	[ih t][s ax] ...
lots of times I'd	... [t ay m][z ay d]
like a	[l ay [k] ax]
makes a lot	[m ey k][s ax] ...
minutes up	... [m ih][n ih t][s ah p]
takes it	[t ey k][s ih t]
takes a long	[t ey k][s ax] ...
that's easier done	[dh ae t][s iy] ...
watch all	[w aa [ch] ao l]
work and	[w er [k] ax n]
wife and	[w ay [f] ax n]
years ago	[y iy r][z ax] ...
years it	[y iy r][z ih t]

labels in the training data and involved many different phones. In comparison, we observed about 4.5% vowel-initial re-syllabifications in ICSI transcripts. Thus, we may have captured about one in four re-syllabifications. Examination of the chosen re-syllabifications indicate

that a large number of them involved phoneme “s” and “z” or were ambisyllabic. Since the total data affected was low, we did not expect re-syllabification to affect our acoustic model significantly, and continued using the same models without bootstrapping.

The impact on test data was evaluated using N-best re-scoring. After expanding each of the 100 hypotheses into a lattice with alternate re-syllabification paths, we let the decoder choose the best path using the acoustic models developed in Section 4.2. The word error rate did not show any improvements.

In a large number (90%) of re-syllabifications in the ICSI transcriptions, the syllable boundary moves across the word by one phone (e.g. [th ih ng z] [aa n] \rightarrow [th ih ng] [z aa n]). It is likely that in these instances the right context of coded-triphone already encodes re-syllabification implicitly. Re-syllabification tends to occur in high frequency word pairs which may provide sufficient instances to learn this. In addition, it may also be the case that re-syllabifications where a coda becomes ambisyllabic are missed by our system because of the acoustic similarity. In acoustic clustering, among the data affected by questions about syllable position, only 5% was affected by questions about coda and 11% about ambisyllabic.

4.4 *Summary*

In summary, we used syllable- and word-level features in a clustering framework and found small but consistent improvement in recognition accuracy over the triphone system. The gains found are higher than any previous work for a conversational speech recognition system with a baseline of similar complexity [52, 110], and this may be due to the more extensive lexical annotation and/or the larger training data used. Among the different syllable- and word-level features, position of the phone in the word and that in the syllable appear to impact the clustering most.

Compared to pentaphones, the models clustered using syllable- and word-level features were found to be computationally less expensive to train, more than a factor of four in savings. The syllable models also lead to faster decoding than pentaphones, in part because

of the smaller state space. It could also be due to tighter distributions of the new models, as observed from the higher log-likelihood on the held out set. This opens the possibility of using these models in first pass decoding.

We also found that although instances of re-syllabification can be located automatically in the data, they do not impact recognition performance. However, the cost of allowing re-syllabification is small and it may be worth including it in a system with more extensive pronunciation modeling than that used here.

Chapter 5

MULTI-STAGE CLUSTERING

In the previous chapter, we investigated the use of syllable and word level features for clustering acoustic models in a decision tree framework. As a natural sequel one might ask - why not add more linguistic features in the decision tree to improve acoustic models further? Speaking rate, prosodic structure, and part-of-speech class labels are good candidates, although admittedly few of these can be obtained from the hypothesized word string alone and need to be treated as hidden variables. These features can reduce the uncertainty encountered in conversational speech, and researchers have been studying different ways of using them in an ASR system [47]. Few of these features, particularly word type, pitch and speaking rate have been successfully used to augment ASR system in different ways [78, 39, 132].

In this chapter, we address the problem associated with incorporating a large number of features in training decision trees. In Section 5.1, we examine the difficulties in clustering high-dimensional feature vectors. For ASR applications, we propose an extension that overcomes these limitations, which is described in Section 5.2. The method was verified on a large task of clustering pentaphones, the results on recognition performance are reported and discussed in Section 5.3.

5.1 Clustering with Large Feature Set

Decision tree based distribution clustering in ASR was first used to cluster triphone contexts in a read speech corpus, where two separate trees were grown for male and female speakers, with a total of about 80 hours of speech [70, 130]. Progressively, it has been employed for larger tasks, and now as much as 250 hours of speech are clustered with pentaphones and

word position features [55].

When the number of feature values increase, a few factors start affecting the automatic training of decision trees. The number of unique labels tend to increase, and the associated sufficient statistics needed to train the tree requires large amounts of memory. For example, Figure 5.1 shows the amount of memory required for sufficient statistics (means and full covariances of single Gaussian distributions for 5-state phones) in about 80 hours of speech as the triphones are incrementally augmented with word position and syllable features, and pentaphone contexts. The number of tokens also increase with more training data, as more infrequent phone contexts are observed. With the increase in number of features, the number of partitions that need to be tested also rises. The computational cost rises in proportion to the product of these two increases.

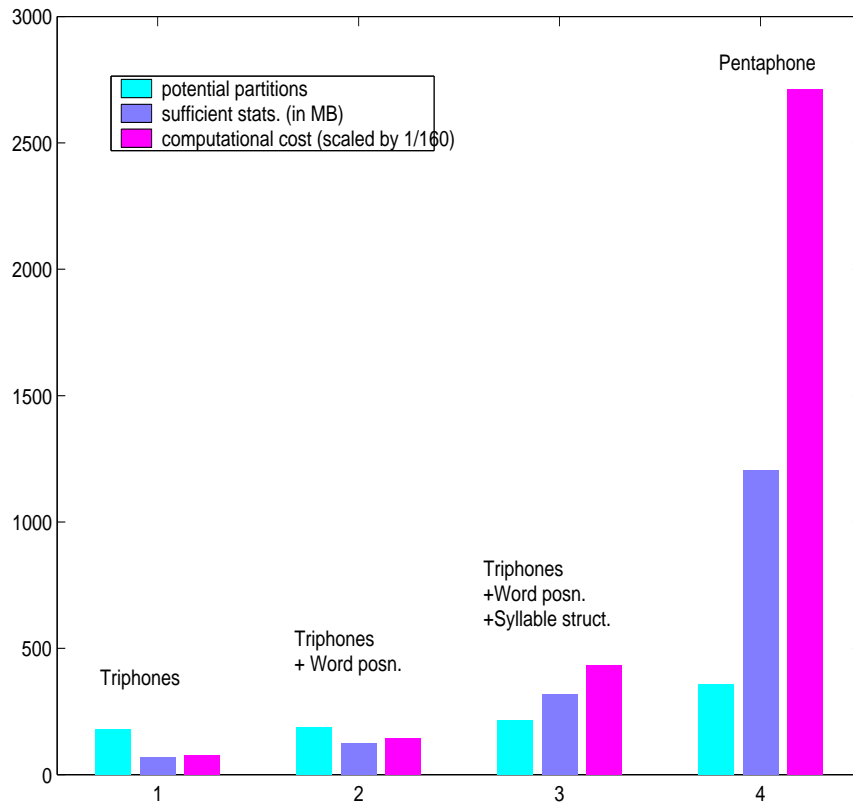


Figure 5.1: Increase in training complexity with features - triphone through pentaphone.

These problems have restricted previous work on tagged clustering. For example, in [97], experimental costs were reduced by restricting the use of syllable boundary and stress only to word-internal triphones, discarding triphones that spanned word boundaries. In other work, cross-word context is used but only with simple tag sets such as word position (begin, middle, end). We address these problems by introducing a new tree design technique based on multi-stage clustering.

When the number of feature values increase, typically the proportion of infrequent labels also increases, resulting in less reliable sufficient statistics. Use of a large number of features, also increases the data fragmentation, as a result of which a number of copies of a subtree are potentially formed. Or, certain good splits might not be selected because the associated examples are distributed across many nodes. Consequently, the partitions learned using singleton features may not represent general trends in speech. In the machine learning literature a number of solutions have been attempted to alleviate this problem. Use of compound questions has been a prime candidate, however the cost of selecting the best set of features for a test at a node also increases with number of features. A variety of methods have been attempted to address the problem of data sparsity, including use of decision graphs [76, 96], pylons [4] and soft decisions [107] (for additional pointers, see Murthy’s comprehensive multi-disciplinary survey on decision trees [91]), however, no clear solution has emerged.

5.2 *Multi-stage Clustering*

Our approach to reduce the storage and computational costs for clustering is based on dividing the task into multiple stages. The decision tree can be viewed as a function, \mathcal{T} , that maps a feature vector, \mathbf{f} , consisting of contextual information to an index, a , of an acoustic model, thus $\mathcal{T} : \mathbf{f} \rightarrow a$. As illustrated in Figure 5.2, for two-stage clustering, we group the contextual information into two feature vectors \mathbf{f}_1 and \mathbf{f}_2 , optionally allowing some common components between them. In the first stage, the training data is annotated only with the values of vector \mathbf{f}_1 . Using the annotated data, we grow a decision tree, \mathcal{T}_1 ,

which maps the different values of \mathbf{f}_1 to the index of its leaves b , thus $\mathcal{T}_1: \mathbf{f}_1 \rightarrow b$. In the second stage, the training data is annotated with \mathbf{f}_2 along with the value of b which is obtained by dropping its context \mathbf{f}_1 down the tree \mathcal{T}_1 . Using the newly annotated training data, a new decision tree, \mathcal{T}_2 is grown that maps $[b \ \mathbf{f}_2]$ to the index of acoustic models as represented by the leaves of \mathcal{T}_2 , thus $\mathcal{T}_2: [b \ \mathbf{f}_2] \rightarrow a$.

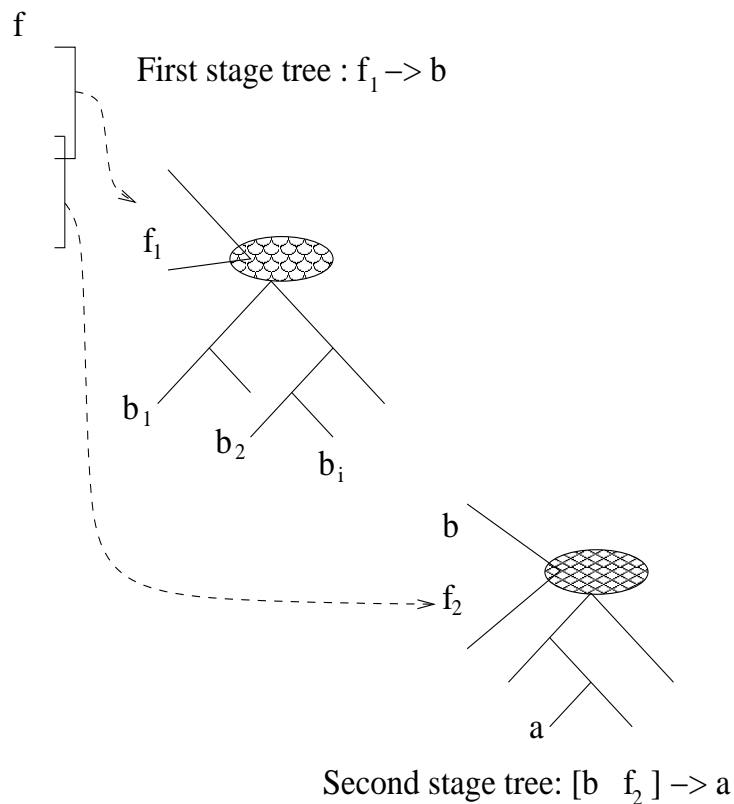


Figure 5.2: Multi-stage clustering illustrated with two stages.

In current decision tree clustering for speech recognition, questions about features are defined by hand and are linguistically motivated. This same scheme can be used for the features in \mathbf{f}_1 and \mathbf{f}_2 , but not for the index b . Allowing all possible partitions of b is impractical since there are $2^{|b|}$ binary partitions, and to use the features in the first stage

adequately, the size of \mathcal{T}_1 needs to be large. To solve this problem we define questions that test whether a node b belongs to a subtree of the first tree \mathcal{T}_1 or not. Such questions are equivalent to compound questions which are obtained by performing an “and” operation on a set of binary questions about the features in \mathbf{f}_1 . Defining questions on subtrees permits the decision tree to test a large number of partitions, and is more efficient than allowing all partitions.

Once the second stage tree \mathcal{T}_2 is grown, the questions on subtrees in \mathcal{T}_1 are replaced with the equivalent compound questions to obtain a single tree. The compound questions can also be expanded out as a series of questions on one feature at a time. During decoding, this tree can then be used just like the current trees. Note that, in principle, there is no limit to the number of stages, but this work considers only two.

The multi-stage clustering techniques helps ameliorate the problem of sparse data by reducing the number of coded units for which sufficient statistics need to be estimated, since only a subset of the features are used at each clustering stage. The root node at every stage has all the data available to it or, in this case, all the data associated with the particular state and the phone. The number of elementary units that need to be clustered in stage i depends on the features \mathbf{f}_i used in that stage and, if $i > 1$, the number of leaves of the preceding tree \mathcal{T}_{i-1} . Both of these factors can be controlled to reduce the effects of data fragmentation, essentially by trading off the potential for more directly modeling interaction between features (with a large dimension \mathbf{f}_i) with the robustness (and computational) advantages of a low dimension feature set \mathbf{f}_i . Note that robust estimation of statistics of elementary units also benefits from the general principle of increasing system complexity incrementally. In particular, we use phone alignments from our best triphone system, rather than bootstrapping from monophone models, as shown to be important in [52].

The storage and computational cost of the multi-stage clustering depends on various factors. The number of sufficient statistics that need to be clustered in the two stages is determined by the number of components used in \mathbf{f}_1 and \mathbf{f}_2 , and the size of \mathcal{T}_1 , as mentioned

above. To approximate the order of computation, consider N features, each of which takes M values and can be partitioned into Q classes. To train a single stage tree, the order of memory required would be $O(M^N)$, and the computation to split a node would be $O(QNM^N)$. Now, if the N features are divided into two sets N_1 and N_2 features, the order of memory required would be $O(M^{N_i})$, and the computation would be $O(QN_iM^{N_i})$. Since $N > N_i$, the savings in memory and computation is exponential.

In practice, the number of sufficient statistics is limited by the diversity of the data, and also depends on how uniformly the training data is divided into the clusters (how balanced the tree is). If a maximal tree is grown for \mathcal{T}_1 , then the multistage clustering will be computationally more expensive than clustering a single tree. The size of \mathcal{T}_1 may be set using a constraint on total number of leaves, or saturation of likelihood increase. To reduce the number of partitions that are tested in the second stage, we selected only a subset of subtrees (the largest) from the first stage during clustering at the top of the second stage, where the largest proportion of the computation occurs in training decision trees.

5.3 Experiments and Discussion

To evaluate the effectiveness of multi-stage clustering we trained gender-specific pentaphone systems using standard single-stage clustering and two-stage clustering. The systems were trained from a base triphone alignment with one pass of Viterbi training and a few passes of EM.

For the two-stage clustering, the first stage tree was grown in two steps, using only the identity of second neighbors as features. Initially, a large tree was grown for each phoneme and each state. For each state, a size was chosen to contain 98% of the likelihood gain across all the phonemes for that state. This resulted in a tree size of about 1000 clusters for each state. Then, all trees for each state were pruned concurrently to this chosen size. In the second stage, we clustered the data using the leaf indices of \mathcal{T}_1 along with the identity of first neighbors (the triphone context) to obtain the final models. When the features were clustered in the other order, that is with the first neighbors in the first stage and the second

neighbors in the second stage, the average log-likelihood of the models on the training data was about 5% lower. This suggests that it is better to cluster the least important features in the earlier trees, and to cluster the important features in later stages (thus allowing them to interact with features from previous stages).

After EM iterations converged, the models were tested on a held-out set and the observed log likelihood are compared in Table 5.1. The models learned in the two-stage system have a about the same average log-likelihood.

Table 5.1: Average log-likelihood on a held-out data set using one vs. two stages of clustering.

System	Held out
a) Pentaphone: 1 stage	-19.93
b) Pentaphone: 2 stage	-19.92

The two-stage pentaphone system for both genders performed as well as the one-stage systems, as reported in Table 5.2. Thus the result shows that incorporating features in multiple stages is a viable method for using a large number of features in acoustic modeling.

Table 5.2: Word error rates of systems trained with one vs. two stages of clustering.

System	WER
a) Pentaphone: 1 stage	44.37 %
b) Pentaphone: 2 stage	44.39 %

The memory used in clustering is directly proportional to the number of unique contexts to be clustered. In the single-stage pentaphone system, we had about 2M unique contexts, whereas in the two stage system, we had only about 78K in the first stage and less than 0.5M in the second stage, thus reducing the memory requirement at any time by a factor

of 4. The computational cost of two-stage clustering in this case is half that of single-stage clustering.

5.4 Summary

In this chapter, we have described an algorithm that could potentially reduce the difficulties encountered in clustering a large number of features. The algorithm breaks up the task into multiple stages, and yet allows significant interaction between stages. Experiments on Switchboard task with pentaphones verify that the method of interaction between the stages is effective. Potentially, it gives an exponential saving in memory and computational cost, and at the same time reduces the risk of unreliable labels and data fragmentation.

The subtrees define compound questions over the subset of features used in that tree. The use of these subtrees as questions in a subsequent stage helps creation of robust representation of the structure in data, thus ameliorating the data fragmentation problem to a certain extent. Our limited experiments on different splits of features set show that the subsets should be chosen in an order such that the earlier stages have features which influence acoustic models less than the later stages. Much more sophisticated algorithms could be envisioned for choosing subsets of features for different trees; we chose not to explore these since we expected the impact on performance would be minimal.

Chapter 6

TOPOLOGY REFINEMENT

Typically, in a large vocabulary speech recognition system phones are modeled by a fixed HMM topology, which may contain transitions to skip certain states. The most popular topology consists of 3-5 states, each with a multivariate observation density and the same set of non-zero transitions between the states, irrespective of the phoneme and its context. This may not be an optimal representation of a phone, for reasons described in Section 1.2. The questions addressed in this chapter are whether we can model the phonemes better with variable HMM topology, and whether that helps recognition.

Several researchers have worked on general techniques to learn HMM topologies appropriate for different phones and contexts, and their work is summarized in Section 6.1. In Section 6.2, we pursue our theme of tailoring sub-word units, in this case topology of a phone-sized unit, to suit its wider context. We simplify the topology design problem, and describe an algorithm to solve it. Further, in Section 6.3, we outline a method for initializing these models, and describe a series of experiments performed on the Switchboard task to understand its impact on modeling phones and on recognizing spontaneous speech. The lessons learned are summarized in Section 6.4.

6.1 Learning HMM Topologies

For automatic speech recognition, researchers have investigated various schemes for learning HMM topologies [82, 121, 99, 3, 31, 119, 53, 54, 33]. Lockwood and Blanchet used iterations of corrective training procedure, where new topologies were hypothesized based on recognition error, and tested on a held out set [82]. They reported a performance gain on a small task, but for a large task their algorithm would be computationally expensive

and might not converge. Stolcke and Omohundro proposed a method that uses a Bayesian posterior probability criterion to reduce an initial network by successively collapsing similar states [121]. One drawback of this algorithm is that it does not have a mechanism to hypothesize topologies for infrequent context-dependent phonemes. In a divisive state splitting algorithm developed by Ostendorf and Singer [99], clusters are formed by contextual and temporal splits of speech segments to maximize likelihood. However, for a large vocabulary task testing all combinations of temporal and contextual splits would be computationally prohibitive. Use of automatically learned sub-word units has been reported to perform better than the phone-based units for small tasks, however they do not perform as well as phone-based systems on large tasks, mainly due to poor models for infrequent words [3]. At a speech recognition workshop at JHU, a method of separating “modalities” (which they describe as classifiable variation in acoustic data) in the syllable was investigated, and the results reported did not show significant improvements [31]. Another variation of this approach is to cluster training data using a collection of pairwise distance measures, where each measure is evaluated as the log-likelihood of a training sequence given an HMM model [119]. The number of HMMs need to be known *a priori*, which is difficult to know in the case of a large vocabulary recognizer. In summary, these approaches are not directly useful for large vocabulary speech recognition tasks.

Recently, two methods have been tested on spontaneous speech tasks and were found to improve performance. Hain and Woodland developed a method for learning context-dependent topologies using an intermediate symbol sequence, where symbols represent strictly left-to-right HMM topologies with possibly different numbers of states [54, 53]. A stochastic mapping is learned from a phoneme sequence to this symbol sequence using an instance of EM algorithm. They report an overall performance gain of 2.4% (absolute) on a Switchboard task. Their method of selecting symbols requires careful initialization, and a series of complex and incremental model changes. Eide developed a method where a phoneme in a sequence is mapped to a network representing an HMM [33]. This is done in three steps. First, a decision-tree-based pronunciation model is designed to map a dic-

tionary phoneme in a particular context to a context-dependent unit. Second, an initial sub-phoneme level network is constructed at each leaf node to explain all observations in a training set that fall in it. Then, this network is compressed using Stolcke merging [121] on a held out set. This method gave a 0.7% (absolute) improvement in WER on the spontaneous segment of the Broadcast news task. In these methods, only phoneme contexts were taken into account, and conditioning on a wider-context is likely to help further.

6.2 An Algorithm to Cluster HMMs

In our work, we investigate a different approach where the emphasis is on using contextual information to a greater extent than the previous methods. The contextual information we use include syllable- and word-level features that are described in Chapter 4. The underlying idea is that the wide context would enable us to predict a more constrained HMM topology than the current fixed topology. These variations may occur across phonemes, and also across contexts in the same phoneme. For example, the phoneme “d” in “and now” may require fewer states than “d” in “mandate”. In both cases, phoneme identity and wide-contexts are likely to be good predictors.

The general problem of HMM topology design is a difficult one. To simplify the task of learning HMM topologies, we assume that a context-dependent phoneme can be adequately modeled by one or two parallel paths. These paths are chosen from a closed set, where each path represents a strictly left-to-right HMM and has a different number of states than the other paths. For example, a set of candidate HMM paths may be as shown in Figure 6.1.

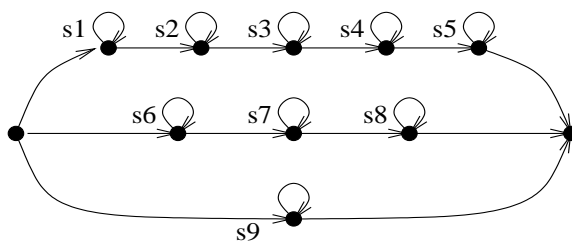


Figure 6.1: A set of candidate HMM paths for a context-dependent phoneme.

In principle, by sharing a few state observation densities across the paths, phonemes could have a more complex topology without substantially increasing the number of free parameters. The parallel paths here are meant to model both the temporal variation and the associated variation in observation space. This is unlike the use of parallel paths in Iyer et al. [61], where a fixed number of paths containing the same number of states were used to represent a mixture of trajectories.

Now, the task becomes one of learning a mapping from a context-dependent phoneme to a set of HMM paths. In pronunciation modeling, where a similar situation is encountered, many researchers have observed that the number of paths allowed in a context should be small to reduce confusability. So, we learn a mapping from a context-dependent phoneme to one or two HMM paths. To have the ability to hypothesize an HMM topology for all contexts and to pursue our aim of using linguistic information, we use a decision tree to learn this mapping. In the rest of this Chapter, these decision trees are referred to as the path decision trees to differentiate them from decision trees which model observation densities (or the state decision trees).

Among the set of candidate HMM paths, if there were no additional constraints, it is possible that two or more HMM paths may adequately model the same observation sequence for a given context. For example, a five frame cepstral sequence may be modeled by any of the three paths shown in Figure 6.1. To minimize this redundancy, we impose an additional constraint using an HMM distance metric, which is further described in Section 6.2.1. If two HMMs paths are near each other, we eliminate the path with the larger number of states.

Motivated by the above ideas, we propose an algorithm for mapping context-dependent phonemes to HMM topologies, as outlined below.

1. Train a system with context-dependent phonemes (context includes triphones, syllable- and word-level features) which are modeled by a 5-state HMM with skips, as described in Chapter 4. The state observation densities are modeled with Gaussian mixtures and smaller decision trees.

2. Build models of the form in Figure 6.1 by cloning states and associated decision trees, as described in Section 6.3; run EM; and prune low-occupancy leaves of the cloned state-level decision trees and components of Gaussian mixture models.
3. For each context-dependent phoneme in the training data, prune similar paths using an HMM distance (keeping the shorter ones). Using the resulting models, obtain state-level Viterbi alignments of the training data.
4. Estimate posterior path frequency count for each context from the state labels, and grow the path decision trees. Further prune low probability paths at the leaves of the path decision trees.
5. Using these models generate a new state-level Viterbi alignment, train a final set of state observation trees, and re-estimate the model parameters with EM.

Of these steps, (3) and (4) introduce new ideas which are explained in the sections below. The specific implementation details related to these and other steps are further described in Section 6.3. The HMMs learned using this algorithm are referred to as multi-path HMMs in this chapter.

6.2.1 Distance Metric for HMMs

A distance metric $D(h_1, h_2)$ between two HMMs, say h_1 and h_2 , can be formulated as follows.

$$D_s(h_1, h_2) = \frac{1}{2} [D(h_1, h_2) + D(h_2, h_1)] \quad (6.1)$$

$$D(h_1, h_2) \triangleq \log p(a(h_1)|h_1) - \log p(a(h_1)|h_2) \quad (6.2)$$

where $p(a(h_i)|h_j)$ represents the likelihood that HMM h_j observes all the sequences $a(h_i)$ generated by h_i . This distance measure is similar to the directed divergence used by Juang and Rabiner [68]. In their work, the likelihoods were computed over all the examples of the two units h_1 and h_2 in the training data, as depicted in Figure 6.2.

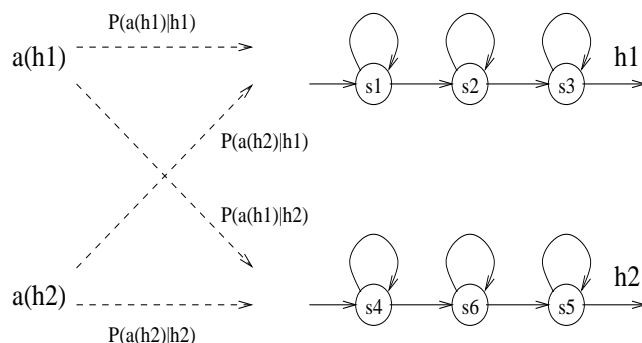


Figure 6.2: Computing $D(h_1, h_2)$ by evaluating likelihoods over all examples of h_1 and h_2 in the training data using both h_1 and h_2 .

Instead, we evaluate $D(h_1, h_2)$ efficiently using an analytical expression for $p(a(h_i)|h_j)$ which is based on an approximation that was recently developed by Printz and Olsen in the context of incorporating acoustic confusion into a language model [105]. This expression is equivalent to computing likelihood on segments synthesized by h_i (in place of examples from the training data), as depicted in Figure 6.3.

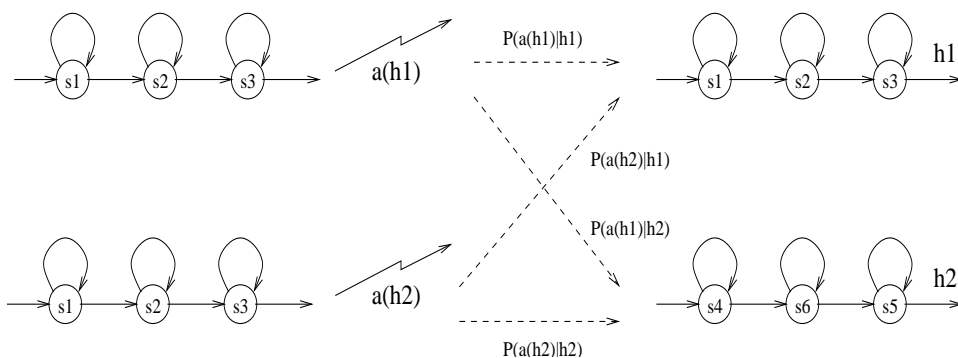


Figure 6.3: Computing $D(h_1, h_2)$ by evaluating the likelihood of all the sequences that can be synthesized with h_1 and h_2 .

For ease of notation, let $h_1 = w$ and $h_2 = v$. Then, $p(a(w)|v)$ can be written as:

$$p(a(w)|v) = \sum_{t=1}^{\infty} p(a_t(w)|v) \quad (6.3)$$

$$p(a_t(w)|v) = E_w [p(x_1^t|v)] \quad (6.4)$$

Here, the expectation is over all possible sequences x_1^t emitted by w . Let $\alpha_{w_i, v_j}(t)$ represent the likelihood of a sequence evaluated by v with terminal state j , and expectation is performed over all sequences generated by w with terminal state i . Then,

$$p(a_t(w)|v) = \sum_i E_{w, w_t=i} \left[\sum_j p(x_1^t, v_t = j|v) \right] \quad (6.5)$$

$$= \sum_i \sum_j \alpha_{w_i, v_j}(t) \quad (6.6)$$

where

$$\alpha_{w_i, v_j}(t) \triangleq E_{w, w_t=i} \left[\sum_j p(x_1^t, v_t = j|v) \right] \quad (6.7)$$

The likelihood $\alpha_{w_i, v_j}(t)$ can be written recursively in terms of the state observation densities and the transition probabilities w_{li} and v_{mj} , similar to the forward equation in HMM parameter estimation.

$$\alpha_{w_i, v_j}(t) = E_{w_t=i} [p(x_t|v_t = j)] \sum_{l=1}^i \sum_{m=1}^j \alpha_{w_l, v_m}(t-1) w_{li} v_{mj} \quad (6.8)$$

The likelihood $p(a(w)|v)$ in Equation 6.4 can be computed in a closed form, using the following approximation for the above recursion.

$$E_{w_t=i} [p(x_t|v_t = j)] \approx \exp(E_{w_t=i} [\log p(x_t|v_t = j)]) \quad (6.9)$$

$$= k(w_i, v_j) \quad (6.10)$$

The quantity $k(w_i, v_j)$ is the cross-entropy of two states w_i and v_j . All the quantities in the above equation depend only on the parameters of v and w , and the recursion in Equation 6.8 can be computed in terms of $k(w_i, v_j)$ and the transition probabilities, similar to forward algorithm over the the product space of the two HMMs shown in Figure 6.4. If we define a vector $\alpha(t)$ stacked with elements $\alpha_{w_i, v_j}(t)$ from the product space, the time update equation can be written as $\alpha(t+1) = M\alpha(t)$ where M is $|w||v| \times |w||v|$ matrix whose elements are a function of $k(w_i, v_j)$ and the transition probabilities of v and w . Equation 6.6 can then be written as:

$$p(a(w)|v) \approx u_F^T (I + M + M^2 + M^3 + \dots) u_I \quad (6.11)$$

$$= u_F^T (I - M)^{-1} u_I \quad (6.12)$$

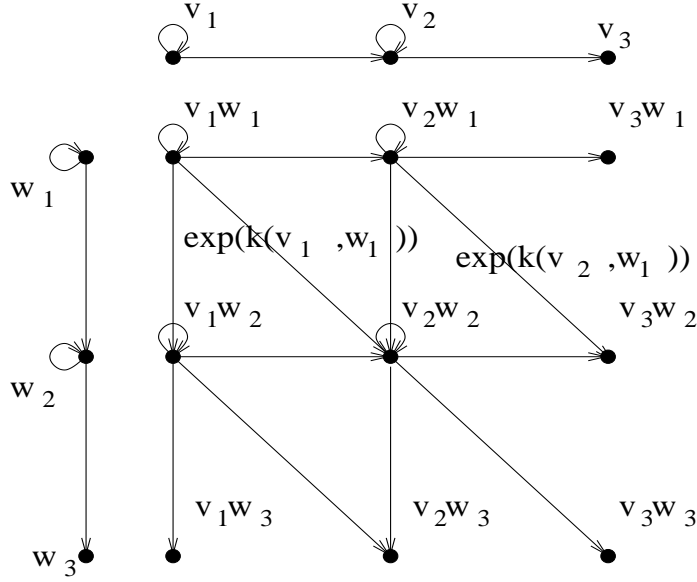


Figure 6.4: Representation of product space of two HMMs x and w for computing $p(a(w)|v)$.

Here, u_I and u_F are vectors with all elements zero except the product states corresponding to the starting states and the ending states of the two HMMs (which are assigned a unit value), respectively. By a linear transformation (stretching) of the observation space it is possible to ensure that each eigenvalue λ of M satisfies $|\lambda| < 1$, in which case $\sum_{i=0}^{\infty} M^i = (I - M)^{-1}$. The details of the derivation for $p(a(w)|v)$ can be found in [105].

In [105], the state observation densities are modeled as Gaussian mixture models and the cross-entropy $k(s_1, s_2)$ between two states s_1 and s_2 is computed using Monte-Carlo. Instead, by modeling state distributions with single Gaussians (as in decision tree clustering of state observation densities), we compute cross-entropy in terms of means and covariances of the Gaussians. For d -dimensional observation densities, $k(s_1, s_2)$ can then be evaluated in terms of its means and covariances (see Appendix B for details).

$$k(s_1, s_2) = -\frac{1}{2}(\mu_1 - \mu_2)^T \Sigma_2^{-1} (\mu_1 - \mu_2) - \frac{1}{2}(\Sigma_1 \Sigma_2^{-1}) - \frac{1}{2} \log((2\pi)^d |\Sigma_2|) \quad (6.13)$$

Using the closed form expression in Equation 6.12, the distance metric in Equation 6.2 can be evaluated. When the distance between two HMM paths are below an empirically

determined threshold, we prune the longer HMM path.

6.2.2 Mapping Contexts to HMMs

After the redundant paths for each unique context are removed using the above distance metric, a new Viterbi alignment is obtained. We can then associate a count (posterior probability over training data) of picking each path m_i for a particular context. Each of the path lengths can be considered to be a class, and a class-based distribution clustering can be performed to reduce the entropy in each context cluster [63]. The entropy $H(M|G)$ over all clusters g_j can be computed from the normalized counts of each path in each cluster $P(m_i|g_j)$ and is given by the following equation:

$$H(M|G) = - \sum_{g_j} P(g_j) \sum_{m_i \in M} P(m_i|g_j) \log P(m_i|g_j). \quad (6.14)$$

Here $P(g_j)$ is a prior term denoting the fraction of the whole population found in cluster g_j . Candidate partitions of a cluster are obtained by asking questions about triphone, syllable- and word-level features in the context, as in observation clustering. Every split is guaranteed to decrease the entropy, and the reduction can be computed locally at each node. In other words, we choose those questions that result in maximum gain in mutual information between the context and paths that are associated with it. The new clusters will tend to have counts aggregated in certain paths. We used minimum occupancy constraints on the leaf size, and a maximum tree size. At the leaf node, the paths with low probability $P(m_i|g_j)$ are pruned out. In actual implementation, if a context for which a particular path m_i was pruned out earlier in step (3) lands in a cluster with a non-zero path probability $P(m_i|g_j)$, mark m_i as unusable for that context and re-normalize the other probabilities. The resulting trees are evaluated on a held-out set. It was found that discarding contexts with very low occurrences (less than 5) improved path probabilities predicted by trees.

During decoding a context-dependent phoneme can be dropped down the path decision tree to obtain the HMM paths that are appropriate for it. Subsequently, the observation distribution for each state in those paths can be obtained from the usual state decision tree.

When more than one path is allowed for each leaf cluster, then all the path probabilities are normalized so that the highest path has a weight of one. The path probabilities are scaled by a factor to bring this in the range of acoustic likelihoods. Many researchers suggest these two tricks for pronunciation modeling whenever Viterbi algorithm is used for decoding [113].

6.3 Experiments and Results

To evaluate our algorithm we performed a series of experiments on the Switchboard corpus. We used a system developed in Chapter 4 as our baseline system, where 12000 gender-dependent clustered states were used, and each state was modeled using an 8-mixture Gaussian distribution, and 5-state HMMs with skips were used to model context specific phonemes. The error rate of the baseline system was 44.1%.

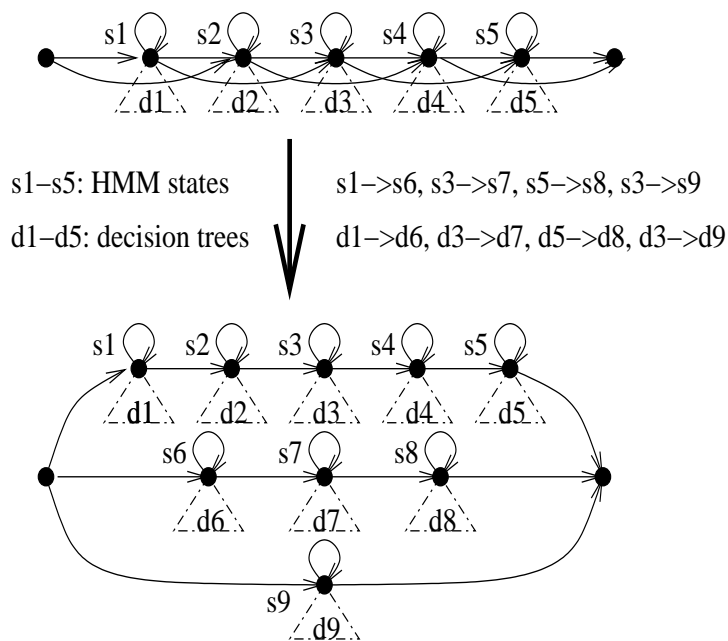


Figure 6.5: Initializing the parallel HMM paths from baseline fixed topology.

With the aim of increasing the number of paths gradually in a series of experiments, we started with a candidate set of three paths. Since most systems used either three or

five states, we included paths with these lengths. To pursue our aim of modeling phone reduction, we also included a one state path.

Initially, all contexts were allowed to have all the three paths. The five state paths were directly taken from the baseline system, and the skips were removed. The three state and the one state paths were initialized by cloning the states and the associated decision trees as illustrated in Figure 6.5. The states s_6 , s_7 , s_8 and s_9 were cloned from s_1 , s_3 , s_5 and s_3 , respectively along with their decision trees. Then, the models for each state was separately re-estimated using a few passes of the EM algorithm. The state decision trees were not redesigned. However, at the end of each EM iteration, the leaves in the state decision trees with low occupancy as well as mixture components of the state observation densities with low occupancy are pruned.

The likelihood gain on the training data starts diminishing a lot after three iterations as shown in Figure 6.6. The re-estimated models were then used to obtain a Viterbi alignment of the training data at the state level.

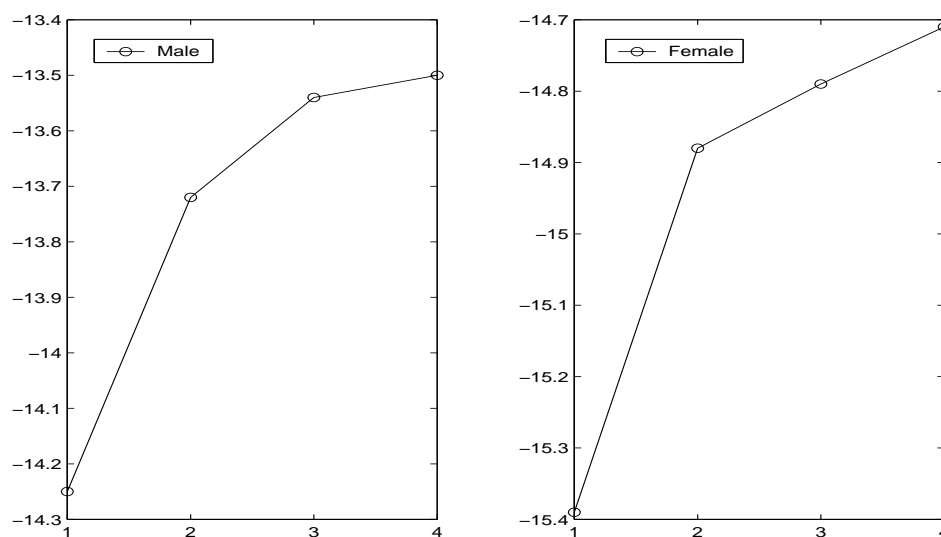


Figure 6.6: Average log-likelihood on the training data (male and female speakers) using all three HMM paths.

To examine whether there was any phone-specific preference for any of the paths, we computed the entropy of three different paths over all contexts c of each phoneme q using the posterior counts from the Viterbi alignment.

$$H(M|q) = \sum_{c \in q} P(c|q) \sum_{m_i=1,2,3} \log P(m_i|c) \quad (6.15)$$

The result is plotted in Figure 6.7. The broad patterns across gender were similar, and as expected a few phones such as EL, ER, OY, and SH showed marked preference for the 5-state path. When the baseline models were constrained to have 5-state HMM paths for these phonemes, the performance improved by a small amount (5 fewer substitutions) which was not significant. Decoding with all the paths increased word error rate to 45%, demonstrating the need for pruning the number of paths at each context (steps (3) and (4) in the algorithm).

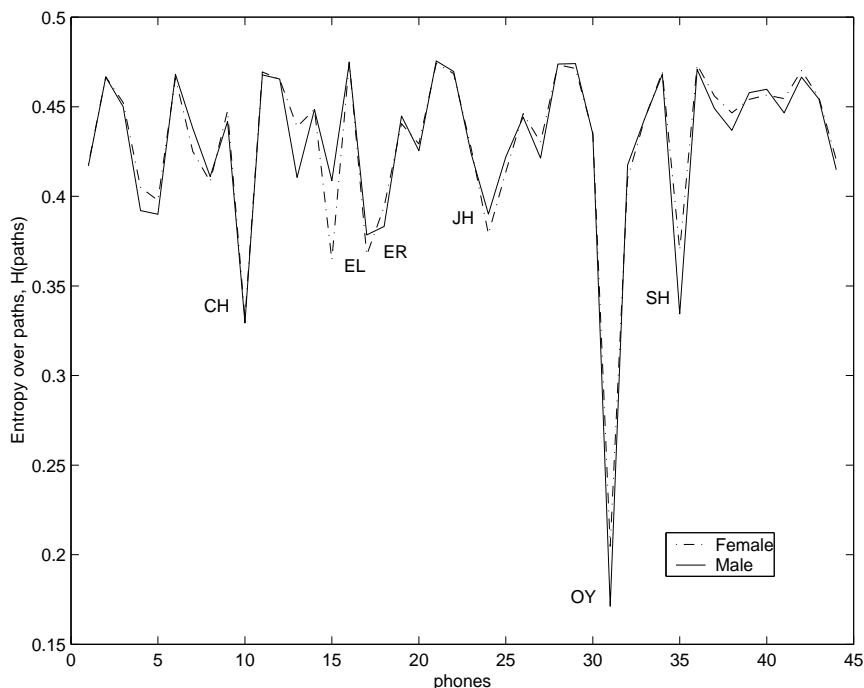


Figure 6.7: Average entropy of the distribution of path probabilities (male and female speakers).

Based on the new alignments, new state decision trees were grown across all the states to have the same number (12000) of clusters as the baseline system. The sharing between the states across the paths was found to be minimal (and mostly in (s1,s6) and (s4,s8)), so separate trees were grown for each phone state.

We used the second step in the algorithm and found that most HMM paths were different from each other. The cross entropy between the individual states across the paths was quite low for all phones. Among all pairs of states across the paths only 0.4% of them had cross-entropy of more than 0.00005, with no phone taking a share of more than 0.05%. In other words, the overlap of Gaussians state observation densities across the paths was very low, and so the HMMs in each path were very different from each other for all contexts. From the observation about low sharing of state densities across paths, this was not too surprising. So, step (3) in the algorithm was not needed in our case.

The probabilities $P(m_i|c)$ from the Viterbi counts for all context c were collected, and the path decision trees were grown by asking questions about the contextual features of c . It was found that syllable- and word-level features were used about 37% of time and affected about 14% of the data. Questions about stress appear to play a significant role in the path decision tree, and they are amongst the top six questions in a large number of phones across both genders.

Using the path decision trees, two sets of experiments were performed - one in which two paths out of three were kept in each leaf node, and another in which only the most likely path was kept. After the path assignments were performed, the observation densities were once again re-estimated using the EM algorithm. Decoding (second pass re-scoring) was performed using these models on the 1998 Switchboard development subset, and the results are tabulated in Table 6.1. The systems with multi-path HMMs performed poorly compared to the baseline system. In terms of gender, the degradation from (a) to (c) was solely from female speakers. For female speakers, substitution increased by about 0.1% and insertion by about 0.2% (and increasing the insertion penalty increased the total errors).

To understand the weakness of the multi-path HMMs, we asked the following questions.

Table 6.1: Recognition results on mapping context to HMM paths.

System	WER
(a) Baseline	44.1%
(b) Choosing 1 path	44.6%
(c) Choosing 2 paths	44.4%
(d) Choosing all paths	45.0%

Do these models match the acoustics of speech better than the baseline? Do certain words tend to be affected adversely by the multi-path HMMs?

We compared the log-likelihood of the baseline system and the systems with multi-path HMMs. In Table 6.2, the systems (a) and (b) use the same number of parameters, while the third one (c) uses more parameters. The first column shows the likelihood of the training data using the multi-path HMMs, and the second column shows the likelihood on a held out set. The likelihood of the multi-path HMMs over the training set and the held out set are better than that of the fixed topology. This suggests that they are modeling the acoustics of speech better than the baseline system for known word transcriptions.

Table 6.2: Average log-likelihood using different multi-path HMMs.

System	Training	Held-Out
a) Baseline	-14.73	-21.24
b) Choosing 2 path	-14.72	-21.06
c) Choosing all paths	-14.19	-20.52

We compared the word-level errors committed by the two systems on the test set. The errors were obtained from string alignment of the recognition output with the word transcripts of the test. We found that the errors committed by the two systems were clearly different. While some word confusions were mitigated, it increased others; a sampling is

tabulated in Table 6.3. Each column represents a particular type of error, and the numbers in braces indicate the instances of change (a positive value means higher for the new system). Here, we fail to observe any clear pattern.

Table 6.3: Change in word errors, computed with string alignment; negative counts represent improvement with the topology modeling.

Confusions	Insertions	Deletions
and → in (3)	oh (5)	that (4)
to → it (2)	of (2)	you (3)
as → and (2)	so (2)	but (3)
in → one (2)	her (2)	it's (3)
is → it (2)	what (2)	that's (2)
it → and (2)	said (2)	day (2)
great → right (2)	no (2)	tell (2)
⋮	⋮	⋮
know → no (-3)	all (-2)	as (-2)
the → that (-3)	if (-2)	what (-3)
that → and (-2)	is (-1)	of (-3)
now → know (-2)	have (-1)	one (-3)
you → to (-2)	it's (-1)	or (-3)
to → and (-2)	me (-1)	then (-3)
you → to (-2)	to (-1)	how (-3)

There is yet another open question regarding the multi-path HMMs. The skips in the baseline topologies allow the context-dependent phonemes to trace a number of paths with different numbers of states. On the other hand, the multi-path topologies constrain the number of paths to two at the most. It is possible that this loss in degrees of freedom is hurting the multi-path models. To test this hypothesis, we retained the baseline topologies

for the majority of context-dependent phonemes. When the path probabilities were above a threshold of 0.9 for 5-state path in a cluster, the skips were removed. Similarly, 1-state paths were allowed only above this threshold. The recognition results, reported in Table 6.4, show that there is a small improvement for female speakers, but it is not statistically significant. This gain may further improve on re-estimation.

Table 6.4: Recognition results from a hybrid approach.

System	Male WER	Female WER	Avg. WER
Baseline	42.3 %	45.7%	44.1%
Hybrid	42.3 %	45.6%	44.1%

6.4 Summary

In this chapter, we developed a method for mapping context-dependent phonemes to highly constrained HMM topologies. This technique is aimed at using higher level knowledge such as syllable- and word-level structure to hypothesize context-specific topologies, particular for phone reduction. Potentially, a variety of configurations and features (such as part-of-speech) could be explored using this approach. Not all of these have been tried here.

Based on the fact that likelihood increases on a held-out data set when the word transcriptions are given, we can conclude that the multi-path HMMs models speech better than the baseline model with fixed topology. However, this does not easily translate into recognition gain. Additionally, conditioning multi-path HMMs on part-of-speech may help improve recognition. Function words are likely to need more paths than content words. Our analysis suggests that the loss of degrees of freedom in HMM paths also contributes to performance degradation.

Further, the baseline topologies were augmented with a one-state path to represent reduction. This brought about a small improvement, which is not statistically significant.

One explanation could be that the states in a context-dependent triphones corresponding to a reduced phone may actually be modeling the edges of neighboring phonemes. Recent studies by Jurafsky et al. find that vowel reduction is sufficiently modeled by triphones [69]. They compared the phones alignments obtained from canonical lexicon using triphones with the surface form in ICSI data. They find that syllable deletions are not modeled by triphones. Perhaps the techniques developed in this chapter applied at a syllable level may tackle syllable reductions better than phone reductions.

Chapter 7

CONCLUSIONS AND FUTURE DIRECTIONS

The currently popular framework for recognizing conversational speech is based on a statistical paradigm, where the task of recognizing speech is broken up into two levels - a model that scores how likely a word sequence is in a language, and a model that scores how likely a particular observation sequence is given a word sequence. In a large vocabulary system, the second score is obtained by decomposing the model for a word in terms of phone-sized sub-word units. Phonetic knowledge as provided by a dictionary is used to learn characteristics of a phonetic context, and to generalize to rare and unseen words. Currently, this generalization is achieved through the use of a fixed HMM topology and mapping states to observation distributions using the phonetic context. In this thesis, we examined methods of using a wider context with high-level linguistic features such as syllable- and word-level structure to improve both the state observation distribution and the HMM topology of the sub-word units.

7.1 Contributions and Conclusions

7.1.1 Modeling Contextual Variation with Syllable and Word Features

We investigated the impact of syllable- and word-level features in modeling contextual variation through state distribution clustering. Previous studies such as [78, 102] were carried out on read speech, and few results were reported on conversational speech. Unlike the work in [52], we used a rich set of syllabic features, more training data, and possibly a better alignment. For the same number of parameters, we found that the syllable and word-level features give significantly better recognition accuracy on large vocabulary conversational speech than distributions based on triphone clustering. These models perform at least as

well as pentaphones, and the computational cost incurred in training them is much lower than a pentaphone system, by more than a factor of four. In decoding speech, the new models are faster than pentaphones for two reasons. Triphones coded with syllable- and word-level features occupy smaller state space than pentaphones on the same task. Conditioning on syllable- and word-level features produces observation densities which are tighter than pentaphones of the same complexity as inferred from increase in likelihood over a held-out test. Another advantage of these new models is that they can be easily incorporated into first pass decoding of current triphone-based recognizers without significant changes to the current software.

From the automatically trained decision trees, we learned that the position of phone in a word and the position of phone in a syllable impacts the contextual variation more than factors such as lexical stress, position of syllable in a word, and the phoneme contexts. These findings are consistent with previous studies on small data sets [49, 38, 51, 109]. When the number of questions defined on the contexts are taken into account, the impact of the new features, in terms of acoustic data affected, is disproportionately higher than phoneme context.

Coding of syllable-level structure in the lexicon and in acoustic modeling has been criticized for not taking re-syllabification into account. We developed a method to model re-syllabifications explicitly. Using this method, we could locate a large fraction of re-syllabifications automatically. It is likely that remaining instances are implicitly accounted for by the wide-context of the acoustic units. Instances where coda becomes ambisyllabic may have been missed by our system because of the acoustic similarity. We found no impact on recognition performance from explicit modeling of re-syllabification.

7.1.2 Multi-stage Clustering for Incorporating Wide Contexts

We developed a novel clustering mechanism to alleviate the adverse impact of conditioning state distributions on wide contexts. It gives exponential savings in the computational cost and memory required for clustering. The savings depend on the number of stages, the

number of features clustered in each stage, and the size of intermediate trees. It has a few advantages by design. Since only a subset of features is clustered in each stage, the atomic statistics (associated with each label) are more reliable. At each stage, all the data is pooled in the root node, this reduces data fragmentation to a certain extent. The multi-stage clustering was verified using a pentaphone system on the Switchboard task, and was found to be effective. Likelihood on a held-out data set was used to define subsets for the two-stages used in our experiment.

7.1.3 Modeling Temporal Variations with Flexible Topology

Finally, in Chapter 6, we developed an algorithm for mapping context to specific HMM topologies, and thus provide a framework for incorporating syllable- and word-level features to learn sub-word structures. A number of settings are possible, and we explored a few.

Our experiments reveal that multi-path HMMs, where context-dependent phonemes are mapped to one or two paths, can model conversational speech better than phonemes with fixed topology, when the word transcriptions are given. Lexical stress plays an important role in deciding the paths of the multi-path HMMs. However, the improvement in modeling phonemes does not easily translate into improvement in recognition performance.

One deficiency of the configuration we explored was that the longer HMMs paths were devoid of skips. This was intended to impose tighter constraints on phoneme models, and reduce confusion. However, this is not reflected in the results. Perhaps, removing the 3-state constraint on the phoneme increased confusion more dramatically than any possible gains.

7.2 Implications and Future Directions

The results reported in this thesis could potentially impact the lower layers of an automatic speech recognition system in different frameworks. The improvements in both recognition accuracy and likelihood on a held-out set obtained from clustering state distributions with syllable- and word-level has direct implications for two different approaches. First, these improvements were obtained even without considering asynchronous acoustics events. It

is more than likely that the impact of these features on asynchronous articulatory events, which are closely tied with phonology, will be considerably higher (e.g. the modeling framework in [94]). Second, these models capture a gradient of acoustic variation that cannot be easily modeled through intermediate layers. Hence, it can complement the efforts in pronunciation modeling with articulatory features or modeling sub-lexical constraints using other frameworks (e.g. as in [77]). In such a case, it may also be useful to take re-syllabification into account with the scheme that we developed, which is computationally very cheap.

Current methods for adapting acoustic models to account for speaker variation (both in training and testing) do not take syllable structure into account. The syllable- and word-level effects on acoustic may be more consistent in a speaker than across speakers (e.g. speaker-dependent domain-initial strengthening on “n” observed in [72]). If so, adaptation on these new models may give additional gains.

The multi-stage clustering algorithm that we developed in this thesis, paves the way for incorporating a large number of contextual features, which may include part-of-speech, hypothesized prosodic structures, and quantized speaking rate. Already, in a pilot experiment on about 3.5 hours of data hand-labeled with prosodic features, we found that prominence and break indices influence the observation space of current recognition systems, even when duration and F0 are not taken explicitly taken into account. The data was labeled with two levels of prominence and six levels of break indices according to ToBI conventions, and clustered along with triphone and word-position features. We found that on the average the questions about prosodic features were preferred over phonetic and word features about 14% of the times.

An interesting extension of multi-stage clustering would be to learn the subsets of features appropriate for each stage in an efficient manner, using measures such as mutual information. Instead of defining questions on all subtrees of previous stages, a minimal subset, whose logical combinations covers all subtrees, can be obtained using formal concept analysis [125], and this could potentially reduce the computation cost further. Instead of trying to capture the interaction between the conditioning variables within one decision tree,

another approach would be to grow different decision trees for each subset of features and capture the interaction through their product state space, with approximations employed in a factorial HMM [42]. As a framework for incorporating high-level knowledge, it may also be interesting to compare traditional decision trees with mixture of experts [65].

In designing HMM topologies, a few extensions of our multi-path HMM framework may be worth exploring. Apart from syllable- and word-level features, the temporal variation in the realization of a phone may be affected by part-of-speech, prosodic cues, and may also be speaker dependent. Being large in number, sloppiness in pronunciation of function words could potentially wipe out certain systematic temporal variations that may be present in content words. We have already seen that lexical stress plays an important role in the path decision trees. Hypothesized break indices and prominence may also explain certain systematic temporal variations.

The underlying theme in this thesis has been to improve phone-sized units, which allow maximum amount of sharing between sub-word units. An immediate extension of designing context-specific HMM topologies, would be to use these units in our multi-path framework to compose HMM topologies on-the-fly for bigger units such as frequent words or syllables. This would provide benefits similar to that observed in modeling bigger units, and at the same time avoid the pitfalls, as observed in [31], that arise from poor sharing of parameters between dissimilar units (such as triphones and syllables) which are necessary to cover a large vocabulary.

The underlying parametric distributions used for modeling sub-word units may not be close to their true distribution. As a result, improving these units under maximum likelihood framework may not translate into recognition performance. Use of a discriminative framework in designing the HMM topologies for each context-dependent sub-word unit may bring more benefits to the recognition performance.

BIBLIOGRAPHY

- [1] Theodore W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, New York, 1984.
- [2] Michiel Bacchiani. *The ASSM toolkit*. Speech Processing and Interpretation Laboratory Report (unpublished), 1998.
- [3] Michiel Bacchiani and Mari Ostendorf. Joint lexicon, acoustic unit inventory and model design. *Speech Communication*, 29(2-4):99–114, 1999.
- [4] L. R. Bahl, P. F. Brown, P. V. deSouza, and R. L. Mercer. A tree-based statistical language model for natural language speech recognition. *IEEE Trans. on Acoustics, Speech, and Signal Processing (ICASSP)*, 36(7):1001–1008, 1989.
- [5] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, and M. A. Picheny. Decision trees for phonological rules in continuous speech. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, pages 185–188, 1991.
- [6] L. E. Baum and J. A. Eagon. An inequality with applications to statistical prediction for functions of Markov process and to model of ecology. *Bull. American Math. Soc.*, 73:360–363, 1967.
- [7] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique in the statistical analysis of probabilistic functions of finite state Markov chain. *Annals of Mathematical Statistics*, 41:164–171, 1970.
- [8] Jeff Bilmes. Buried Markov Models for speech recognition. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, pages 2105–2108, 1999.

- [9] Jeff Bilmes. Natural statistical models for automatic speech recognition. *Ph.D. Thesis, International Computer Science Institute*, 1999.
- [10] Jeff Bilmes. Dynamic Bayesian networks. *The 16th Conference on Uncertainty in Artificial Intelligence*, July 2000.
- [11] Jeff Bilmes. Factored sparse inverse covariance matrices. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2:1009–1011, 2000.
- [12] Antonio Bonafonte, Rafael Estany, and Eugenio Vives. Study of subword units for Spanish speech recognition. *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1607–1610, 1995.
- [13] Herve Bourlard and Nelson Morgan. *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Press, 1993.
- [14] Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. *Classification and regression trees*. Wadsworth Inc., Belmont, CA, 1984.
- [15] Wray Buntine. *Learning classification trees*. Chapman and Hall, London, 1993.
- [16] William Byrne. Generalization and maximum likelihood from small data sets. *Proc. of IEEE-SP Workshop on Neural Networks for Signal Processing*, 1996.
- [17] William Byrne, Michael Finke, Sanjeev Khudanpur, John McDonough, Harriet Nock, Michael Riley, Murrat Saraclar, Chuck Wooters, and George Zavaliagkos. Pronunciation modeling for conversational speech recognition: a status report from WS97. *IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings (ASRU)*, pages 26–33, 1997.
- [18] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4), 2000.

- [19] Grace Chung. Towards multi-domain speech understanding with flexible and dynamic vocabulary. *PhD. Thesis, Massachusetts Institute of Technology*, 2001.
- [20] Kenneth W. Church. *Phonotactic Constraints*. Kluwer Academic Publishers, 1987.
- [21] Arthur C. Clarke. Sentinel of eternity. *Ten Story Fantasy*, 1951.
- [22] Arthur C. Clarke and Stanley Kubrick. *2001: A Space Odyssey, 25th Anniversary Edition*. N. A. L., Dutton, 1993.
- [23] Philip Clarkson and Roni Rosenfeld. Statistical language modeling toolkit. <http://www-svr.eng.cam.ac.uk/prc14/toolkit.html>, 1997.
- [24] George N. Clements. The role of the sonority cycle in core syllabification. *Papers in Laboratory Phonology I: Between the Grammar and Physics of Speech (John Kingston and Mary E. Beckman eds.)*, pages 283–333, 1990.
- [25] Thomas Colthurst, Owen Kimball, Fred Richardson, Han Shu, Chuck Wooters, Rukmini Iyer, and Herbert Gish. The 2000 BBN Byblos LVCSR system. *Proc. Int'l Conf. on Spoken Language Processing (ICSLP)*, 2:1007–1010, 2000.
- [26] I. Csisz'ar and G. Tusn'ady. Information geometry and alternating minimization procedures. *Statistics and Decisions, Supplementary Issue*, pages 205–237, 1984.
- [27] Sabine Deligne and Frederic Bimbot. Inference of variable-length acoustic units for continuous speech recognition. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 3:1731–1734, 1997.
- [28] John R. Deller, John H. L. Hansen, and John G. Proakis. *Discrete-time processing of speech signals*. IEEE Press, 2000.

- [29] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.
- [30] Vassilis Digalakis, Sid Berkowitz, Enrico Bocchieri, Costas Boulis, William Byrne, Heather Collier, Adrian Corduneanu, Ashvin Kannan, Sanjeev Khudanpur, and Ananth Sankar. Rapid speech recognizer adaptation to new speakers. *Report at JHU Summer Workshop on Speech Recognition*, 1998.
- [31] George Doddington, Andres Corrada, and Barbara Wheatley et al. Syllable based speech processing. *Report at JHU Summer Workshop on Speech Recognition*, 1997.
- [32] Stephane Dupont and Herve Bourlard. Using multiple time scale in a multi-stream speech recognition system. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 3:1767–1770, 1997.
- [33] Ellen Eide. Automatic modeling of pronunciation variation. *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, 1:451–454, 1999.
- [34] Michael Finke and Ivica Rogina. Wide context acoustic modeling in read vs. spontaneous speech. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, pages 1743–1746, 1997.
- [35] Jonathan Fiscus. A post processing system to yield reduced word error rate: recognizer output voting error reduction (ROVER). *IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings (ASRU)*, pages 347–354, 1997.
- [36] William Fisher. A C implementation of Daniel Kahn's theory of English syllable structure. *ftp://jaguar.ncsl.nist.gov/pub/tsylb2-1.1.tar.Z*, 1996.

- [37] Malcolm R. Forester. Key concepts in model selection: performance and generalizability. *Journal of Mathematical Psychology*, 44:205–231, 2000.
- [38] Eric Fosler-Lussier, Steven Greenberg, and Nelson Morgan. Incorporating contextual phonetics into automatic speech recognition. *Proc. ESCA Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, pages 611–614, 1999.
- [39] Ramana Gadde, Elizabeth Shriberg, Andreas Stolcke, Dilek Hakani-Tur, and Gokhan Tur. Prosody modeling for speech recognition and understanding. *Proc. Hub-5 Conversational Speech Understanding Workshop*, 1999.
- [40] Mark J. F. Gales. Semi-tied covariance matrices. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2:657–660, 1998.
- [41] Zoubin Ghahramani and Geoffrey Hinton. The EM algorithm for mixtures of factor analyzers. *Technical Report CRG-TR-96-1, Dept. of Computer Science, University of Toronto*, 1997.
- [42] Zoubin Ghahramani and Michael Jordan. Factorial Hidden Markov Models. *Machine Learning*, 29:245–273, 1997.
- [43] Herbert Gish and Kenney Ng. A segmental speech model with application to word spotting. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2:447–450, 1993.
- [44] John Godfrey, Edward Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 1:517–520, 1992.
- [45] P. S. Gopalkrishnan, Dimitri Kanvesky, Arthur Nadas, and David Nahamoo. An

- inequality for rational functions with applications to some statistical estimation problem. *IEEE Trans. on Information Theory*, pages 107–113, 1991.
- [46] Ramesh A. Gopinath, Bhuvana Ramabhadran, and Satya Dharanipragada. Factor analysis invariant to linear transformation of data. *Proc. Int'l Conf. on Spoken Language Processing (ICSLP)*, 5:2223–26, 1998.
- [47] Steven Greenberg. On the origins of speech intelligibility in the real world. *Proc. ESCA Workshop in Robust Speech Recognition for Unknown Communication Channels*, pages 23–32, 1996.
- [48] Steven Greenberg. Understanding speech understanding - towards a unified theory of speech perception. *Proc. ESCA Tutorials and Advanced Research Workshop on the Auditory Basis of Speech Preception*, pages 1–8, 1996.
- [49] Steven Greenberg. Speaking in shorthand - a syllable-centric perspective for understanding pronunciation variation. *Proc. Workshop Modeling Pronunciation Variation in Automatic Speech Recognition*, pages 47–56, 1998.
- [50] Steven Greenberg and Shawn Chang. Linguistic dissection of Switchboard-corpus automatic speech recognition systems. *Proc. ISCA ITRW Workshop on Automatic Speech Recognition: Challenges for the new Millenium*, 2000.
- [51] Steven Greenberg and Eric Fosler-Lussier. The uninvited guest: information's role in guiding the production of spontaneous speech. *Proceedings of the Crest Workshop on Models of Speech Production: Motor Planning and Articulatory Modeling, Germany*, 2000.
- [52] Asela Gunawardana. Using stress and syllabification features in state clustering for Switchboard acoustic models. *Unpublished Technical Report*, 1998.

- [53] Thomas Hain and Phil C. Woodland. Dynamic HMM selection for continuous speech recognition. *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, 2:1327–1330, 1999.
- [54] Thomas Hain and Phil C. Woodland. Modelling sub-phone insertions and deletions in continuous speech recognition. *Proc. Int'l Conf. on Spoken Language Processing (ICSLP)*, 4:172–176, 2000.
- [55] Thomas Hain, Philip Woodland, Gunnar Evermann, and D. Povey. The CU-HTK March 2000 Hub5e transcription system. *Proc. Speech Transcription Workshop*, 2000.
- [56] Jonathan Hamaker, Aravind Ganapathiraju, Joseph Picone, and John Godfrey. Advances in Alphadigit recognition using syllables. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 1:421–424, 1998.
- [57] Michael Hammond. *The Phonology of English - A Prosodic Optimality-Theoretic Approach*. Oxford Linguistics, Oxford University Press, 1999.
- [58] Toshiyuki Hanazawa, Jun Ishii, Yohei Okato, and Kunio Nakajima. Acoustic modeling for spontaneous speech recognition using syllable dependent models. *Proc. Int'l Conf. on Spoken Language Processing (ICSLP)*, 4:157–160, 2000.
- [59] Hynek Hermansky and Nelson Morgan. RASTA processing of speech. *IEEE Trans. on Speech and Audio Processing*, 2(4):578–589, 1994.
- [60] Zhihong Hu, Johan Schalkwyk, Etienne Barnard, and Ronald Cole. Speech recognition using syllable like units. *Proc. Int'l Conf. on Spoken Language Processing (ICSLP)*, 2:426–429, 1996.
- [61] Rukmini Iyer, Owen Kimball, and Herb Gish. Modeling trajectories in the HMM

- framework. *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, 1:479–482, 1999.
- [62] Rukmini Iyer and Mari Ostendorf. Transforming out-of-domain estimates to improve in-domain language models. *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, 4:1975–1978, 1997.
- [63] Frederick Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997.
- [64] Rhys J. Jones, Simon Downey, and John S. Mason. Continuous speech recognition using syllables. *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, 3:1171–1174, 1997.
- [65] Michael Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.
- [66] Biing-Hwang Juang, Wu Chou, and Chin-Hui Lee. Minimum classification error rate methods for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 5(3):257–265, 1997.
- [67] Biing-Hwang Juang and Lawrence Rabiner. The segmental k-means algorithm for estimating parameters of hidden Markov models. *IEEE Trans. on Acoustics, Speech, and Signal Processing (ICASSP)*, 38(9):1639–1641, 1990.
- [68] Biing-Hwang Juang and Lawrence R. Rabiner. A probabilistic distance measure for Hidden Markov models. *AT&T Technical Journal*, 64(2):391–408, 1985.
- [69] Dan Jurafsky, Wayne Ward, Zhang Jiaping, Keith Herold, Yu Xiuyang, and Zhang Sen. What kind of pronunciation variation is hard for triphones to model? *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2001.

- [70] Ashvin Kannan, Mari Ostendorf, and Robin Rohlicek. Maximum likelihood clustering of Gaussians for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 2(3):453–455, 1994.
- [71] Patricia Keating. Word-level phonetic variation in large speech corpora. *ZAS Papers in Linguistics 11 (A. Alexiadou et al. eds.)*, pages 35–50, 1998.
- [72] Patricia Keating, Taehong Cho, Cecile Fougeron, and Chai-Shune Hsu. Domain-initial articulatory strengthening in four languages. *Laboratory Phonology*, 6, 1998.
- [73] Patricia Keating, Richard Wright, and Jintao Zhang. Word-level asymmetries in consonant articulation. *UCLA Working Papers in Phonetics*, 97, 1999.
- [74] Michael Kenstowicz. The syllable and syllabification. *Phonology in Generative Grammar*, pages 250–309, 1994.
- [75] Katrin Kirchhoff. Syllable-level desynchronisation of phonetic features for speech recognition. *Proc. Int'l Conf. on Spoken Language Processing (ICSLP)*, 4:2274–2276, 1996.
- [76] Ron Kohavi. Bottom-up induction of oblivious read-once decision graphs. *Proc. European Conf. on Machine Learning*, pages 154–169, 1994.
- [77] Raymond Lau. Subword lexical modelling for speech recognition. *Ph.D. thesis, Massachusetts Institute of Technology*, 1998.
- [78] Clark Z. Lee and Douglas O'Shaughnessy. Clustering beyond phoneme contexts for speech recognition. *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, 1:19–22, 1997.
- [79] Li Lee and Richard C. Rose. A frequency warping approach to speaker normalization. *IEEE Trans. on Speech and Audio Processing*, 6(1):49–60, 1998.

- [80] Sung-Chien Lin, Lee-Feng Chien, Keh-Jiann Chen, and Lin-Shan Lee. A syllable-based very-large-vocabulary voice retrieval system for Chinese databases with textual attributes. *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, 1:203–206, 1995.
- [81] Richard P. Lippmann. Review of neural networks for speech recognition. *Neural Computation*, 1:1–38, 1989.
- [82] Philip Lockwood and Marc Blanchet. An algorithm for the dynamic inference of Hidden Markov models. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2:251–254, 1993.
- [83] Xiaoqiang Luo and Frederick Jelinek. Probabilistic classification of HMM states for large vocabulary continuous speech recognition. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, pages 2044–2047, 1999.
- [84] Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding consensus among words: lattice-based word error minimization. *Proc. European Conference on Speech Communication and Technology*, pages 495–498, 1999.
- [85] Jose B. Marino, Albino Nogueiras, and Antonio Bonafonte. The demiphone : an efficient sub-word unit for continuous speech recognition. *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, 3:1212–1218, 1997.
- [86] Dominic W. Massaro. Perceptual images, processing time, and perceptual units in auditory perception. *Psychological Review*, 79:124–145, 1972.
- [87] Shoichi Matsunaga and Takeshi Matsumura. Non-uniform unit based HMMs for continuous speech recognition. *Speech Communication*, pages 321–329, 1995.

- [88] Don McAllaster, Larry Gillick, Francesco Scattone, and Mike Newman. Studies with fabricated Switchboard data: exploring sources of model-data mismatch. *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998.
- [89] Mehryar Mohri, Michael Riley, and Fernando Pereira. Weighted finite-state transducers in speech recognition. *Proc. Automated Speech Recognition: Challenges for the Next Millennium*, August 2000.
- [90] Nelson Morgan and Herve Bourlard. An introduction to hybrid HMM / connectionist continuous speech recognition. *IEEE Signal Processing Magazine*, pages 25–42, May 1995.
- [91] Sreerama K. Murthy. Automatic construction of decision trees from data: a multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2(4):345–389, 1998.
- [92] Herman Ney and Stefan Ortmanns. Dynamic programming search for continuous speech recognition. *IEEE Signal Processing Magazine*, 16(5):64–83, 1999.
- [93] Hermann Ney, Ute Essen, and Reinhard Kneser. On structuring probabilistic dependences in stochastic language modeling. *Computer Speech and Language*, 8:1–38, 1994.
- [94] Harriet Nock and Steve Young. Loosely coupled HMMs for ASR. *Proc. Int'l Conf. on Spoken Language Processing (ICSLP)*, 3:143–146, 2000.
- [95] Yves Normandin and S. D. Morgan. An improved MMIE training algorithm for speaker-independent small vocabulary continuous speech recognition. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, pages 537–540, 1991.
- [96] Arlindo L. Oliveira. Inferring reduced ordered decision graphs of minimum description length. *Proc. 12th Int'l Conf. on Machine Learning*, pages 421–429, 1995.

- [97] Mari Ostendorf, William Byrne, and Michiel Bacchiani et al. Modeling systematic variation in pronunciation via language-dependent hidden speaking mode. *Report at JHU Summer Workshop on Speech Recognition*, 1996.
- [98] Mari Ostendorf, Vassilis Digalakis, and Owen A. Kimball. From HMM's to segment models: A unified view to stochastic modeling for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 4:360–379, sept 1996.
- [99] Mari Ostendorf and Harold Singer. HMM topology design using maximum likelihood successive state splitting. *Computer Speech and Language*, 11:17–42, 1997.
- [100] David S. Pallett, Jonathan G. Fiscus, John S. Garofolo, Alvin Martin, and Mark Przybocki. 1998 Broadcast News benchmark test results: English and non-English word error rate performance measures. *Proc. DARPA Broadcast News Workshop*, pages 5–12, 1999.
- [101] Douglas B. Paul. An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, pages 25–28, 1992.
- [102] Douglas B. Paul. Extensions to phone-state decision tree clustering: single tree and tagged clustering. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2:1487–1990, 1997.
- [103] Thilo Pfau, M. Beham, and Guenther Ruske. Creating large sub-word units for speech reconstruction. *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1191–1194, 1997.
- [104] D. Povey and Phil C. Woodland. Frame discrimination training of HMMs for large vocabulary speech recognition. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, pages 333–336, 1999.

- [105] Harry Printz and Peder Olsen. Theory and practice of acoustic confusability. *Proc. ISCA ITRW Workshop on Automatic Speech Recognition: Challenges for the new Millenium*, pages 77–84, 2000.
- [106] D. Pye and Phil C. Woodland. Experiments in speaker normalization and adaptation for large vocabulary speech recognition. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2:1047–1050, 1997.
- [107] John R. Quinlan. Probabilistic decision trees. *Machine Learning: An Artificial Intelligence Approach - Volume 3 (R.S.Michalski and Y. Kodratoff eds.)*, 1990.
- [108] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Signal processing series. Prentice Hall, 1993.
- [109] Mark A. Randolph. Syllable-based constraints on properties of English sounds. *Ph.D. Thesis, Massachusetts Institute of Technology*, 1989.
- [110] Wolfgang Reichl and Wu Cho. A unified approach of incorporating general features in decision tree based acoustic modeling. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2:573–576, 1999.
- [111] Steve Renals, Nelson Morgan, Herve Bourlard, Michael Cohen, and Horacio Franco. Connectionist probability estimators in HMM speech recognition. *IEEE Trans. on Speech and Audio Processing*, 2(1):161–174, 1994.
- [112] Michael Riley. A statistical model for generating pronunciation networks. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, pages 737–740, 1991.
- [113] Michael Riley, William Byrne, Michael Finke, Sanjeev Khudanpur, Andre Lolje, John McDonough, Harriet Nock, Murrat Saraclar, Chuck Wooters, and George Zavaliagos.

- Stochastic pronunciation modeling from hand-labeled phonetic corpora. *Speech Communication*, 29:209–224, 1999.
- [114] Ananth Sankar. A new look at HMM parameter for large vocabulary conversational speech. *Proc. Int'l Conf. on Spoken Language Processing (ICSLP)*, 5:2219–22, 1998.
- [115] Murat Saraclar, Harriet Nock, and Sanjeev Khudanpur. Pronunciation modeling by sharing Gaussian densities across phonetic models. *Computer Speech and Language*, 14(2):137–160, 2000.
- [116] Lawrence Saul and Michael Jordan. Mixed memory Markov models. *Machine Learning*, 37:75–87, 1999.
- [117] Lawrence Saul and Mazin Rahim. Markov processes on curves for automatic speech recognition. *Advances in Neural Information Processing Systems (M. S. Kearns, S. A. Solla, and D. A. Cohn eds.)*, 11, 1999.
- [118] Ralf Schlüter, Boris Mueller, Frank Wessel, and Hermann Ney. Interdependence of language models and discriminative training. *IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings (ASRU)*, pages 119–122, 1999.
- [119] Padhraic Smyth. Clustering sequences with Hidden Markov Models. *Advances in Neural Information Processing Systems*, 9:648–655, 1997.
- [120] Andreas Stolcke, Yochai Konig, and Mitchel Weintraub. Explicit word error minimization in N-best list rescoring. *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 163–166, 1997.
- [121] Andreas Stolcke and Stephen Omohundro. Grammatical inference and applications. *Inducing Probabilistic Grammars by Bayesian Model Merging*, pages 106–118, 1994.

- [122] Valtcho Valtchev, Julian Odell, Phil C. Woodland, and Steve J. Young. MMIE training of large vocabulary recognition systems. *Speech Communication*, pages 303–314, 1997.
- [123] Steve Wegmann, Don McAllaster, J. Orloff, and Barbara Peskin. Speaker normalization on conversational telephone speech. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 1:339–342, 1996.
- [124] Mitch Weintraub, Kelsey Taussig, Kate Hunicke-Smith, and Amy Snodgrass. Effect of speaking style on LVCSR performance. *Int'l Conf. on Speech and Language Processing supplement*, pages 16–19, 1996.
- [125] D. Willett, C. Neukirchen, J. Rottland, and G. Rigoll. Refining tree-based state clustering by means of formal concept analysis, balanced decision trees and automatically generated model-sets. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2:565–568, 1999.
- [126] Su-Lin Wu. Incorporating information from syllable-length time scales into automatic speech recognition. *PhD. Thesis, International Computer Science Institute*, 1998.
- [127] Su-Lin Wu, Brian Kingsbury, Nelson Morgan, and Steven Greenberg. Incorporating information from syllable length time scales into automatic speech recognition. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2:721–724, 1998.
- [128] Su-Lin Wu, Michael L. Shire, Steven Greenberg, and Nelson Morgan. Integrating syllable boundary information into speech recognition. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2:987–990, 1997.
- [129] Steve J. Young. A review of large-vocabulary continuous-speech recognition. *IEEE Signal Processing Magazine*, pages 45–57, 1996.

- [130] Steve J. Young and Phil C. Woodland. State clustering in HMM-based continuous speech recognition. *Computer Speech and Language*, 8:369–384, 1994.
- [131] George Zavaliagkos, John McDonough, David Miller, Amro El-Jaroudi, and et al. The BBN Byblos 1997 large vocabulary conversational speech recognition system. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2:905–908, 1998.
- [132] Jing Zheng, Horacio Franco, F. Weng, Ananth Sankar, and H. Bratt. Word-level rate of speech modeling using rate-specific phones and pronunciations. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 3:1775–1778, 2000.
- [133] Walter Zucchini. An introduction to model selection. *Journal of Mathematical Psychology*, 44:41–61, 2000.
- [134] Geoffery Zweig and Mukund Padmanabhan. Exact alpha-beta computation in logarithm space with application to map word graph construction. *Proc. Int'l Conf. on Acoustic, Speech and Signal Processing*, 2:855–858, 2000.
- [135] Geoffrey Zweig and Stuart J. Russell. Speech recognition with dynamic Bayesian networks. *Proc. National Conference on Artificial Intelligence/Conference on Innovative Applications of Artificial Intelligence*, pages 173–180, 1998.

Appendix A

QUESTIONS ON SYLLABLE- AND WORD-LEVEL FEATURES

Table A.1: Questions about position of phone (syllable) in word, asked on the phone and its immediate neighbors

Position of phone (syllable) in word	
first	(0,3) vs. (1,2)
last	(0,1) vs. (2,3)
first but not last	(0) vs. (1,2,3)
middle	(1) vs. (0,2,3)
last but not first	(2) vs. (0,1,3)
only one	(3) vs. (0,1,2)

Table A.2: Questions about lexical stress, asked on the phone and its immediate neighbors

Lexical stress	
unstressed	(0) vs. (1,2)
primary stress	(1) vs. (0,2)

Table A.3: Questions about phone in syllable

On center phone	
in onset	(0,1,6) vs. (2,3,4,5)
in onset, not ambisyllabic	(0,1) vs. (2,3,4,5,6)
syllable-initial consonant	(0,6) vs. (1,2,3,4,5)
in coda	(0,1,2) vs. (3,4,5,6)
in coda, not onset	(0,1,2,4) vs. (3,5,6)
possibly syllable-final	(0,1,2,4) vs. (3,5,6)
5 may follow another 5 or 6	
syllable-final, not ambisyllabic	(0,1,2,4,6) vs. (3,5)
singleton questions	(0) vs. (1,2,3,4,5,6)
:	:
On left phone	
center phone is C: non-initial in onset cluster	(0,1,6) vs. (2,3,4,5)
center phone is V: syllable-initial	
center phone is C, post-vocalic	(2) vs. (0,1,3,4,5,6)
center phone is V, V-V syllable onset	
consonant non-initial in coda cluster (never have V or onset C after 4)	(4) vs. (0,1,2,3,5,6)
On right phone	
center phone is V: open syllable	(0,2) vs. (1,3,4,5,6)
center phone is V: open syllable before C	(0) vs (1,2,3,4,5,6)
center phone is C: syllable-final before heterosyllabic C	
center phone is V: open syllable before V	(2) vs. (0,1,3,4,5,6)
center phone is C: pre-vocalic	
if right phone is in onset-other, then in onset consonant cluster	(1) vs. (0,2,3,4,5,6)

Appendix B

**CROSS-ENTROPY BETWEEN SINGLE GAUSSIAN
DISTRIBUTIONS**

$$\begin{aligned}
k(s_1, s_2) &= \int P(x|s_1) \log P(x|s_2) dx \\
P(x|s_2) &= \frac{1}{((2\pi)^d |\Sigma_2|)^{1/2}} \exp -\frac{1}{2}(x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) \\
\log P(x|s_2) &= -\frac{1}{2} \left[(x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) + \log((2\pi)^d |\Sigma_2|) \right] \\
k(s_1, s_2) &= E [\log P(x|s_2) | s_1] \\
&= -\frac{1}{2} E \left[x^T \Sigma_2^{-1} x - x^T \Sigma_2^{-1} \mu_2 - \mu_2^T \Sigma_2^{-1} x + \mu_2^T \Sigma_2^{-1} \mu_2 + \log((2\pi)^d |\Sigma_2|) \right] \\
&= -\frac{1}{2} E \left[\text{tr}(x^T x \Sigma_2^{-1}) - x^T \Sigma_2^{-1} \mu_2 - \mu_2^T \Sigma_2^{-1} x + \mu_2^T \Sigma_2^{-1} \mu_2 + \log((2\pi)^d |\Sigma_2|) \right] \\
&= -\frac{1}{2} \left[\text{tr}(E[x^T x] \Sigma_2^{-1}) - \mu_1^T \Sigma_2^{-1} \mu_2 - \mu_2^T \Sigma_2^{-1} \mu_1 + \mu_2^T \Sigma_2^{-1} \mu_2 + \log((2\pi)^d |\Sigma_2|) \right] \\
&= -\frac{1}{2} \left[\text{tr}((\Sigma_1 + \mu_1^T \mu) \Sigma_2^{-1}) - \mu_1^T \Sigma_2^{-1} \mu_2 - \mu_2^T \Sigma_2^{-1} \mu_1 + \mu_2^T \Sigma_2^{-1} \mu_2 + \log((2\pi)^d |\Sigma_2|) \right] \\
&= -\frac{1}{2} \left[\text{tr}(\Sigma_1 \Sigma_2^{-1}) + \mu_1^T \Sigma_2^{-1} \mu - \mu_1^T \Sigma_2^{-1} \mu_2 - \mu_2^T \Sigma_2^{-1} \mu_1 + \mu_2^T \Sigma_2^{-1} \mu_2 + \log((2\pi)^d |\Sigma_2|) \right] \\
&= -\frac{1}{2} \left[\text{tr}(\Sigma_1 \Sigma_2^{-1}) + (\mu_1 - \mu_2)^T \Sigma_2^{-1} (\mu_1 - \mu_2) + \log((2\pi)^d |\Sigma_2|) \right]
\end{aligned}$$