Applications of Broad Class Knowledge for Noise Robust Speech Recognition

by

Tara N. Sainath

B.S., Massachusetts Institute of Technology (2004)
M. Eng., Massachusetts Institute of Technology (2005)

Submitted to the Department of
Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author .....
Department of
Electrical Engineering and Computer Science
May 21, 2009

Certified by .....
Victor W. Zue
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Terry P. Orlando
Chairman, Department Committee on Graduate Theses

Applications of Broad Class Knowledge for Noise Robust Speech Recognition

by

Tara N. Sainath

Submitted to the Department of
Electrical Engineering and Computer Science
on May 21, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This thesis introduces a novel technique for noise robust speech recognition by first describing a speech signal through a set of broad speech units, and then conducting a more detailed analysis from these broad classes. These classes are formed by grouping together parts of the acoustic signal that have similar temporal and spectral characteristics, and therefore have much less variability than typical sub-word units used in speech recognition (i.e., phonemes, acoustic units). We explore broad classes formed along phonetic and acoustic dimensions.

This thesis first introduces an instantaneous adaptation technique to robustly recognize broad classes in the input signal. Given an initial set of broad class models and input speech data, we explore a gradient steepness metric using the Extended Baum-Welch (EBW) transformations to explain how much these initial model must be adapted to fit the target data. We incorporate this gradient metric into a Hidden Markov Model (HMM) framework for broad class recognition and illustrate that this metric allows for a simple and effective adaptation technique which does not suffer from issues such as data scarcity and computational intensity that affect other adaptation methods such as Maximum a-Posteriori (MAP), Maximum Likelihood Linear Regression (MLLR) and feature-space Maximum Likelihood Linear Regression (fMLLR). Broad class recognition experiments indicate that the EBW gradient metric method outperforms the standard likelihood technique, both when initial models are adapted via MLLR and without adaptation.

Next, we explore utilizing broad class knowledge as a pre-processor for segment-based speech recognition systems, which have been observed to be quite sensitive to noise. The experiments are conducted with the SUMMIT segment-based speech recognizer, which detects landmarks - representing possible transitions between phonemes - from large energy changes in the acoustic signal. These landmarks are often poorly detected in noisy conditions. We investigate using the transitions between broad classes, which typically occur at areas of large acoustic change in the audio signal, to aid in landmark detection. We also explore broad classes motivated along both acous-

tic and phonetic dimensions. Phonetic recognition experiments indicate that utilizing either phonetically or acoustically motivated broad classes offers significant recognition improvements compared to the baseline landmark method in both stationary and non-stationary noise conditions.

Finally, this thesis investigates using broad class knowledge for island-driven search. Reliable regions of a speech signal, known as islands, carry most information in the signal compared to unreliable regions, known as gaps. Most speech recognizers do not differentiate between island and gap regions during search and as a result most of the search computation is spent in unreliable regions. Island-driven search addresses this problem by first identifying islands in the speech signal and directing the search outwards from these islands. In this thesis, we develop a technique to identify islands from broad classes which have been confidently identified from the input signal. We explore a technique to prune the search space given island/gap knowledge. Finally, to further limit the amount of computation in unreliable regions, we investigate scoring less detailed broad class models in gap regions and more detailed phonetic models in island regions. Experiments on both small and large scale vocabulary tasks indicate that the island-driven search strategy results in an improvement in recognition accuracy and computation time.

Thesis Supervisor: Victor W. Zue
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to express my sincere gratitude to my advisor, Victor Zue. Victor inspired me to pursue a PhD right from my undergraduate days and I am thankful for the many opportunities he has provided. He has kept me organized and focused, and every discussion with him makes me a better researcher. His encouragement, patience and support have helped guide me through the research process and this thesis.

In addition, I am thankful to the members of my thesis committee, Jim Glass, Ken Stevens and Bill Freeman, whose comments greatly helped to improve this thesis. Jim has been an invaluable mentor over the last five years in helping to shape and continuously advance my research. Discussions with Ken helped improve the island-driven search aspect of this thesis and reminded me to keep a larger audience in mind. Finally, Bill encouraged me to think about the similarity of techniques presented in this thesis to other fields of computer science.

I am grateful to the staff of the Spoken Language Systems Group (SLS). T.J. Hazen was my masters thesis advisor, and helped introduce me various aspects of speech recognition research, as well as many tools and techniques used in the field. Discussions with Lee Hetherington about speech recognition search and FSTs were extremely helpful in shaping the search experiments in this thesis. Stephanie Seneff has always offered valuable suggestions on my research, whether in group meetings, papers for conferences, and this thesis. Thank you to Scott Cyphers for helping with various software related issues. Chao Wang helped in creating the CSAIL-info corpus, and answered many questions about her lexical stress and pitch-tracking research. Meetings with Victor were always smoothly organized thanks to Colleen Russell, not to mention all the entertaining discussions about the Celtics we had. And finally, thank you to Marcia Davidson for handling many logistical issues and always being there to make me laugh.

My summer internships with the speech group at IBM Research helped introduce me to state of the art research in speech recognition. Thank you to the research staff members at IBM, including Bhuvana Ramabhadran, Dimitri Kanevsky, Brian

Kingsbury, Michael Picheny and David Nahamoo. I am thrilled to soon be joining such a stimulating research environment.

The research in this thesis would not have been possible without the generous support of various sponsors, including the Quanta-MIT T-Party Project and the Agile Robotics for Logistics Project.

It was always a joy to come to the office every day, largely in part to my exciting officemates. Thank you to Alex Gruenstein, Harr Chen and Ali Mohammed for all the fun, laughter, and frequent water/coffee breaks. In addition, students in the SLS group have helped to create an enjoyable research atmosphere, including Hung-An Chang, Ghinwa Choueiter, Alex Gruenstein, Paul Hsu, Karen Livescu, John Lee, Ian McGraw, Alex Park, Mitch Peabody, Ken Schutte, Han Shu and Yaodong Zhang. Also thank you to the undergrad UROPs I have had the pleasure to interact with, namely Sean Liu, Kevin Luu and Steven Wu.

To all the friends I have made at MIT during my undergraduate and graduate years, thank you for making life outside of school so rewardable and enjoyable.

Thank you to Premal for his constant encouragement, for pushing me to finish this thesis, and for all the support throughout the years.

I would certainly not be where I am without the love and care of my family. My aunt and uncle have been like second parents to me, and are always there for guidance. My cousins have been wonderful role models for me since the day I was born. And thank you for having such great husbands and adorable children who are a constant source of entertainment. My sister is an amazing companion and is always there to listen and talk. And finally words cannot express thanks to my parents, whose constant love, support, and endless sacrifices have made so many wonderful opportunities in my life possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent years, improvements in speech recognition systems have resulted in high performance for certain tasks under clean conditions. For example, digit recognition can be performed with a word error rate of less than 0.3% [67]. In addition, a less than 1% error rate has been achieved on a speaker-independent isolated word recognition task with a 20,000 word vocabulary [19]. The performance of speech recognition systems, however, rapidly degrades in noisy environments. For example, the accuracy of a speech recognizer in a clean speech environment can drop by over 30% when the same input speech is corrupted by the noise that is present over long-distance telephone lines [66].

While the performance of speech recognition systems can degrade in noisy environments, human performance is much more robust. For example, [64] compares the performance of humans and machines on over 100 utterances from the *Wall Street Journal* task [55], with automobile noise artificially added at four different signal-to-noise ratios (SNRs) (i.e., Clean, 22dB, 16dB and 10dB). The word error rate for machines exceeds 40% at SNRs of 10dB and 16dB. However, human error rate remains at around 1% in all four noise conditions. In addition, [92] compares human and machine performance for isolated digits at five different SNRs ranging between 18dB and -6dB, in 6dB increments. 6 different noise types from the Noisex-92 database [93] are

explored, varying in their stationarity and harmonicity properties. The study indicates that human error rate is less than 2% across all noise types. However, machine performance degrades rapidly, and reaches an error rate of almost 100% at 0dB.

The degradation of speech recognition systems in noisy conditions can be explained by various phenomena [49]. First, additive noise can alter the speech signal and corresponding feature vectors used by speech recognizers to represent this signal. Second, reverberations from the recording environment as well as the recording microphone itself can also distort the speech signal. Third, changes in articulation caused by adverse conditions, known as the Lombard effect[1], can also have a profound effect on the signal [72].

To date, it has not been possible to develop a universally successful and robust speech recognition system in the presence of background noise. Systems which perform well in one scenario can seriously degrade in performance under a different set of environmental conditions. The increased focus on natural human-to-computer interaction has placed greater emphasis on moving speech recognition performance closer to human level, particularly in noisy conditions. In addition, with the increased availability and popularity of mobile information devices, the interest in noise robust system performance has also grown, since speech-based interactions are more likely to be conducted in a wide variety of noise-corrupted environments. Numerous techniques have been studied to improve the robustness of speech systems under noisy conditions. These techniques can be divided into four main categories based on their focus, namely noise resistant features, speech and feature enhancement, noise adaptation [34], and multi-modal information [54].

Noise resistant feature methods attempt to use features which are less sensitive to noise and distortion [34]. These methods focus on identifying better speech recognition features or estimating robust features in the presence of noise. Perhaps the most popular technique is cepstral mean normalization (CMN) [4], which involves subtracting the mean of the cepstral feature vector [47], typically calculated across

[1]The Lombard effect is a phenomenon in which speakers raise their vocal intensity in the presence of noise.

subsections of the utterance, from each frame in the corresponding section, to reduce the effect of channel disturbances. Auditory-inspired features are also a popular example of noise robust features. For example, perceptual linear predictive (PLP) features [42] and wavelet features [97] have both been shown to offer improvements in noisy conditions. Techniques such as relative spectral processing (RASTA) [43] attempt to remove noises which vary more slowly compared to the variations in a speech signal. While many of the above techniques make neither assumptions nor estimations about noise characteristics, this is sometimes a disadvantage since these techniques may be far from optimal in certain noise conditions. For example, in a babble noise environment where the noise characteristics are similar to those of speech, RASTA processing could potentially be ineffective.

Speech enhancement techniques attempt to suppress the impact of noise on speech by extracting out clean speech or feature vectors from a contaminated signal. Spectral subtraction [9] methods subtract noise from the speech signal with the assumption that noise characteristics are slowly varying and uncorrelated with the signal. Parameter mapping techniques [27] attempt to transform noisy speech into clean speech, and typically do not make any assumptions about noise characteristics. Finally, Bayesian estimation methods [18] attempt to estimate a clean speech vector by minimizing a cost function, usually the mean squared error (MSE) between noisy speech and clean speech. Many speech enhancement techniques were originally developed to improve speech quality for human listeners. Thus, while many of the algorithms have been shown to enhance the quality of speech to the human listener, the deformation of the signal induced by some methods does not always lead to improvements in speech recognition.

Instead of deriving an estimate of clean speech, *noise adaptation* techniques attempt to adapt recognition models to noisy environments. This includes, for example, changes to the recognizer formulation, such as changing model parameters of the recognizer to accommodate noisy speech. Parallel Model Combination [26] is one such method for compensating model parameters under noisy conditions in a computationally efficient manner. In addition, some techniques also explore designing noise

models within the recognizer itself. While this technique performs well at high SNRs, at low SNRs compensated model parameters often show large variances, resulting in a rapid degradation of performance.

Finally, *multi-modal information* techniques use multiple sources of information about the speech signal, such as different sets of temporal features, articulatory features or audio-visual information, in conjunction with standard acoustic representations. For example [54] shows the benefits of using articulatory features in addition to standard speech recognition features in adverse environments. Furthermore, [39] shows the benefit of incorporating both audio and visual cues in noisy speech, rather than just utilizing audio cues.

Many of the noise robust techniques discussed apply general pattern recognition and statistical learning techniques to improve noise robustness without incorporating speech-specific knowledge. Instead, they focus solely on the noise type and signal-to-noise ratio when adapting to a specific environmental condition. The limited utilization of speech knowledge is partly due to the fact that the most commonly used sub-word unit representation for speech knowledge, phonemes [24], is subject to a high degree of variability in noisy conditions.

This thesis explores the use of speech knowledge for robust speech recognition by first describing a speech signal through a set of broad speech units, and then conducting a more detailed analysis from these broad classes. These classes are formed by grouping together parts of the acoustic signal that have similar temporal and spectral characteristics, and therefore have much less variability than the underlying sub-word units. Typically, these broad classes can be formed along phonetic or acoustic dimensions.

Broad classes, which are phonetically motivated, are created by grouping together underlying phonemes into a set of broad phonetic classes (BPCs), for example vowels, nasals, stops, fricatives and closures. Linguists have agreed on a pre-defined mapping between phonemes and a corresponding set of BPCs [14]. An example of broad classes learned from phonetic units is displayed in Figure 1-1. This figure shows a speech time-frequency representation, known as a spectrogram, of the spoken word

"nine". The phonemes corresponding to the word, namely /n/, /ay/ and /n/ are also indicated. Finally, the mapping from these phonetic units to a set of broad classes, namely nasal (*nas*) and vowel (*vow*), is shown on the last line. For further details regarding these phonetic and broad class representations, refer to Appendix B.

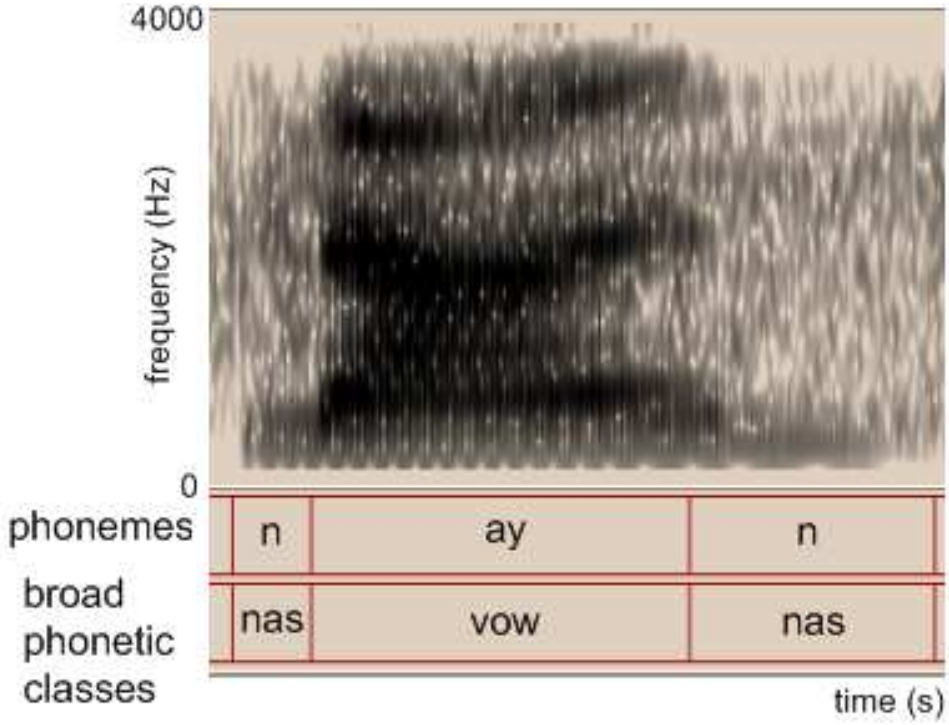

Figure 1-1: Speech spectrogram of the word "nine". The corresponding phonemes (/n/, /ay/, /n/) as well as the set of broad phonetic classes (*nas*, *vow*, *nas*) are also delineated.

Broad classes can also be motivated along acoustic dimensions, by using acoustical characteristics to group the signal into a set of broad acoustic classes (BACs). An example of a set of broad classes learned from the acoustic signal is illustrated in Figure 1-2. The diagram illustrates a spectrogram of the spoken word "zero". The phonemes corresponding to the word (i.e., /z/, /ih/, /r/, /ow/), as well as the learned BACs (i.e., *bac1*, *bac2*, *bac3*) are also displayed in the figure. Notice that when broad classes are motivated by acoustics, the number of learned BACs does not always correspond to the number of phonetic units. For example, /r/ and /ow/ are grouped into one broad class, namely *bac3*.

This thesis explores using broad classes for robust speech recognition because these

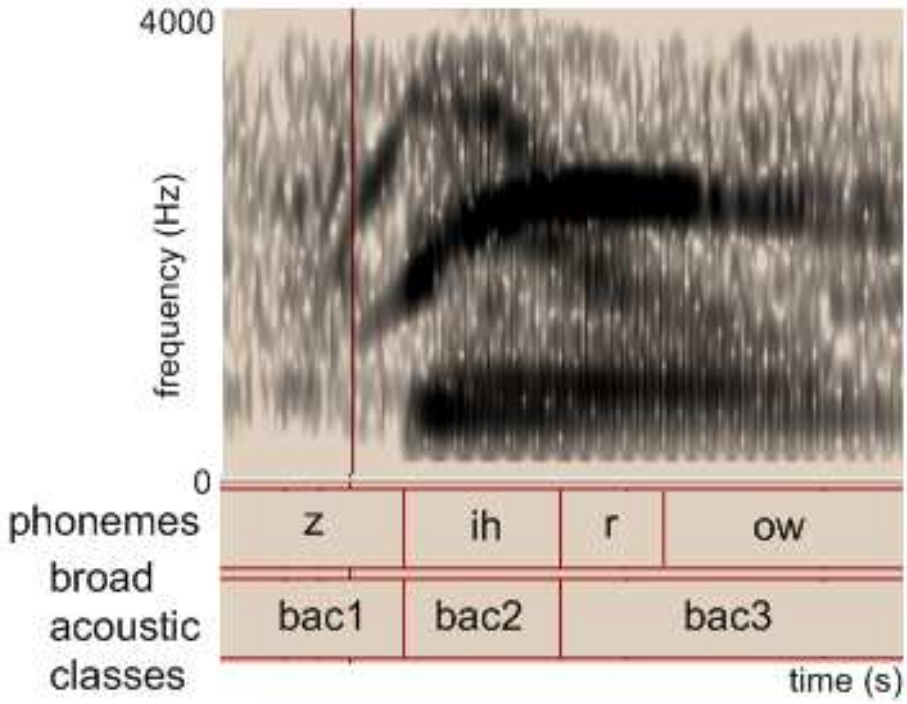

Figure 1-2: Speech spectrogram of the word "zero". The corresponding phonemes (/z/, /ih/, /r/, /ow/) as well as the set of learned broad acoustic classes (*bac1*, *bac2*, *bac3*) are also delineated.

classes have many important characteristics which make them attractive. First, broad classes are prominently visible in speech spectrograms, as discussed in [102] and further indicated in Figures 1-1 and 1-2. In addition, [37] and [65] further demonstrate through experimental studies the salience of broad classes by showing that most of the confusions between English phonemes occur in the same broad class. Second, because the broad classes are formed by pooling together different sub-word units (i.e., phonemes or acoustic units), there is more training data available compared to the full set of sub-word units, allowing for better model representation and robustness. Third, generally sub-words which belong to the same manner/articulatory class convey similar spectral and temporal properties and can be categorized as belonging to the same broad class, while sub-word units in different broad classes are acoustically distinct. Grouping together sub-word units into broad classes, which behave differently in noise, provides the advantage of applying distinct class-specific methods to each broad class. For example, certain broad classes corresponding to high-energy,

voiced parts of the speech signal, are more reliably identified in noisy conditions, so a detailed analysis of reliable parts of the spectra can help to fill in information about unreliable sections [16]. Fourth, [11] suggests the possible language-independence of broad classes by illustrating that various languages use the lexical space in a similar fashion when represented by a set of broad classes. While the experiments in this thesis are explored only in English, the use of broad classes allows for the possibly of exploring the proposed techniques across multiple languages.

The rest of this chapter is organized as follows. In Section 1.2, an overview of previous work on utilizing broad class knowledge in speech recognition is provided. In Section 1.3 the main contributions of this thesis are discussed. Finally in Section 1.4, the structure of the thesis is outlined.

1.2 Previous Work

The use of broad classes has been explored extensively for many tasks in speech recognition. One of the most popular uses of broad classes is for lexical access. For example, the Huttenlocher-Zue model [48] explores isolated word recognition by first characterizing a word by a broad class representation and using this partial description to retrieve a cohort of words. A detailed analysis is then performed on this cohort to determine the best word. In addition, [90] explores the Huttenlocher-Zue model for lexical access in continuous speech recognition.

Broad classes have also been utilized in designing mixture of expert classifiers. Both [38] and [85] investigate using expert classifiers specific to each broad phonetic class. Phonetic classification is then performed by combining scores from the different experts.

In addition, many acoustic modeling techniques investigate grouping together phonemes within broad classes during training. As discussed in [100], during context-dependent acoustic model training, enough training data is often unavailable to accurately train each context-dependent model. Thus, context-dependent phones which fall into the same broad class are often merged together and the aggregate of their

data is used to train the models. In addition, [25] explores using broad classes for Maximum Likelihood Linear Regression (MLLR) transformations. Again, because enough training data is often unavailable to estimate a transform for each phoneme, the authors explore applying the same transformation to all phonemes within the same broad class and show that this approach outperforms applying just one uniform transformation for all models.

Furthermore, broad classes have been used for language identification [41], [46]. Both works explore representing sentences in different languages by a set of broad phonetic strings, and demonstrate good performance for language identification by using a less detailed broad class analysis.

The above uses of broad classes in speech recognition reveal several underlying themes. First, the lexical access and mixture of experts research illustrate that broad classes can be utilized to conduct a less detailed but robust analysis of the signal, after which a more detailed analysis is performed with the broad class knowledge. Secondly, the acoustic modeling work demonstrates that broad classes allow for a natural grouping among sub-word units which behave similarly, while differentiating among those that behave differently. Finally, the language identification work shows that broad classes capture very robust and salient portions of the signal. It is these three main themes that we take advantage of in our utilization of broad class knowledge for robust speech recognition.

1.3 Contributions

In this thesis, the use of broad class knowledge as a pre-processor is explored for two noise robust speech recognition applications. The main contributions of the thesis are outlined in the following subsections.

1.3.1 Instantaneous Adaptation for Broad Class Recognition

First, we introduce an instantaneous adaptation technique to robustly recognize broad classes from the input signal. In general pattern recognition tasks, given some input

data and an initial model, a probabilistic likelihood score is often computed to measure how well the model describes the data. Typically the model is trained in a condition that is different from the target environment. While popular adaptation techniques, such as *Maximum a-Posteriori (MAP)* [28] and *Maximum Likelihood Linear Regression (MLLR)* [62] have been explored to adapt initial models to the target domain, these methods frequently require a few utterances of data in the target domain to perform the adaptation.

The Extended Baum-Welch (EBW) transformations [35] are one of a variety of discriminative training techniques ([75], [86]) that have been explored in the speech recognition community to estimate model parameters of Gaussian mixtures. Recently however, the EBW transformations have also been used to derive a gradient steepness measurement ([51], [52]) to explain model fit to data. More specifically, given an initial model and some input data, the gradient steepness measurement quantifies how much we have to adapt the initial model to explain the target data. The better the initial model fits the data, the less the initial model needs to be adapted and the flatter the gradient steepness. In addition, this gradient steepness measurement can be thought of as an instantaneous adaptation technique to explain model fit to data, since very little data is required to measure the gradient required to adapt the initial model.

We incorporate this instantaneous adaptation technique into a Hidden Markov Model (HMM) [76] framework for broad class recognition. We demonstrate the effectiveness of this gradient metric over scoring models using likelihood in a variety of noise environments, both when initial models are adapted using MLLR and without MLLR. We then utilize this broad class knowledge for two noise robust speech applications discussed in the next two subsections.

1.3.2 Utilization of Broad Class Knowledge For Landmark Detection

A segment-based framework for acoustic modeling ([31], [68]), which can also be formulated as a variable frame-rate HMM [101] has shown success in recognizing

speech in noise-free environments. For example, the SUMMIT speech recognizer developed at MIT has shown success in phonetic recognition tasks [37], as well as word recognition tasks such as in speech recorded over telephone lines [33] or in lecture halls [32]. However, we suspect that the performance of a segment-based system like SUMMIT may be sensitive to certain types of noise.

SUMMIT computes a temporal sequence of frame-based feature vectors from the speech signal, and performs landmark detection based on the spectral energy change of these feature vectors. These landmarks, representing possible transitions between phones, are then connected together to form a graph of possible segmentations of the utterance. This segment graph is then passed to a scoring and search phase to find the best set of hypothesized words. A block diagram of this segment-based system is shown in Figure 1-3. In this thesis, we refer to this segmentation algorithm as the *spectral change* segmentation method.

Figure 1-3: Block Diagram of the SUMMIT Segment-Based Speech Recognition System

While this spectral method works well in clean conditions ([31], [37]), the system has difficulty locating landmarks in noise and often produces poor segmentation hypotheses [80]. Thus, in this thesis, we explore broad class knowledge as a preprocessor in designing a robust landmark detection algorithm. More specifically, we take advantage of the fact that transitions between broad classes occur at areas of large acoustic change in the speech signal, even in the presence of noise. We utilize the locations of these transitions to aid in landmark detection. Once landmarks are detected, the segment graph is formed and scored using methods similar to the spectral method.

In addition, the use of phonetically *vs.* acoustically motivated broad classes is

also explored. While both phonetically and acoustically motivated representations have been explored in clean speech, little work has been done in comparing these representations under noisy conditions. Given different noise conditions, for example stationary *vs.* non-stationary or harmonic *vs.* non-harmonic, we suspect that sometimes a phonetic approach is preferred, while other times an acoustic approach might be preferred.

We demonstrate that using broad class knowledge as a pre-processor to aid in landmark detection offers significant improvements in noisy speech environments relative to the baseline spectral change method. In addition, we show under which noise conditions a phonetic *vs.* acoustic method is preferable.

1.3.3 Utilization of Broad Class Knowledge for Island-Driven Search

Finally, we explore broad classes to aid in island-driven search [12]. [3] and [88] hypothesize that human speech processing is done by first identifying "regions of reliability" in the speech signal and then filling in unreliable regions using a combination of contextual and stored phonological information. However, most current speech recognizers treat the reliability of information as uniformly distributed throughout the signal. Hence, many decoding paradigms consist of a left-to-right scoring and search component, and an optional right-to-left component, without utilizing knowledge of reliable speech regions. More specifically, speech systems often spend the bulk of their computation efforts in unreliable regions, when, in reality, most of the information in the signal can be extracted from the reliable regions [103]. In the case of noisy speech, if phrases are unintelligible, this may even lead the search astray and make it impossible to recover the correct answer [73]. Furthermore, this is particularly a problem in large vocabulary speech systems, where pruning is required to limit the size of the search space. Pruning algorithms generally do not make use of the reliability of portions of the speech signal, and hence may remove too many hypotheses in the unreliable regions of the speech signal and keep too many hypotheses in the

reliable regions [56].

Island-driven search [12] is an alternative method that may better handle noisy and unintelligible speech. This strategy works by first hypothesizing islands as regions in the signal which are reliable. Recognition then works outwards from these anchor points to hypothesize unreliable gap regions. While island-driven search has been explored for both parsing [17] and character recognition [73] there has been limited research (i.e., [56]) in applying these techniques to continuous speech recognition.

In this thesis, we explore utilizing information about reliable speech regions to develop a noise robust island-driven search strategy. First, we take advantage of the salience of broad classes to identify regions of reliability in the speech signal. Next, these island/gap regions are utilized to efficiently prune the search space and decrease the amount of computational effort spent in unreliable regions. Specifically we investigate pruning more aggressively in island regions and less aggressively in gap regions. However, unlike most confidence based pruning techniques [2], [23], the island regions are used to influence the pruning in gaps, which allows an increased number of hypotheses to be pruned away. This decreases the search space and increases the chances of going through reliable island regions.

Secondly, we investigate island information during final recognition. Specifically, to limit spending time unnecessarily in gap regions, less detailed models are scored in gap regions in the form of broad classes. In island regions, more detailed acoustic models are utilized. We demonstrate that taking advantage of island/gap knowledge, both for segment pruning and during final search, offers improvements in both recognition accuracy and computation time.

1.4 Overview

The remainder of this thesis is organized in the following manner. First, in Chapter 2, the various recognition frameworks and corpora used for experiments in this thesis are described. Next, Chapter 3 discusses the formulation of the Extended Baum-Welch Transformation gradient steepness metric which we apply to broad class recognition.

Chapter 4 compares broad phonetically *vs.* acoustically motivated broad classes in designing a robust landmark detection and segmentation algorithm, while Chapter 5 discusses using broad class knowledge in island-driven search. Finally, Chapter 6 concludes the thesis and discusses future work.

Chapter 2

Experimental Background

Given a set of acoustic observations $O = \{o_1, o_2, o_3, \ldots, o_n\}$ associated with a speech waveform, the goal of an automatic speech recognition (ASR) system is to find the corresponding sequence of words $\hat{W} = \{w_1 w_2 ... w_m\}$ which has the maximum a posteriori probability $P(W|O)$. This goal is expressed more formally by Equation 2.1.

$$\hat{W} = \arg\max_{W} P(W|O) \tag{2.1}$$

In most ASR systems, a sequence of sub-word units U and a sequence of sub-phone states S are also decoded along with the optimal word sequence W. These sub-word units can correspond to context-independent phones or context-dependent phones. Context-independent phones are modeled by just one phone. Context-dependent phones are modeled by multiple phones, for example as diphones (i.e., two phones), triphones (i.e., three phones), or quinphones (i.e., five phones). Taking into account the sub-phone states and sub-word units, Equation 2.1 can be rewritten as

$$\hat{W} = \arg\max_{W} \sum_{S} \sum_{U} P(W, U, S|O) \tag{2.2}$$

To simplify computation, most ASR systems also use dynamic programming (e.g., Viterbi [76]) or graph-searches (e.g., A^* [44]) to find a single optimal sub-phone sequence $\hat{S}$, along with an optimal sub-word sequence $\hat{U}$ and words $\hat{W}$. Equation 2.2

then simplifies to:

$$\hat{W}, \hat{U}, \hat{S} \approx \arg \max_{W,U,S} P(W, U, S|O) \tag{2.3}$$

Applying Bayes' rule to the above Equation gives:

$$P(W, U, S|O) = \frac{P(O|S, U, W)P(S|U, W)P(U|W)P(W)}{P(O)} \tag{2.4}$$

As presented in [68], the term $P(O|S, U, W)$ is known as the *feature observation model*; $P(S|U, W)$ is called the *model topology*; $P(U|W)$ is referred to as the *pronunciation model*; and $P(W)$ is the *language model*. Since $P(O)$ is constant for a given utterance and does not affect the outcome of the search, it is usually ignored.

The two most common model topologies in ASR systems include frame-based [76] and segment-based [31] systems. Since the ideas central to this thesis employ both segment-based and frame-based recognizers, the behavior of the four terms outlined in Equation 2.4 within both systems is discussed below.

2.1 Attila Speech Recognition System

The broad class pre-processor utilized in this thesis uses the frame-based Attila Speech Recognizer developed at IBM [74]. Below the four components in the Attila system are described in more detail.

2.1.1 Model Topology

In Attila, each sub-word unit $u_n \in U$ is represented by a Hidden Markov Model (HMM) [76]. The model topology for each sub-word unit u_n consists of a sequence of P sub-phone states $S = \{s_1, s_2 \ldots s_P\}$ which typically go from left-to-right. Sub-word units are usually modeled by 3 or 5 left-to-right HMM states. Figure 2-1 shows the model topology for a 3-state left-to-right HMM.

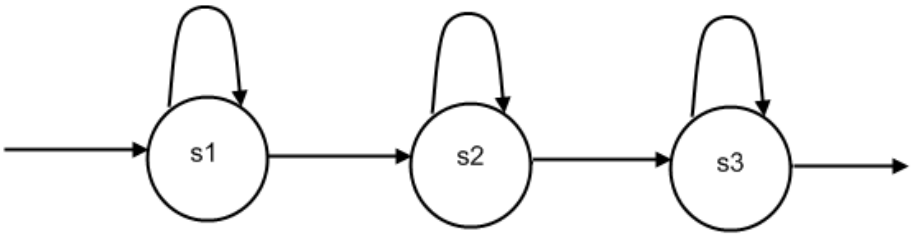

Figure 2-1: A 3-state left-to-right HMM. States s_1, s_2 and s_3 correspond to the three states.

2.1.2 Observation Model

In frame-based modeling, the acoustic observation space, O, consists of a temporal sequence of acoustic features (e.g., Mel-frequency cepstral coefficients (MFCCs) [20]) which are computed at a fixed-frame rate. Thus, the feature observation model computes the probability of each observation frame o_i given a particular state in the HMM s_k from sub-word model u_n. This observation model is typically represented by a Gaussian mixture model (GMM). Let us assume that GMM for state s_k has N Gaussian components, where each component j is parameterized by the following mean, covariance and weight parameters respectively $\lambda_j^k = \{\mu_j^k, \Sigma_j^k, w_j^k\}$. Thus, the probability of observation o_i given state s_k is expressed as

$$P(o_i|s_k) = \sum_{j=1}^{N} w_j^k P(o_i|\lambda_j^k) \tag{2.5}$$

Each component $P(o_i|\lambda_j^k)$ can be expressed by a Gaussian probability density function, as expressed in Equation 2.6, where d is the dimension of the observation vector o_i.

$$P(o_i|\lambda_j^k) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_j^k|^{\frac{1}{2}}} exp\left(-\frac{1}{2}(o_i - \mu_j^k)^T(\Sigma_j^k)^{-1}(o_i - \mu_j^k)\right) \tag{2.6}$$

2.1.3 Pronounciation/Lexical Model

$P(U|W)$ is the pronunciation or lexical model which gives the likelihood that a sequence of sub-word units, U, was generated from a given word sequence W. This is achieved by a lexical lookup. Each word in the lexicon may have multiple pronunciations to account for phonetic variability [40].

2.1.4 Language Model

The language model is denoted by $P(W)$. $P(W)$ represents the *a priori* probability of a particular word sequence $W = \{w_1, w_2, \ldots, w_m\}$. Attila typically uses an n-gram language model where the probability of each successive word depends only on the previous $n-1$ words, as shown by Equation 2.7.

$$P(W) = P(w_1, w_2, \ldots, w_m) = \prod_{i=1}^{m} P(w_i|w_1, \ldots, w_{i-1}) \approx \prod_{i=1}^{m} P(w_i|w_{i-(n-1)}, \ldots, w_{i-1}) \tag{2.7}$$

2.2 SUMMIT Speech Recognition System

The segment-based recognition experiments discussed in this thesis utilize the SUMMIT segment-based speech recognition system, developed at the Spoken Language Systems Group at MIT's Computer Science and Artificial Intelligence Laboratory. In this section we will briefly discuss the different components of the SUMMIT recognition system [31].

2.2.1 Model Topology

In segment-based modeling, frame-level feature vectors (e.g., MFCCs) are computed at regular time intervals. An additional processing stage in segment-based modeling then converts frame-level feature vectors to segmental feature vectors.

SUMMIT creates segmental feature vectors by first hypothesizing acoustic landmarks at regions of large change in the frame-level feature vectors. We can think of

these landmarks as representing hypothetical transitions between spectrally distinct events. More specifically, major landmarks are hypothesized at locations where the spectral change exceeds a specified global threshold. A fixed density of minor landmarks are detected between major landmarks where the spectral change, based on the fixed minor landmark density, exceeds a specified local threshold.

These acoustic landmarks are then connected together to specify a collection of possible segmentations S for the utterance. Since it is computationally expensive to search through this large segmentation network, an explicit segmentation phase is incorporated into the recognizer to reduce the size of the search space and the computation time of the recognizer. More specifically, all minor landmarks are fully interconnected between, but not across, major landmarks, to form a segment network representing possible segmentations of the speech utterance. In addition, each major landmark is connected to two major landmarks forward. In this thesis, we will refer to the segmentation algorithm just described as the *spectral change* segmentation method. Figure 2-2 shows a typical segment network formed from major and minor landmarks, and Figure 2-3 illustrates a graphical display of the segment network from SUMMIT.

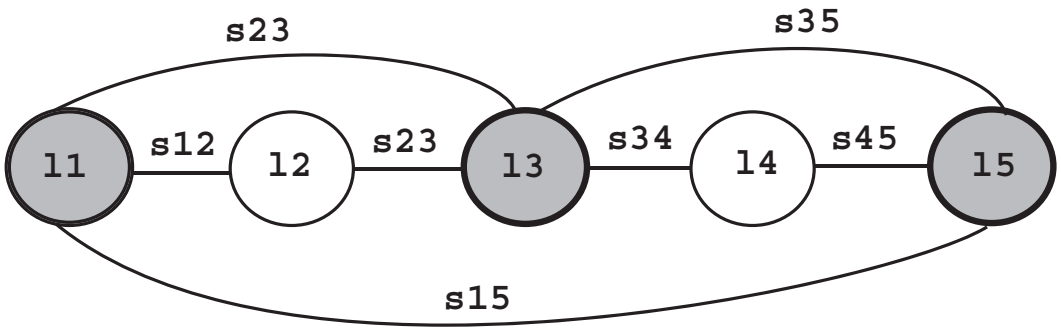

Figure 2-2: Segment network for the Spectral Change Segmentation technique. Major landmarks are indicated by shaded circles. Each minor landmark l_i between major landmarks is fully connected to every adjacent landmark l_j in the graph via segments s_{ij}. In addition, each major landmark is connected to two major landmarks forward.

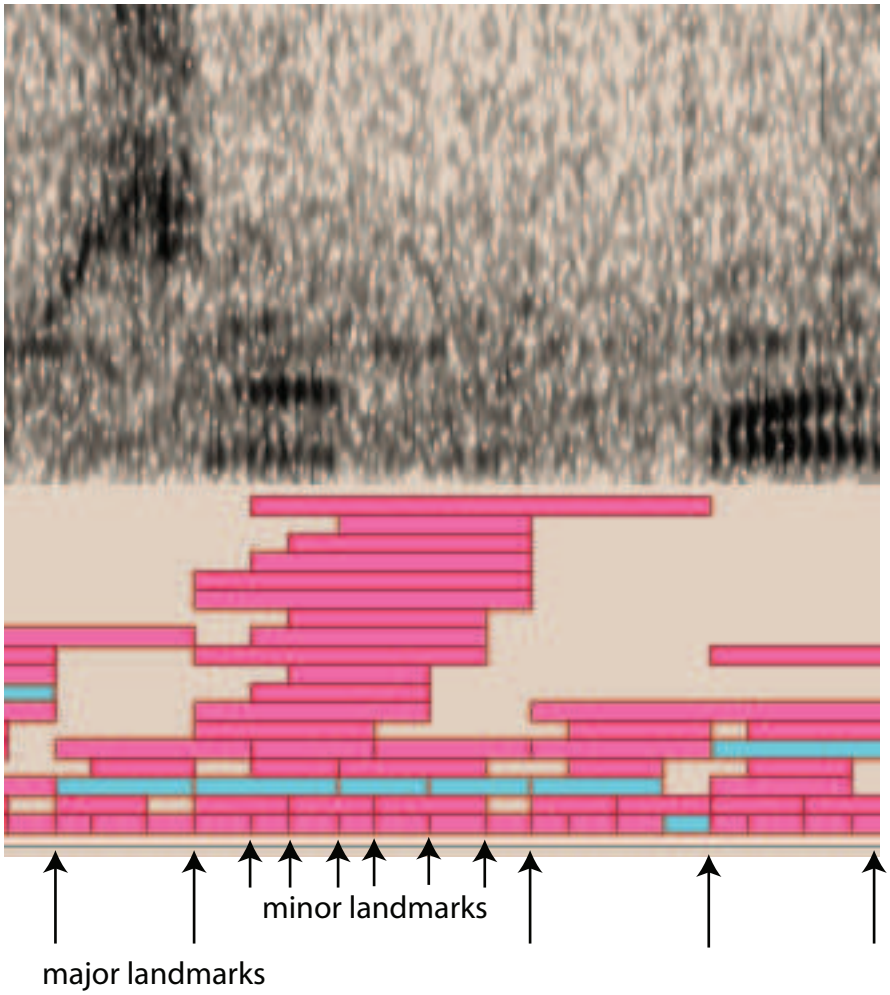

Figure 2-3: Graphical display from the SUMMIT recognizer. The top panel displays a spectrogram of the speech signal which has been contaminated by noise. The bottom panel shows the segmentation network for the spectral change method. The major landmarks are indicated by the long arrows while the corresponding set of minor landmarks are illustrated by shorter arrows. The darker colored segments illustrate the segmentation with the highest recognition score during search.

2.2.2 Observation Model

In frame-based modeling, acoustic features are computed and scored at a fixed frame-rate. In segment-based modeling, features are computed across segments. In SUMMIT, two types of features are computed for each hypothesized segment in the segmentation network, namely *segmental* features and *landmark* features. Segmental features are computed for each hypothesized segment by taking averages of the frame-based features across a particular segment. Landmark features are calculated from

features centered around landmarks. Figure 2-4 shows a diagram of frame-based features, and corresponding landmark and segment-based features.

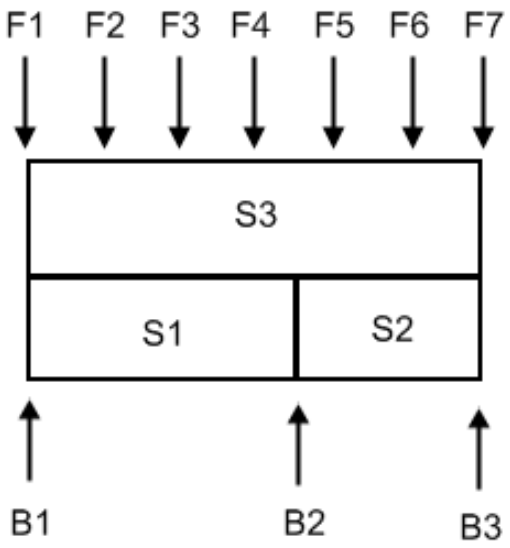

Figure 2-4: Diagram of frame-based, landmark and segmental features. The frame-based features, $F1, \ldots F7$, are computed at a fixed frame rate. The landmark features, denoted by $B1 \ldots B3$, are calculated at segmental boundaries. Finally, the segmental features, $S1 \ldots S3$, span across each segment.

A corresponding set of sub-word unit models U, known as segment and landmark models, are trained on the corresponding segment and landmark features respectively. Again, the landmark and segment models are both modeled as GMMs. The observation model then computes an acoustic score for a particular sub-word unit at each segment by summing the landmark and segment model scores for that segment.

2.2.3 Pronunciation and Language Models

The pronunciation and language models in SUMMIT are similar to those discussed in Sections 2.1.3 and 2.1.4 respectively.

2.2.4 Recognition Phase

Recognition in the SUMMIT system is implemented using a weighted finite-state transducer (FST) [33], which is represented as a cascade of smaller FSTs:

$$R = (S \circ O) \circ (C \circ P \circ L \circ G) \tag{2.8}$$

In Equation 2.8:

- S represents the acoustic segmentation described in Section 2.2.1
- O represents the acoustic observation space
- C relabels context-dependent acoustic model labels as context-independent phonetic labels
- P applies phonological rules mapping phonetic sequences to phoneme sequences
- L represents the lexicon which maps phoneme sequences to words
- G is the language model that assigns probabilities to word sequences

Intuitively, the composition of $(C \circ P \circ L \circ G)$ represents a pronunciation graph of all possible word sequences and their associated pronunciations. Similarly, the composition of $(S \circ O)$ is the acoustic segmentation graph representing all possible segmentations and acoustic model labelings of a speech signal. Finally, the composition of all terms in R represents an FST which takes acoustic feature vectors as input and assigns a probabilistic score to hypothetical word sequences. The single best sentence is found by a Viterbi search through R. If n-best sentence hypotheses are needed, an A^* search is then applied.

2.3 Broad Class Pre-processor in Attila and SUMMIT Frameworks

Both the Attila and SUMMIT recognizers are utilized for recognition experiments in this thesis. A block diagram of the proposed system is shown in Figure 2-5.

First, the Attila HMM system is used to recognize broad classes. In Chapter 3 we introduce an instantaneous adaptation technique within the HMM framework for recognizing broad classes. These broad classes are then used as a pre-processor within the SUMMIT framework in the landmark detection and search phases.

In Chapter 4 we discuss how to use broad classes, which are robustly identified in noise, as a pre-processor to aid in landmark detection. Once the set of acoustic landmarks are generated, the landmarks are connected together to form a set of possible segmentations of the utterance. The segment graph is then passed to the scoring and search phase to find the best set of hypothesized words.

Furthermore, in Chapter 5 we discuss using broad classes to identify reliable regions in the speech signal, and hence limit the number of paths searched and models scored during the scoring and search component of the recognition process. Before moving on to discuss these contributions in more detail, we first outline the main corpora used in this thesis.

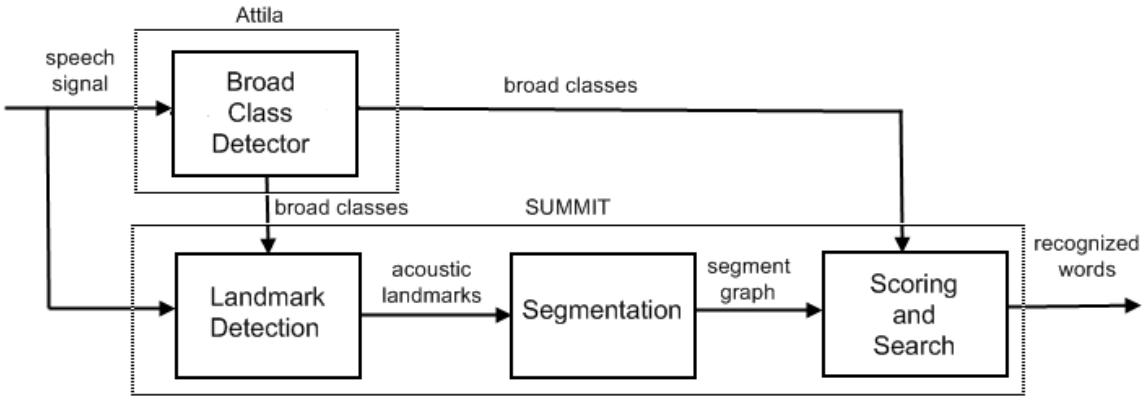

Figure 2-5: A block diagram of the speech recognition system utilized in thesis. The broad class recognizer in Attila is used as a pre-processor to aid in landmark detection and search within the SUMMIT framework.

2.4 Speech Recognition Corpora

Both phonetic recognition and word recognition tasks are explored in this thesis. The following sections describe, in more detail, the different corpora used.

2.4.1 TIMIT

TIMIT [57] is a continuous speech recognition corpus recorded and transcribed by Texas Instruments (TI) and the Massachusetts Institute of Technology (MIT), respectively. It contains over 6,300 utterances read by 630 speakers, including 438 males and 192 females, representing the 8 major dialects of English. Each speaker reads 10 sentences, including 2 *sa* sentences designed to represent dialectical differences, 5 *sx* sentences which cover all phoneme pairs and 3 phonetically diverse *si* utterances.

The sentences from the corpus are divided into three sets. The training set consists of 3,696 sentences from 462 speakers. This set is used to train various models used by the recognizer. The development set is compromised of 400 utterances from 50 speakers and is used to train various tuning parameters in the broad class algorithms. The full test set includes 944 utterances from 118 speakers, while the core test set is a subset of the full test set containing 192 utterances from 24 speakers. In this thesis, results are only reported on the full test set.

2.4.2 Noisex-92

To simulate noisy speech in TIMIT, we add various types of noise from the Noisex-92 speech-in-noise corpus [93], which was created by the Speech Research Unit at the Defense Research Agency to study the effect of additive noise on speech recognition systems. The corpus contains the following noises:

- White noise
- Pink noise
- High frequency radio channel noise

- Speech babble
- Factory noise
- Military Noises: fighter jets (Buccaneer, F16), engine room noise, factory operations room noise, tank noise (Leopard, M109), machine gun
- Volvo 340 car noise

In this thesis we look at three specific types of noise - pink, speech babble and factory noise. We specifically focus on these three noise types as they differ in their stationarity and harmonicity properties. The pink noise was acquired by sampling a high-quality analog noise generator. The speech babble was obtained by recording samples of 100 people speaking in a canteen. Finally, the factory noise was obtained by recording noise samples in a car production hall onto a digital audio tape. We simulate noisy speech by adding noise from the Noisex-92 set to clean TIMIT speech at signal-to-noise ratios (SNRs) in the range of -5dB to 30dB. ‘

2.4.3 Aurora

Experiments are also conducted using the Aurora-2 corpus [45], which consists of clean TI-digit utterances with artificially added noises. The TI-digits consist of male and female English speakers reading digit sequences up to 7 digits. To simulate noisy speech, a diverse set of noises are selected to represent various telecommunication areas. These noises include suburban train, crowd of people (babble), car, exhibition hall, restaurant, street, airport and train station noises. The noise signals are added to the TI-digits in SNRs ranging between 20dB and -5dB in 5dB increments.

The training set consists of 8,440 recordings of 55 male and 55 female adults. To simulate noisy speech, the recordings are equally split into 20 subsets, with each subset representing one of 4 different noise types and one of 5 different SNRs. The four noise types include suburban train, babble, car and exhibition noises while the SNRs range from 20dB to 5dB and clean conditions.

In the test set, 4,004 utterances from 52 male and 52 female speakers are split into 4 subsets of 1,001 utterances each. One of the noise conditions is added to each subset of 1,001 utterances in SNRs ranging between clean and -5dB. In the first test set, known as *Test Set A*, the 4 noises which match the training set, namely suburban train, babble, car and exhibition, are added to one of the subsets. In *Test Set B*, restaurant, street, airport and train station noises, different from the training set, are added to created a training-test mismatched scenario. Finally in *Test Set C* only 2 of the 4 subsets of 1,001 utterances are used, and suburban train and street are used as the noise signals. In this set, the speech and noise signals are first filtered with a filter that attenuates lower frequencies. In this thesis, experiments are only conducted using *Test Set A*.

2.4.4 CSAIL-info

CSAIL-info is a speech-enabled kiosk which provides information about people, rooms, and events in the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT. The system is linked to an updated online database of CSAIL personnel and seminars, and thus it is constantly updated to adapt to new user names and seminar announcements.

The CSAIL-info Kiosk is located in a high-traffic public area in the lobby of CSAIL, with a microphone mounted near the touch screen of the tablet PC. In addition to noise from the computer, recordings of user speech are also contaminated by various types of non-stationary noises, including background speech, elevator door opening/closing, and reverberation caused by hard surfaces of the large surrounding space.

The Spoken Language Systems Group at MIT has collected over 9,000 utterances from users interacting with the kiosk. Since the focus of our experiments is noise robustness, we have removed all sentences that contain out of vocabulary (OOV) words, allowing for a total vocabulary size of over 8,000 words. The data is divided into the following three sets:

- The *training* set consists of 6,140 sentences. This is used to train various models used by the recognizer.
- The *development* set contains 859 sentences used to design and develop the various broad class algorithms.
- The *test* set includes 876 sentences used to test our developed model.

To create unbiased experimental conditions, the sentences in the training, development and test sets do not overlap.

2.5 Chapter Summary

In this chapter, we presented a framework for the frame-based and segment-based systems utilized in this thesis. We first discussed the frame-based Attila HMM system, which will be used for broad class recognition. Next, we introduced the segment-based SUMMIT recognizer, which we will investigate for using broad class knowledge in the landmark detection and search stages. We also reviewed the main corpora used for phonetic and word recognition experiments in this thesis. In the next three chapters, we present various experiments on these corpora, using both the Attila and SUMMIT recognizers.

Chapter 3

Incremental Adaptation with Extended Baum-Welch Transformations

3.1 Introduction

In order to utilize broad classes for robust speech recognition, we first explore a method to robustly recognize broad classes in the presence of noise. In this chapter, we introduce a novel instantaneous adaptation technique using the Extended Baum-Welch (EBW) Transformations to robustly identify these broad classes. Unlike most adaptation methods, which are computationally expensive and require a lot of data for adaptation, the adaptation method presented is much less data intensive.

We then incorporate this adaptation technique into a frame-based Hidden Markov Model (HMM) framework for recognition of broad classes. We explore a frame-based HMM to recognize broad classes for three reasons. First, segment-based models are very sensitive to noise and have a difficult time in detecting variable frame boundaries (i.e., acoustic landmarks) [80]. However, since frame-based techniques compute observations at a fixed frame rate, they are less sensitive. Second, HMMs still continue to be the dominant acoustic modeling technique in speech recognition to date

[68]. Therefore, we hope that incorporating our instantaneous measure into an HMM framework will introduce a new decoding metric that can be explored for general speech recognition tasks. Third, because of the salience of broad classes, we are keen on exploring their benefits in noisy conditions. Specifically, we are interested in investigating broad classes as a pre-processor in determining more reliable boundaries for segment-based speech recognition systems and aiding in island-driven search. Both of these ideas will be discussed in Chapters 4 and 5 respectively.

3.1.1 Related Work

Noise adaptation is a popular technique for noise robust speech recognition. This method attempts to adapt initial recognition models to a noisy target environment in order to accommodate the noise and recognize noisy speech [34]. Current approaches for adaptation include both batch and instantaneous adaptation, depending on the amount of data used to adapt the initial models.

Batch adaptation generally requires a large amount of data from the target domain to adapt initial models. The most common technique is *Maximum a-Posteriori (MAP)* adaptation [28], which estimates an adapted model as a linear interpolation of the initial model and a model estimated from target data. *Maximum Likelihood Linear Regression (MLLR)* [62] is another popular method, generally requiring a few utterances from the target domain to estimate the updated model. In this technique, a set of linear transformation matrices is estimated to transform model parameters and maximize the likelihood on the test data. While both MLLR and MAP have shown success in a variety of tasks, these techniques perform poorly when limited adaptation data is available, as maximum likelihood estimates of the transformed model are poor.

Incremental adaptation techniques require a minimal amount of data, and attempt to improve recognition on the same data that is used during recognition. The most common technique is *Feature Space Maximum Likelihood Linear Transform* (fMLLR) [63], where the feature vectors themselves are transformed to a new space to maximize the likelihood of this transformed data given an initial set of models. While fMLLR

has shown success for incremental adaptation, it requires storing a large number of parameters and is computationally expensive to implement.

In this work, we explore using the EBW transformations to derive an incremental adaptation measure, which suffers from neither the computational complexity of other incremental adaptation techniques nor the data scarcity issues of batch adaptation.

3.1.2 Proposed Approach

Given some input data and a family of models, the goal of a typical pattern recognition task is to evaluate which model best explains the data. Typically, an objective function such as a likelihood probability, is computed to measure how well the model characterizes the data. Recently, a new approach for evaluating model fitness to data has been explored which is based on the principle of quantifying the effort required to change one model into another given some evaluation data. For example, the Earth Mover's Distance (EMD) [77] evaluates model fitness to data by calculating the minimal cost needed to transform one distribution into another. In addition, feature space Gaussianization [69] computes a distance between models in an original and transformed feature space.

In this chapter, we look to evaluate model fitness by using a gradient steepness measurement. Given a set of initial models, some data, and an objective function, we can re-estimate each of the models given the current data by finding the best step along the gradient of the objective function. During such an update, each of the models changes such that models that fit the data best change the least, and correspondingly have flatter gradient slopes.

One of the popular training methods used to estimate updated models, which we explore in this work, is the Extended Baum-Welch (EBW) transformations [35]. The EBW transformations have been used extensively in the speech recognition community as a discriminative training technique to estimate model parameters of Gaussian mixtures. For example, in [91], the EBW transformations were used for Maximum Mutual Information (MMI) training of large vocabulary speech recognition systems. In addition, [75] explores the EBW update equations under a variety of objective

functions for discriminative training. The EBW transformations have also been used to derive an explicit formula to measure the gradient steepness required to estimate a new model given an initial model and input data [51], [52]. This gradient steepness measurement is an alternative to likelihood to describe how well the initial model explains the data.

The advantages of this gradient steepness measurement have been observed in a variety of tasks. In [81], we redefined the likelihood ratio test, typically used for unsupervised audio segmentation, with this measure of gradient steepness. We showed that our EBW unsupervised audio segmentation method offered improvements over the Bayesian Information Criterion (BIC) and Cumulative Sum (CUSUM) methods. In [84], we used this gradient metric to develop an audio classification method which was able to outperform both the likelihood and Support Vector Machine (SVM) techniques. Finally, in [83] we observed the benefits of the gradient metric on a speech/non-speech segmentation task, which also outperformed the likelihood method, specifically when the initial models were poorly trained.

In this work, we are interested in exploring this gradient metric for broad class recognition via Hidden Markov Models (HMMs) [76]. When HMMs are used for acoustic modeling, the Viterbi algorithm [94] is generally used during decoding to find the most likely sequence of HMM states and corresponding words. This decoding is accomplished by first computing likelihood scores for each frame given all HMM states, and then performing a dynamic programming Viterbi search to find the most likely sequence of states. In this work, we look at replacing the likelihood scores computed at each frame with the EBW gradient steepness measurement. We explore looking at both the absolute change in gradient steepness - i.e., how much the initial model must move to the updated model to explain the current frame of data, as well as the relative change - i.e., how the initial model changes to the updated model relative to the initial model. We show that these EBW metrics, which are computed on a per-frame basis and thus require only a small change to the HMM formulation, are able to provide a simple and effective instantaneous model adaptation technique.

3.1.3 Goals

In this chapter, we demonstrate that the EBW gradient steepness measure is a general technique to explain the quality of a model used to represent the data. In addition, it provides a simple and effective noise adaptation technique, which does not suffer from the data and computational complexities of other adaptation techniques. First, we examine both the absolute and relative change in EBW gradient to explain model fit to the data. We find that the relative EBW metric outperforms the standard likelihood method, both when initial models are adapted via MLLR and without adaptation, for broad phonetic class (BPC) recognition on the TIMIT corpus [57]. In addition, we explore the advantages of EBW model re-estimation in noisy environments, demonstrating the improved performance of our gradient steepness metric over likelihood across a variety of signal-to-noise ratios (SNRs).

3.1.4 Overview

In the following section, we describe the EBW transformations. The implementation of the EBW gradient metric in an HMM framework is described in Section 3.3. Section 3.4 presents the experiments performed, followed by a discussion of the results in Section 3.5. Finally, Section 3.6 summarizes the chapter.

3.2 Extended Baum-Welch Transformations

3.2.1 EBW Transformations Formulation

Assume that observations $O = (o_1, ..., o_M)$, from frames 1 to M, is drawn from a Gaussian λ_j parameterized by the following mean and variance parameters $\lambda_j = \{\mu_j, \sigma_j^2\}$. Let us define the probability of frame $o_i \in O$ given model λ_j as $p(o_i|\lambda_j) = z_{ij} = \mathcal{N}(\mu_j, \sigma_j^2)$. Let $F(z_{ij})$ be some objective function over z_{ij} and $c_{ij} = z_{ij}\frac{\delta}{\delta z_{ij}}F(z_{ij})$. Intuitively, c_{ij} measures the gradient steepness of the objective function, as captured by the objective function derivative term. The steeper the gradient slope, the larger c_{ij}.

Given an objective function, initial model λ_j and observation data, there are many statistical optimization techniques to estimate a new model for the data. In the simplest case, maximizing the objective function directly will lead to a new model estimate. However, in situations where the objective function cannot be maximized directly, an auxiliary function is defined, where maximizing the auxiliary function leads to an increase in the objective function. Standard techniques to re-estimate model parameters by maximizing the auxiliary function include both the Baum-Welch (BW) [6] and Expectation Maximization (EM) [7] algorithms. The disadvantage of these methods is that the auxiliary function is only defined if the objective function is a likelihood function. To address this issue, another optimization technique involves finding the extremum (that is minimum or maximum) of an associated function, Q, given by Equation 3.1. The benefit of the associated function is that it is defined for any rational objective function.

$$Q = \sum_i z_{ij} \frac{\delta F(\{z_{ij}\})}{\delta z_{ij}} \log \hat{z}_{ij} \tag{3.1}$$

Optimizing Equation 3.1 will lead to closed-form solutions to re-estimate model parameters $\hat{\lambda}_j$, known as the EBW transformations [35], such that the re-estimated model parameters increase (or decrease) the associated and corresponding objective functions. The EBW solutions to re-estimate model parameters $\hat{\lambda}_j = \lambda_j(D) = \{\mu_j(D), \sigma_j^2(D)\}$ are given as follows:

$$\hat{\mu}_j = \hat{\mu}_j(D) = \frac{\sum_{i=1}^{M} c_{ij} o_i + D\mu_j}{\sum_{i=1}^{M} c_{ij} + D} \tag{3.2}$$

$$\hat{\sigma}_j^2 = \hat{\sigma}_j^2(D) = \frac{\sum_{i=1}^{M} c_{ij} o_i^2 + D\left(\mu_j^2 + {\sigma_j}^2\right)}{\sum_{i=1}^{M} c_{ij} + D} - \hat{\mu}_j^2 \tag{3.3}$$

Here D is a constant chosen in the EBW model re-estimation formulas, given by Equations 3.2 and 3.3. If D is very large then model re-estimation is slow but the associated function, and corresponding objective function, increase with each

iteration, that is, $F(\hat{z}_{ij}) \geq F(z_{ij})$. However if D is too small, model re-estimation may not increase the objective function on each iteration. For a deeper mathematical understanding of these EBW update equations, we refer the reader to Appendix A.1.

3.2.2 EBW Gradient Steepness

Given the EBW formulas, we now discuss the derivation of the EBW gradient steepness measurement, as defined in [51]. Figure 3-1 gives a graphical illustration of the EBW model updates. The graph shows different values of the objective function F as we change the model parameter $\lambda(\epsilon)$. $\lambda(\epsilon)$ are transformations of the mean and variance as defined in (3.2) and (3.3) respectively. The parameter ϵ controls the rate at which we estimate our updated model. A larger value of ϵ favors the updated model more while a smaller value of ϵ gives more weight to the initial model. A more detailed investigation on the tuning of ϵ is presented in Appendix A.2.

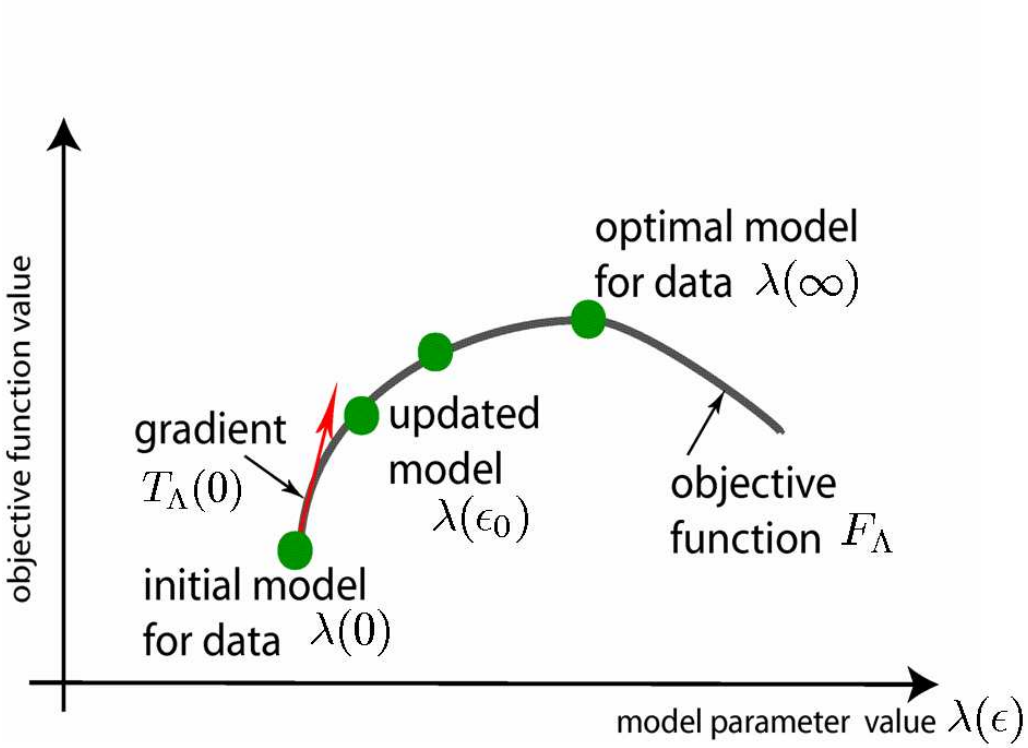

Figure 3-1: Illustration of model re-estimation via the Extended Baum-Welch (EBW) Transformations

Let us denote a tangent to the curve F at point $\{0, F(\lambda(0))\}$ as follows:

$$T_\Lambda(0) = lim_{\epsilon_0 \to 0} \frac{F_\Lambda(\hat{\lambda}(\epsilon_0)) - F_\Lambda(\lambda(0))}{\epsilon_0} \tag{3.4}$$

Intuitively, the flatter the tangent to the curve at point $\lambda(0)$, the better the initial model $\lambda(0)$ fits the data. In [51], it was shown that T could be represented as a sum of squared terms and therefore is always non-negative. This guarantees that F increases per iteration and provides some theoretical justification for using the gradient metric T as a measure of quality of model fit to data.

With the graphical illustration of the EBW gradient steepness measurement given in Figure 3-1, we can now derive our gradient measurement more formally. Note that we will now use $D = \frac{1}{\epsilon}$. Using EBW transformations (3.2) and (3.3) such that $\lambda_j \to \hat{\lambda}_j(D)$ and $z_{ij} \to \hat{z}_{ij}$, [51], [52] derives a linearization formula between $F(\hat{z}_{ij})$ and $F(z_{ij})$ for large D as:

$$F(\hat{z}_{ij}) - F(z_{ij}) = T_{ij}/D + o(1/D) \tag{3.5}$$

A large value in T means the gradient to adapt the initial model λ_j to the data x_i is steep and $F(\hat{z}_{ij})$ is much larger than $F(z_{ij})$. Thus the data is much better explained by the updated model $\hat{\lambda}_j(D)$ compared to the initial model λ_j. However a small value in T indicates that the gradient is relatively flat and $F(\hat{z}_{ij})$ is close to $F(z_{ij})$. Therefore, the initial model λ_j is a good fit for the data.

In [51], Kanevsky also derives a closed form solution for T for large D. For example, using Equation 3.5, defining the objective function F to be the log-likelihood function, i.e., $F(z_{ij}) = \log p(o_i|\lambda_j) = \log(z_{ij})$, and c_{ij} by Equation 3.6,

$$c_{ij} = z_{ij} \frac{\delta}{\delta z_{ij}} F(z_{ij}) = \frac{z_{ij}}{z_{ij}} = 1, \tag{3.6}$$

the following formula gives a closed form solution for the gradient steepness T. Here r indexes a dimension of the feature vector o_i, where o_i has dimension 1 to d.

$$T_{ij} = \left\{ \sum_{r=1}^{d} \frac{\{c_{ij}[(o_{ir} - \mu_{rj})^2 - (\sigma_{rj})^2]\}^2}{2(\sigma_{rj})^4} + \sum_{r=1}^{d} [\frac{c_{ij}(o_{ir} - \mu_{rj})}{\sigma_{rj}}]^2 \right\} \tag{3.7}$$

If we quantify gradient steepness by taking the difference in objective functions at two model parameter values (i.e., the left side of Equation 3.5), the actual model re-estimation using the EBW Transformations must be performed. The benefit of Equation 3.7 is that it gives a closed form solution for gradient steepness without having to explicitly re-estimate models. We have observed the computational benefits of using Equation 3.7 as a measure of gradient steepness in developing an unsupervised audio segmentation algorithm [81]. We will refer to the EBW metric in Equation 3.7 as *EBW-T*.

The disadvantage of using *EBW-T* is that it only holds for large D, meaning the rate of adapting to the updated model, $\hat{\lambda}_j(D)$, cannot be adjusted. Typically, we have found better performance gains by taking the difference in objective function values, as illustrated by the left side of Equation 3.5, where the rate of adaptation can be controlled. This is discussed in more detail in [84], which explores the use of the gradient steepness metric for audio classification.

Therefore, in this thesis we focus our attention on gradient steepness using the difference in objective function values given in Equation 3.5, which we refer to as *EBW-F*. We also introduce a normalized version the left side of Equation 3.5 which we will call *EBW-F Norm*. Recall that Equation 3.5 measures the gradient steepness required to adapt an initial model λ_j to the target data o_i. While this metric can be applied for any general pattern recognition task (i.e., [81], [84], [83]), in this thesis we will concentrate on the use of this gradient metric for broad class recognition via HMMs.

3.3 EBW Gradient Metric for HMMs

Given a set of acoustic observations $O = \{o_1, o_2 \ldots o_T\}$ associated with a speech waveform, the goal of a speech recognition system is to find the sequence of sub-word

units $\hat{W} = \{w_1, \ldots w_k\}$ that most likely produced the given observation sequence. In other words, we want to maximize the following expression:

$$\hat{W} = \arg\max_{W} P(W|O) = \frac{P(O|W)P(W)}{P(O)} \tag{3.8}$$

As discussed in Section 2.1, $P(O|W)$ is referred to as the acoustic model while $P(W)$ is the language model. In this section, we look at representing the acoustic model via an HMM, and will subsequently extend our EBW gradient metric in this context.

3.3.1 HMM Scoring with Likelihood

Given observation sequence O, HMMs can be used to find the optimal state sequence through time $Q = \{q_1, q_2 \ldots q_T\}$ that produced the given T observations. An HMM is defined over a set of N states $S = \{s_1, s_2 \ldots s_N\}$ and observations O, and is represented by the following three parameters [76]:

- State Transition Probability Distribution:
  $a_{ij} = P(q_t = s_j | q_{t-1} = s_i)$
- Observation Symbol Probability Distribution:
  $b_i(o_t) = P(o_t | q_t = s_i)$
- Initial State Distribution:
  $\pi_i = P(q_1 = s_i)$

Typically, the output distribution for each state s_k is drawn from a mixture of L gaussians. Let z_{tj}^k be the likelihood of observation o_t given component j from Gaussian mixture model (GMM) k and w_j^k the *a priori* weight of component j in GMM k. Then the log-likelihood of o_t from model λ^k can be defined as follows:

$$b_k(o_t) = \log P(o_t | q_t = s_k) = \log \sum_{j=1}^{L} w_j^k z_{tj}^k \tag{3.9}$$

Given the set of states S and corresponding models $\Lambda = \{\lambda^1, \lambda^2 \dots \lambda^N\}$, the Viterbi algorithm is generally used to find the optimal state sequence. To find this sequence, first define $\delta_t(i)$ as the best score along a single path up to time t which ends in state s_i at time t as:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1 q_2 \dots q_t = S_i, O_1 O_2 \dots O_t | \Lambda) \tag{3.10}$$

By induction, the probability of the best path up to time t which ends in state s_j at time $t+1$ is defined as:

$$\delta_{t+1}(j) = \max_i [\delta_t(i) + log(a_{ij})] + log(b_j(o_{t+1})) \tag{3.11}$$

Equation 3.11 illustrates that the best state at each time depends on the scores assigned to previous states as well as transition probabilities a_{ij} between states, capturing the inherent HMM structure. In the next section, we discuss how to find the best state sequence using the EBW gradient metric.

3.3.2 HMM Scoring with EBW-F Metric

Instead of scoring each observation frame using standard likelihood, we can score it using the EBW gradient steepness measurement given by the left side of Equation 3.5. Let us define objective function $F(z_t^k)$ to be the log-likelihood of observation o_t given state model λ^k as:

$$F(z_t^k) = \log \sum_{j=1}^{L} w_j^k z_{tj}^k \tag{3.12}$$

and similarly c_{tj}^k as:

$$c_{tj}^k = z_{tj}^k \frac{\delta}{\delta z_{tj}^k} F(z_{tj}^k) = \frac{z_{tj}^k w_j^k}{\sum_{l=1}^{L} w_l^k z_{tl}^k}. \tag{3.13}$$

In addition, given initial state model $\lambda^k = \{\mu^k, \sigma^k\}$ and observation o_t, a new model $\hat{\lambda}^k(D)$ can be re-estimated at each frame t using Equations 3.2 and 3.3. This process is illustrated in Figure 3-2.

Using Equation 3.5, the objective function for $F(z_t^k)$ given by Equation 3.12 and

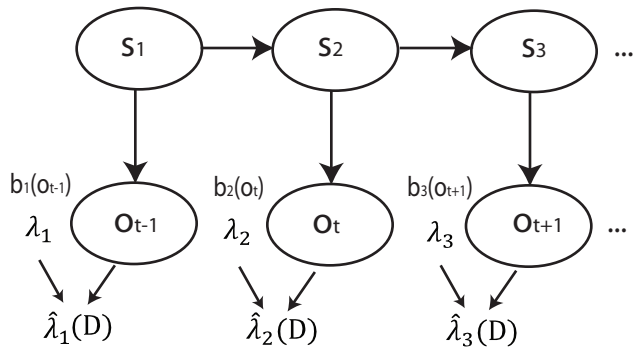

Figure 3-2: HMM State Model Re-estimation using the Extended Baum-Welch (EBW) Transformations

the adapted model $\hat{\lambda}^k(D)$, the state output score at frame o_t can be calculated using Equation 3.14. We will refer to the state output score computed in this manner as *EBW-F*.

$$b_k(o_t) = \left(F(\hat{z}_t^k) - F(z_t^k)\right) \times D \tag{3.14}$$

Note that this gradient steepness metric in Equation 3.14 requires just a simple change to the HMM formulation. As shown in Figure 3-2, models are re-estimated and adaptation occurs on a per-frame basis, allowing for advantages over batch adaptation methods. Furthermore, only the sufficient statistics for the current model being re-estimated are required to be stored in memory, making this method computationally efficient compared to fMLLR, for example.

Using the EBW score assigned to each state from Equation 3.14, the best path is again found through the Viterbi algorithm given in Equation 3.11. However, the better a model fits the data, the smaller the EBW score, so $\delta_t(i)$ is now defined as the set of best (smallest) EBW scores along a single path up to time t which ends in state s_i.

$$\delta_t(i) = \min_{q_1, q_2, \ldots, q_{t-1}} EBW(q_1 q_2 \ldots q_t = S_i, O_1 O_2 \ldots O_t | \Lambda) \tag{3.15}$$

Therefore, by induction, $\delta_{t+1}(j)$ is defined as:

$$\delta_{t+1}(j) = \min_i[\delta_t(i) - log(a_{ij})] + b_j(o_{t+1}) \tag{3.16}$$

Note that, to reflect this minimum change, the negative log-likelihood of a_{ij} is also calculated. The objective function in Equation 3.14 is the same as that used in [84], though now applied to HMMs. In the next section, we discuss a novel change to this objective function which is more appropriate for an HMM framework.

3.3.3 HMM Scoring with EBW-F Normalization Metric

As shown in Equation 3.14, we score how well model λ^k fits o_t by looking at the difference in likelihood given the updated model $F(\hat{z}_t^k)$ compared to the likelihood given the initial model $F(z_t^k)$. Using this absolute measure allows us to compare model scores for a given input frame, as was done in [84]. However, we have observed that the magnitude of these scores loses meaning if we compare them across different frames. In other words, a lower absolute EBW score for one frame and one model does not necessarily imply a better model than a higher EBW score for another frame and another model. Having an EBW measure that can be compared across frames is particularly important in HMMs, as scores for a state sequence are computed by summing up scores assigned to individual frames.

Therefore, we compute the EBW score as the *relative* difference in likelihood given the updated model $F(\hat{z}_t^k)$ compared to the initial model likelihood $F(z_t^k)$. To compute this relative EBW score, we normalize Equation 3.14 by the original likelihood $F(z_t^k)$ as shown in Equation 3.17. We will refer to the state output score computed in this manner as the *EBW-F Norm* metric.

$$b_k(o_t) = \frac{\big(F(\hat{z}_t^k) - F(z_t^k)\big) \times D}{F(z_t^k)} \tag{3.17}$$

Using this relative EBW score provides a measure which can be compared across frames, which is important in the context of HMMs.

3.4 Experiments

Broad Phonetic Class (BPC) recognition is performed on the TIMIT corpus [57]. The 61 TIMIT labels are first mapped into 7 BPCs, ignoring the glottal stop 'q', as shown in Table 3.1. The labeling in Table 3.1 was determined based on a phonetic to BPC mapping defined in [36].

Broad Phonetic Class	TIMIT Labels
Vowels/Semivowels	aa ae ah ao aw ax axh axr ay eh er ey ih ix iy ow oy uh uw el l r w y
Nasals/Flaps	em en eng m n ng nx dx
Strong Fricatives	s z sh zh ch jh
Weak Fricatives	v f dh th hh hv
Stops	b d g p t k
Closures	bcl pcl dcl tcl gcl kcl epi pau
Silence	h#

Table 3.1: Broad Phonetic Classes and corresponding TIMIT Labels

Our experiments are conducted using the IBM Attila recognizer discussed in Section 2.1. We use 13 dimensional, perceptual linear prediction (PLP) features [42] obtained from a Linear Discriminant Analysis (LDA) projection [22] that are mean and variance normalized on a per utterance basis. In addition, each BPC is modeled as a three-state, left-to-right context-independent HMM with no skip states. The output distribution in each state is modeled by a mixture of 32 component diagonal covariance Gaussians. The language model is structured as a trigram. All models are trained on the standard NIST training set (3,969 utterances) in clean speech conditions. To analyze phonetic recognition performance in noise, we simulate noisy speech by adding pink noise from the Noisex-92 database [93] at signal-to-noise ratios (SNRs) in the range of 0dB to 30dB in 5dB increments. We train the EBW-F methods to find the optimal D range, described in more detail in Appendix A.2.2, using the development set (400 utterances).

We report phonetic recognition error rate results on both the development set and the full test set (944 utterances). The phonetic error rate (PER) is calculated by summing the number of hypothesized phonemes inserted (I), reference phonemes

deleted (D) and reference phonemes substituted (S), divided by the true number of reference phonemes N. The equation for the PER is given more explicitly by Equation 3.18. The insertion, deletion and substitution errors are determined using the NIST sclite scoring script [70], which aligns the hypothesized output to a reference text to calculate the three errors.

$$PER = \frac{I + D + S}{N} \tag{3.18}$$

3.5 Results

In this section, we discuss two experiments performed on the TIMIT corpus. First, we analyze the BPC recognition performance of the EBW-F, EBW-F Norm and likelihood methods, with and without MLLR adaptation, in a clean speech environment. Second, we explore the behavior of EBW model re-estimation in noisy environments. Note that all EBW techniques presented in this section use the Adaptive-D method, discussed in Appendix A.2.2, when setting the learning parameter D.

3.5.1 Clean Speech Recognition Performance

Table 3.2 shows the phonetic recognition error rates for the likelihood, EBW-F and EBW-F Norm metrics on the development and test sets, with the best performing method highlighted in bold. In this experiment, models were trained in clean speech conditions, and the test data was also drawn from clean speech. We investigate likelihood decoding using both initial baseline models and MLLR models adapted per utterance, with the number of regression classes optimized on the development set. We only explore adapting MLLR models per utterance since this is the smallest delineation we can use for adaptation and still maintain a fair comparison to the EBW metrics, where adaptation is performed per frame.

Table 3.2 indicates that the EBW-F Norm method outperforms the likelihood metric, with and without MLLR adaptation, on both the development and test sets, but the EBW-F method performs worse than both likelihood metrics. A Matched Pairs

Method	Development	Test
Likelihood - No MLLR	18.4	19.5
Likelihood - MLLR	18.6	19.8
EBW-F	18.7	19.9
EBW-F Norm	**17.7**	**18.9**

Table 3.2: BPC Error Rates on TIMIT development and test sets for clean speech conditions. The best performing technique is indicated in bold.

Sentence Segment Word Error (MPSSWE) significance test [29] indicates that the EBW-F Norm results are statistically significant from the other three metrics. Notice also that adapting models with MLLR using just one utterance actually leads to a higher error rate than using the likelihood metric without MLLR adaptation, showing the inefficiency of batch adaptation with little data. To explain the performance of the EBW metrics compared to the baseline likelihood, we analyze the relationship between EBW-F and likelihood scores (evaluated on a per-frame basis) vs. the EBW-F Norm and likelihood scores, illustrated in Figures 3-3 and 3-4 respectively.[1]

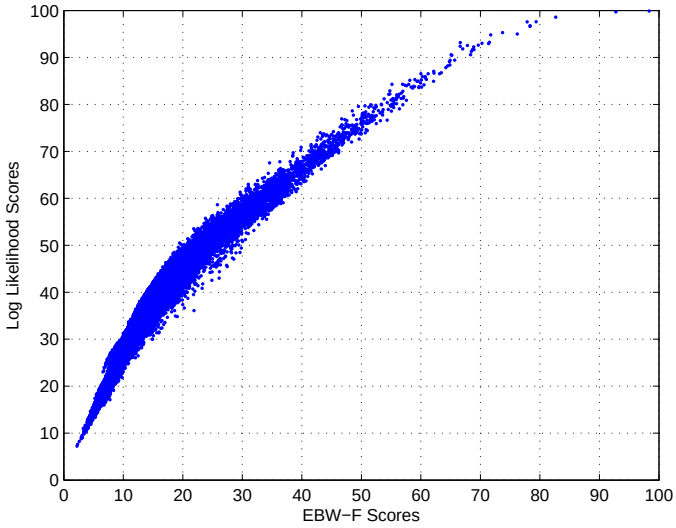

Figure 3-3: Regression of EBW-F scores against log-likelihood scores

First, observe that there is a strong positive correlation between the EBW-F and likelihood scores, as well the EBW-F Norm scores. It appears that the variance of

[1]Note that the likelihood score shown is actually the negative log-likelihood, so the better a model explains an observation, the smaller the negative log-likelihood and EBW scores (i.e., closer to origin).

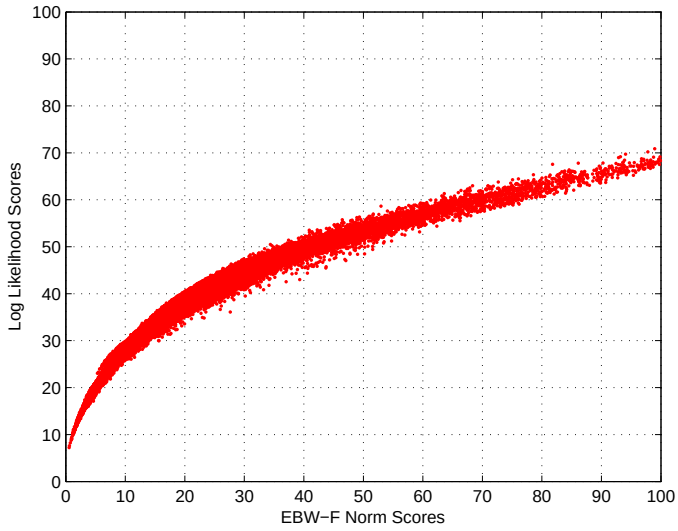

Figure 3-4: Regression of EBW-F Norm scores against log-likelihood scores

likelihood scores for a given EBW-F score is larger than the variance of likelihood scores for a given EBW-F Norm score. To quantify this variance more explicitly, we divide the likelihood scores at increments of 10, and take a weighted average of the variance of likelihood scores that fall within each bin. More specifically, first we define w_i to be the percentage of EBW scores that fall between increments $10 \times i$ and $10 \times (i+1)$. This is given more explicitly by Equation 3.19.

$$w_i = \frac{\text{Number of EBW Points Between } (10 \times i) \text{ and } (10 \times (i+1))}{\text{Total Number of EBW Points}} \tag{3.19}$$

Then, we quantify the total variance of likelihood scores as a weighted average of conditional variance of likelihood scores in each of these bins. This is given by Equation 3.20, where N is the total number of bins.

$$Var_{lik} = \sum_{i=0}^{N} w_i * Var(\text{Log-Likelihood Scores} | (10 \times i) \leq \text{EBW Scores} < (10 \times (i+1)) \tag{3.20}$$

Using this measure, we find the variance of the likelihood scores, when regressed

against the EBW-F scores, to be roughly 20.8. However, the variance of the likelihood scores when regressed against the EBW-F Norm scores is about 8.4, roughly 2.5 times less. This large variance for the EBW-F metric is due to the fact that the EBW-F score is an absolute measure and cannot really be compared across frames. Because Viterbi decoding determines the best path based on the scores of all individual frames in that path, if the EBW score for one frame is large it dominates and can throw off the entire score for the path. This is one reason why the EBW-F metric performs worse than likelihood when used in an HMM context. This motivated us to examine the EBW-F score in terms of relative change, thus introducing the EBW-F Norm metric.

The smaller variance of EBW-F Norm scores for a given likelihood score indicates that using the relative measure allows for a more direct comparison across frames. Also, notice that as models become worse, the EBW scores move even faster and there is a slight curve to the graph. As shown by Equation 3.17, EBW-F Norm captures the relative difference between the likelihood of a data given the initial model and the likelihood given a model estimated from the current observation being scored, while the likelihood just calculates the former. Thus, when the initial model is not a good fit for the data, we see that we must move this model quite a bit to explain the current input, and therefore the EBW score is quite large compared to likelihood.

To better understand the curve between the EBW-F Norm and Likelihood scores depicted in Figure 3-4, we looked at transforming the EBW-F Norm scores to produce a more linear relationship with the likelihood scores. The Box-Cox transformations [10] are a common method used to make the relationship between two variables more linear. The transformations are defined as follows:

$$\tau(EBW;\lambda) = \begin{cases} \frac{(EBW^{\lambda}-1)}{\lambda} \text{ if } \lambda \neq 0 \\ ln(EBW) \text{ if } \lambda = 0 \end{cases} \tag{3.21}$$

Here λ is the transformation parameter which controls the degree to which we transform the EBW scores. Figure 3-5 shows the correlation between likelihood and Box-Cox transformed EBW scores for different values of λ. In addition, Table 3.3

shows the PER for the EBW Box-Cox transformed scores for different λ values.

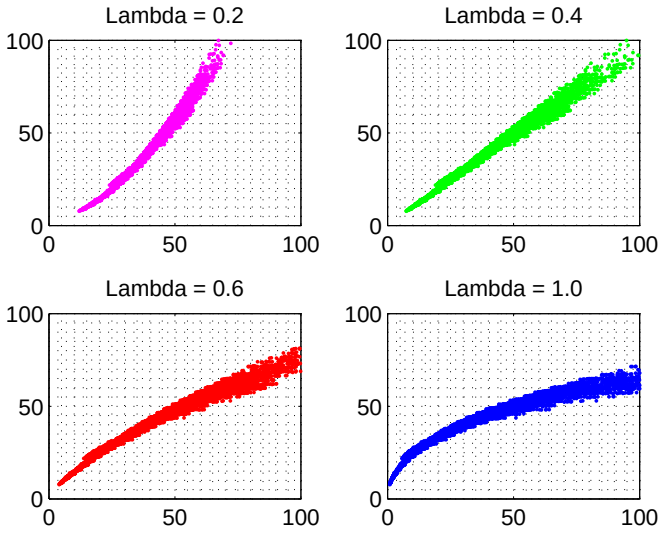

Figure 3-5: EBW Box-Cox Transformed (x-axis) vs. Likelihood (y-axis) scores for different λ values

Notice that as λ decreases, the correlation of EBW and likelihood scores moves from concave to convex and the PER increases. In addition, notice that $\lambda = 0.4$ produces the most linear relationship between the EBW-F Norm and likelihood scores, and the EBW Box-Cox transformed PER (i.e., 19.6) is very close to the likelihood PER given in Table 3.2 (i.e., 19.5).

λ	PER
0.2	20.5
0.4	19.6
0.6	19.1
1.0	**18.9**
1.2	20.0

Table 3.3: BPC Error Rates on the TIMIT Test Set using EBW Box-Cox Transformed scores for variable λ. The best performing metric is indicated in bold.

As we decrease λ and make the relationship of EBW and likelihood more convex, the PER increases. This shows that the true benefit of EBW over the likelihood occurs when models are poor, and the EBW scores are much higher relative to likelihood, producing the curve in Figure 3-4. Because scores from local frames are summed up

to determine the best path, the large EBW scores for models which do not fit the data well allow us to disregard these paths more confidently. However, if λ is increased past 1.0 and the relationship between the EBW and likelihood scores becomes increasingly concave, the PER again increases, indicating that there is a limit to how large EBW scores can be for poor models.

3.5.2 Noisy Speech Recognition Performance

In Section 3.5.1, we showed that the EBW-F Norm metric outperformed the likelihood method due to the model re-estimation inherent in EBW. In this section, given models trained in clean conditions, we analyze the benefit of the EBW-F Norm method when the target data is corrupted by noise. Recall that D controls the rate at which models are re-estimated. We would expect that, as models become a worse fit for the data, we must make D smaller and re-estimate models faster. Figure 3-6 shows the BPC Error Rates for the EBW-F Norm metric on the development set for different SNRs as D is varied. Again note that we are using the Adaptive-D metric discussed in Appendix A.2.2, and here D indicates the average range over which we adapt D.

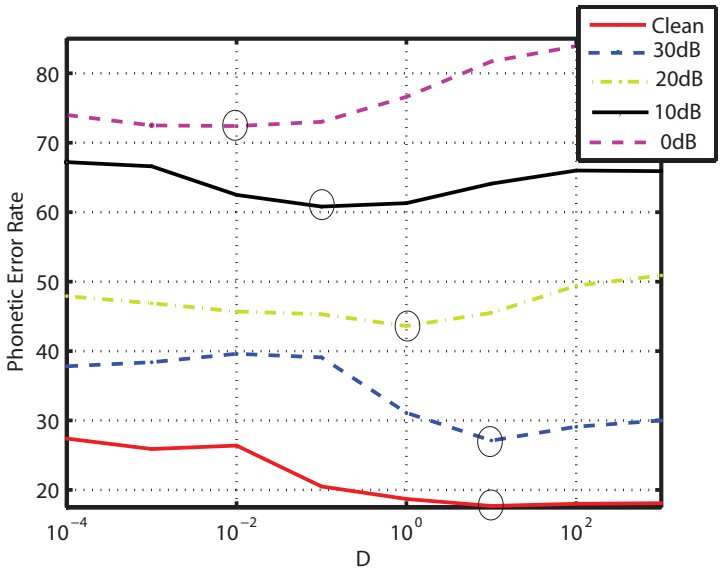

Figure 3-6: BPC Error Rates using EBW-F Norm Metric vs. D for different SNRs. Circles indicate the D at each SNR which gives lowest PER.

As the SNR decreases and the clean speech models become poorer estimates of the noisy data, we must decrease D and train models more quickly for better performance, as indicated by the circles in Figure 3-6. This shows the importance of the rate of model re-estimation, particularly when models are not a good fit for the data.

Table 3.4 shows the PER rate on the development and test sets for the EBW and likelihood methods across a variety of SNRs when models are trained in clean conditions and re-estimated in noisy speech using the optimal D values indicated in Figure 3-6. Notice that as the SNR is increased, the model re-estimation inherent in EBW allows for significant improvement over the likelihood metric. Thus, we see that the EBW-F Norm metric also provides a simple yet effective noise robust technique when compared to the likelihood measure.

Set	Method	clean	30dB	20dB	10dB	0dB
	Likelihood - No MLLR	18.4	28.2	45.0	65.2	75.6
development	EBW-F Norm	**17.7**	**27.1**	**43.6**	**60.8**	**72.4**
	% Err. Red.	3.8	3.9	3.1	7.7	4.2
	Likelihood - No MLLR	19.5	29.7	46.7	66.2	75.9
test	EBW-F Norm	**18.9**	**28.6**	**45.0**	**61.5**	**71.7**
	% Err. Red.	3.1	3.7	3.6	7.1	5.5

Table 3.4: BPC Error Rates on the TIMIT development and test sets for Likelihood and EBW-F Norm Metrics. Note that results are reported across different SNRs of pink noise when models are trained on clean speech. The best performing metric is indicated in bold.

Analyzing the error rates given in Table 3.4 further, Figure 3-7 shows the errors on the development set within each the 7 BPCs as a function of SNR. First, notice that for non-harmonic classes such as nasals, stops, closures and strong fricatives, the error rate increases significantly as the noise level increases. Second, the error rates in the vowel/semi-vowel class do not degrade as quickly, indicating that harmonic classes such as vowels and semi-vowels are much better preserved in noise. Third, notice that the error rate within weak fricatives does not degrade like other non-harmonic classes. A closer analysis reveals that most of the confusions with the other four non-harmonic classes occur with the weak fricative class, indicating that weak fricatives are over hypothesized. Fourth, observe that the silence class also has a relatively lower error

rate. One explanation is because each utterance in the TIMIT data set always begins with the silence class, and thus we have forced the recognizer to hypothesize this class first, resulting in a lower error rate.

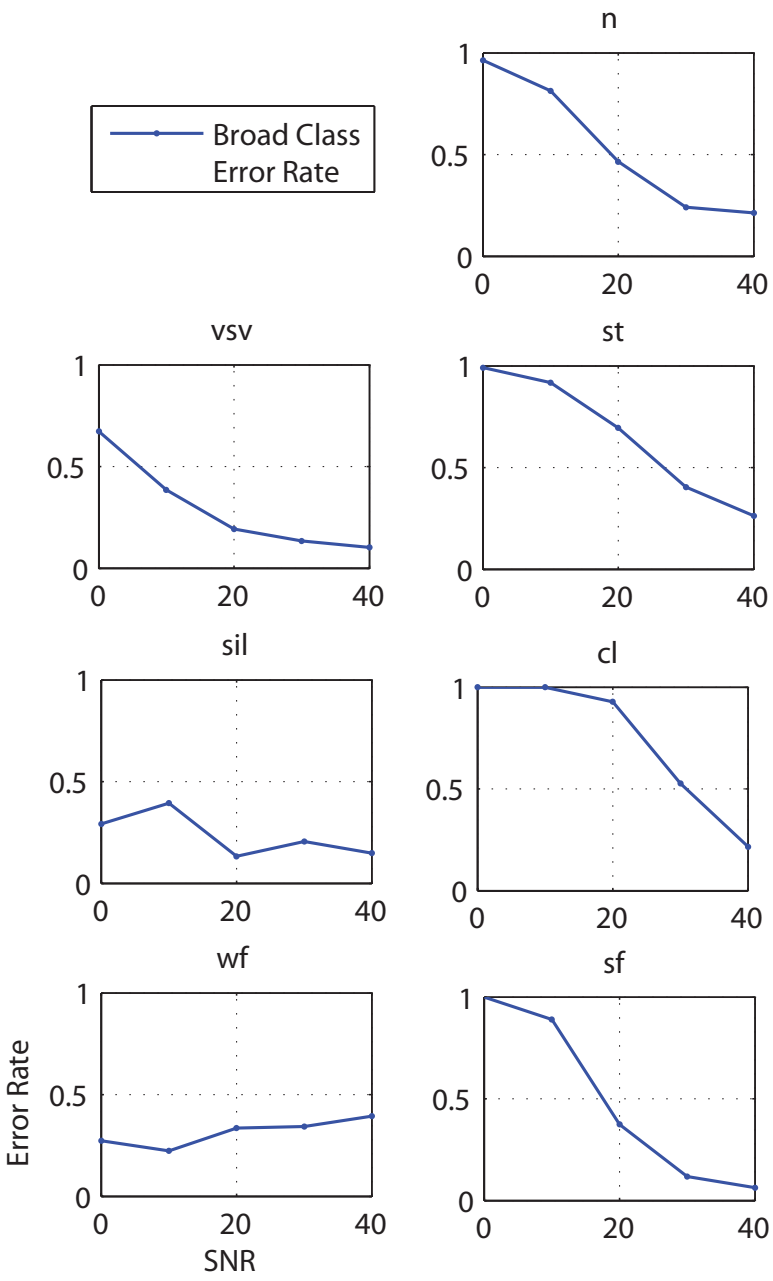

Figure 3-7: Error rate within individual BPCs as a function of SNR

3.6 Chapter Summary

In this chapter, we introduced a novel instantaneous adaptation technique using a gradient steepness measurement derived from the EBW transformations. This gradient steepness metric provided a simple yet effective adaptation technique which did not suffer from the data and computational intensities of other adaptation methods such as MAP, MLLR and fMLLR. We explored looking at both the relative and absolute gradient metrics, which we referred to as EBW-F and EBW-F Norm respectively, and incorporated these gradient metrics into an HMM framework for BPC recognition on the TIMIT task.

We demonstrated that the EBW-F Norm method outperformed the standard likelihood technique, both when initial models are adapted via MLLR and without adaptation. In addition, we demonstrated the EBW-F Norm metric captures the difference between the likelihood of an observation given the initial model and the likelihood given a model estimated from the current observation being scored, while the likelihood metric just calculates the former. We showed that this extra model re-estimation step is a main advantage of the EBW-F Norm technique. In addition, we explored the benefits of the EBW-F norm metric in noise. Specifically, we demonstrated that, when models are trained on clean speech and used to decode noisy speech, the model re-estimation inherent in the EBW algorithm allows for significant improvement over the likelihood method. Note that, while results in this chapter were only presented for BPCs, similar results were observed for broad acoustic classes (BACs) as well.

Now that we have introduced a technique to robustly recognize broad classes in noise, in the next chapter we explore using this broad class knowledge as a preprocessor for robust landmark detection in a segment-based system. Then, in Chapter 5, we further explore broad class knowledge to aid in island-driven search.

Chapter 4

Comparison of Broad Phonetic and Acoustic Units for Noise Robust Segment-Based Speech Recognition

4.1 Introduction

In Chapter 3 we presented an instantaneous adaptation technique to improve the performance of broad class recognition in noisy speech. In this chapter, we explore using the recognized broad classes as a pre-processor to aid in landmark detection in a segment-based speech recognition system, which we have found to be sensitive to noisy conditions. Specifically, we explore utilizing the spectral distinctness of broad class transitions to help in major landmark placement. We also investigate minor landmark placement specific to each detected broad class.

In addition, we probe whether these broad classes should be phonetically or acoustically motivated, an idea which has been studied for clean speech but is relatively unexplored for noisy speech. Given different noise conditions, for example stationary *vs.* non-stationary or harmonic *vs.* non-harmonic, one approach might be superior to the other.

We explore the phonetic *vs.* acoustic pre-processing approaches on the TIMIT

corpus, where we artificially add a variety of different noise types. We demonstrate that using broad class knowledge as a pre-processor to aid in landmark detection offers significant improvements in noisy speech relative to the baseline spectral change method. In addition, we illustrate under which noise conditions a phonetic *vs.* acoustic method is preferred.

4.1.1 Motivation

A segment-based framework [31], [68] for acoustic modeling, which can also be formulated as a variable frame rate Hidden Markov Model (HMM) [101], has shown success in recognizing speech in noise-free environments. However, we suspect the performance of a segment-based system like SUMMIT may be more sensitive to certain types of noise. This is because SUMMIT computes a temporal sequence of frame-based feature vectors from the speech signal, and performs landmark detection based on the spectral energy change of these feature vectors. These landmarks, representing possible transitions between phones, are then connected together to form a graph of possible segmentations of the utterance. While the spectral method works well in clean conditions [31], [37], the system has difficulty locating landmarks in noise and often produces poor segmentation hypotheses [80]. In [80], we found that noise robustness in SUMMIT could be improved with a sinusoidal model segmentation approach, which represents speech as a collection of sinusoidal components and detects landmarks from sinusoidal behavior. This method offered improvements over the spectral approach at low signal-to-noise ratios (SNRs), but landmark detection was not as robust at high SNRs.

Broad classes, whether motivated along acoustic or phonetic dimensions, have been shown to be salient compared to phonemes [37], and are also prominently visible in spectrograms [102]. Furthermore, in Chapter 3 we demonstrated that broad phonetic classes (BPCs) are robustly identified in noisy conditions. In this chapter, we explore whether the transitions between broad classes, representing large areas of acoustic change in the audio signal, can aid in landmark detection in a segment-based system, particularly in noisy conditions.

A large area of study in speech recognition involves choosing an appropriate set of units when mapping from the acoustic signal to words in the lexicon. The choice of these units is typically not well defined, and subsequently a variety of different mappings have been explored at different levels, i.e., sentence, phrase, word, syllable and phoneme. The mapping at each level has a different amount of acoustic ambiguity [59], which correspondingly affects the performance of the speech recognizer depending on the task at hand.

Because of training data issues, most current speech recognizers do not use word-based models. More specifically, word-based modeling requires having many instances of specific words in the training set in order to adequately train the word models. While word-models have shown great success in small vocabulary tasks, they cannot easily be extended to large vocabulary tasks.

Therefore, nearly all state-of-the-art speech recognition systems employ a sub-word based representation for the mapping between the acoustic signal and words in the lexicon. The most commonly used sub-word units are motivated by phonology and phonetics; e.g., phonemes, syllables, etc. [24]. Phonetic units have the advantage that they are well-defined linguistically, and training of these models is straightforward given the phonetic transcription of an utterance [60]. While the training problems present in word-models are eliminated by using phonetic units, these phonetic units may not always be acoustically distinct and therefore acoustic ambiguity can sometimes be a problem when using phonetic units.

For example, consider the varying acoustic characteristics throughout a diphthong such as /ɑʸ/. To address this issue, researchers have explored the use of *acoustically-*motivated units [5], [60]. For example, in [5], the authors find that using acoustically motivated units offers better performance than using phonetic units on a small vocabulary, speaker independent, read speech task. Furthermore, [60] demonstrates comparable results using both acoustic and phonetic units on a small vocabulary, isolated word recognition task.

While both phonetic and acoustic sub-word approaches have been effectively demonstrated for clean speech, we suspect that their performance may vary under

conditions where the speech signal has been corrupted by noise. For example, in noise conditions which are very harmonic in nature (i.e., babble noise or music), finding acoustically distinct units could pose a challenge since harmonic classes such as vowels appear to look more spectrally similar to non-harmonic classes such as fricatives and closures. Therefore, a phonetic sub-word approach might be preferred. However, in non-stationary noises such as pink and white noise, the harmonics of the speech signal are more prevalent and therefore an acoustic method might be preferred over a phonetic approach. Figure 4-1 shows an example of the word "zero" in both stationary and non-stationary noise conditions. Notice that the formants are more prominent in non-stationary subway noise compared to the stationary babble noise condition.

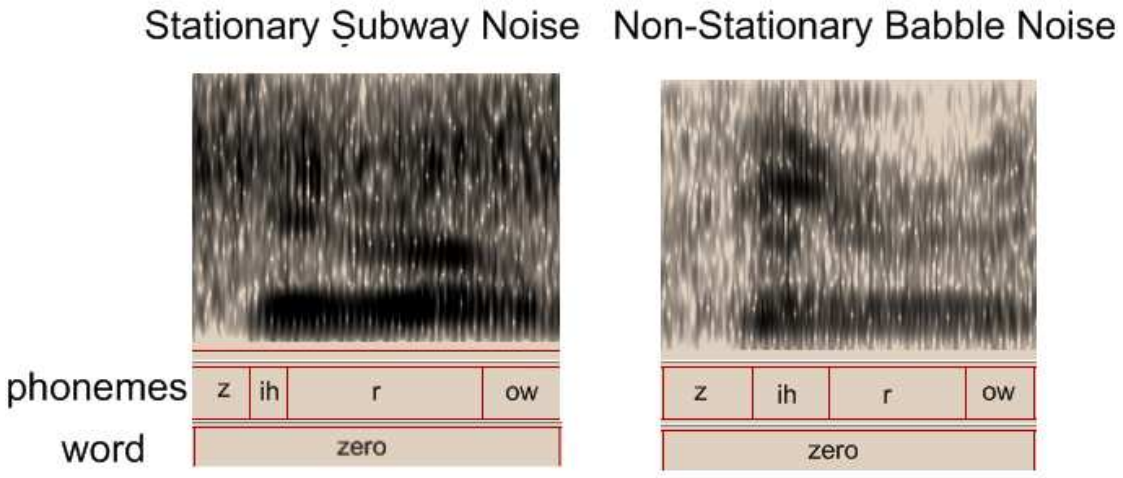

Figure 4-1: Spectrogram of the word "zero" in stationary and non-stationary noise conditions. The corresponding phonemes (i.e., /z/, /ih/, /r/, /ow/) and word label are also indicated.

4.1.2 Proposed Approach

The goal of this chapter is to compare using broad phonetically *vs.* acoustically motivated units as a pre-processor to design a noise robust landmark detection method. A block diagram of the proposed system is given in Figure 4-2.

Specifically, we look at broad classes that are spectrally distinct in noise, as indicated by the broad classes in Figure 4-3. We take advantage of the large acoustic

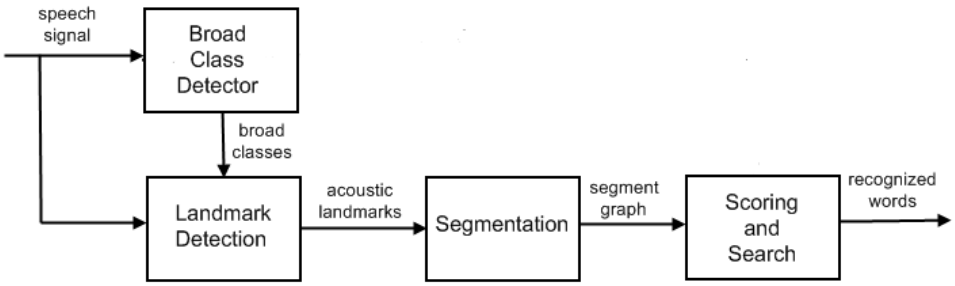

Figure 4-2: Block diagram of broad class pre-processor within segment-based recognition framework of SUMMIT

changes that occur at broad class transitions and thus can aid in landmark detection. Once landmarks are detected, the segment graph is formed and scored similarly to the spectral method [31].

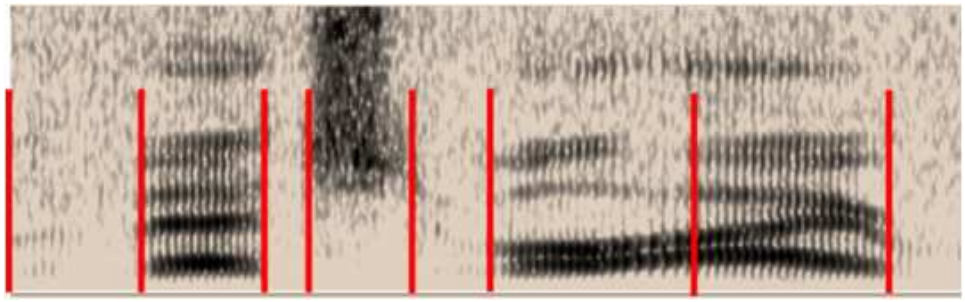

Figure 4-3: Spectrogram of noisy speech in which broad class transitions are delineated by red lines.

First, we compare whether these broad classes should be phonetically *vs.* acoustically motivated. In exploring broad acoustic classes, we also introduce a novel cluster evaluation method to choose an appropriate number of acoustic clusters and evaluate their quality. We evaluate the broad acoustic *vs.* broad phonetic segmentation methods on a noisy TIMIT corpus, exploring pink, speech babble and factory noises. We choose these noises because they differ in their stationarity and harmonic properties, allowing us to compare the behavior of broad phonetic *vs.* broad acoustic units across different types of noise. We find that both the acoustic and the phonetic segmentation methods have much lower error rates than the spectral change and the sinusoidal methods across all noise types. Finally, we observe that the acoustic method has

much faster computation time in stationary noises, while the phonetic approach is faster in non-stationary noises.

4.1.3 Overview

The remainder of this chapter is organized as follows. In Sections 4.2 and 4.3, we describe our broad phonetically- and acoustically- derived broad classes, respectively. Section 4.4 describes our landmark detection and segmentation algorithm using these broad class pre-processors. Section 4.5 presents the experiments performed, followed by a discussion of the results in Section 4.6. Finally, Section 4.7 summarizes the work in the chapter.

4.2 Broad Phonetic Units

[24] argues that a phoneme is the smallest phonetic unit in a language to distinguish meaning. Generally phonemes which belong to the same manner class [87] have similar spectral and temporal properties and can be categorized as belonging to the same broad phonetic class (BPC) [36], while phonemes in different BPCs are spectrally distinct. One representation of these BPCs is vowels/semi-vowels, stops, weak fricatives, strong fricatives, nasals, closures and silence [37]. In phonetic classification experiments on the TIMIT corpus [37], it was shown that almost 80% of misclassified phonemes belonged to the same BPC as the substituted phonemes. These BPCs have been shown to be relatively invariant in noise [82], motivating us to define them as our broad phonetic units.

4.3 Broad Acoustic Units

4.3.1 Learning of Broad Acoustic Units

We learn broad acoustic classes (BACs) from acoustic correlates in the audio signal. The process of learning acoustic units involves a segmentation of the utterance into

quasi-stationary sections followed by a bottom-up clustering [60]. We define our segmentation from the underlying phonetic transcription. Thus, instead of using the underlying phonemes to define BPCs, we learn BACs from acoustic correlates of these phonemes. The segments are then clustered in a bottom-up fashion similar to the clustering method described in [30].

Our first step in agglomerative clustering is to pre-cluster segments using an iterative nearest-neighbor procedure to form a set of seed clusters. Each segment is represented by a feature vector averaged across the entire segment. Then for each feature vector, we compute the zero-mean Euclidean distance from the vector to each cluster mean. If the closest distance falls below a specified threshold, the vector is merged into the existing cluster. Otherwise, a new cluster is formed. After the clustering is complete, all clusters with less than 10 components are merged into one of the closest existing clusters. This step ensures that each cluster has adequate data coverage. Here the distance threshold is chosen to maximize the number of pre-clusters.

After the pre-clustering, stepwise-optimal agglomerative clustering [22] is performed on the seed clusters. In this method, with each iteration the two clusters which cause the smallest increase in distortion are merged. We define distortion between two clusters D_i and D_j by the sum-of-squared-error criterion, as given in Equation 4.1. Here m is the cluster mean and n is the number of constituents of a cluster. We chose this measure of distortion since it tends to favor merging small clusters with larger clusters rather than merging medium-sized clusters, allowing for clusters to have good data coverage.

$$d(D_i, D_j) = \sqrt{\frac{n_i n_j}{n_i + n_j}} ||\mathbf{m}_i - \mathbf{m}_j|| \tag{4.1}$$

This stepwise-optimal agglomerative clustering method produces a hierarchical tree-like structure of acoustic clusters, where each level of the tree indicates a different grouping of clusters. After developing a method to learn the acoustic structure, it is necessary to evaluate a meaningful number of clusters from the tree structure.

4.3.2 Cluster Evaluation with V-Measure

Evaluation measures for supervised clustering methods include *homogeneity* (i.e., purity, entropy), which requires that the clusters contain only data points which are members of a single class, as well as *completeness*, which requires that all data points that are members of a given class are elements of the same cluster. One such recent measure, known as the V-measure [78], derives a clustering metric which evaluates cluster quality by observing the tradeoff between homogeneity and completeness. Evaluation metrics for unsupervised clustering are a bit more difficult, as labels for clusters are not known *a priori*. To evaluate the unsupervised BACs, we slightly alter the V-measure formulation. Below we describe the traditional V-measure approach and then introduce our approach to using the V-measure for unsupervised clustering.

Traditional V-Measure

Assume we have a set of classes C and clusters K. If the number of clusters K is known *a priori*, the conditional entropy of the classes given the clusters, $H(C|K)$, is defined as:

$$H(C|K) = -\sum_{k=1}^{K} p(k) \sum_{c=1}^{C} p(c|k) \log p(c|k) \tag{4.2}$$

Instead of looking at the raw conditional entropy $H(C|K)$, the entropy is normalized by the maximum reduction in entropy the clustering algorithm could provide without any prior cluster information, namely $H(C)$, given by:

$$H(C) = -\sum_{c=1}^{C} p(c) \log p(c) \tag{4.3}$$

Using Equations 4.2 and 4.3, homogeneity is defined as:

$$homg = \begin{cases} 1 - H(C|K)/H(C) & \text{if } H(C) \neq 0 \\ 1 & \text{if } H(C) = 0 \end{cases}$$

Similarly, completeness is computed by looking at the conditional entropy of the

clusters given the classes $H(K|C)$:

$$H(K|C) = -\sum_{c=1}^{C} p(c) \sum_{k=1}^{K} p(k|c) \log p(k|c) \tag{4.4}$$

And the worst case value of $H(K|C)$ is $H(K)$, given by:

$$H(K) = -\sum_{k=1}^{K} p(k) \log p(k) \tag{4.5}$$

Using these metrics, completeness is defined as follows:

$$comp = \begin{cases} 1 - H(K|C)/H(K) & \text{if } H(K) \neq 0 \\ 1 & \text{if } H(K) = 0 \end{cases}$$

The quality of the clustering solution is defined by the V-measure [78], which computes the harmonic mean between homogeneity and completeness as:

$$V_\beta = \frac{(1+\beta) \times homg \times comp}{(\beta \times homg) + comp} \tag{4.6}$$

Here β controls the weight given to completeness *vs.* homogeneity.

Class Similarity V-Measure

The above V-measure assumes that each class C is labeled. In our work the only labeled classes are the underlying phonemes, and therefore for simplicity we choose these as our classes. However, our goal is to find a set of broad spectrally distinct classes in an unsupervised manner, which are subsequently unlabeled. Therefore, to use the V-measure to learn an appropriate set of clusters, we assume that cluster k is made up of some true classes c^* which are hidden. Ideally we would like cluster k to be composed of classes which are acoustically similar. We cannot observe these true classes c^*. However, we estimate the distribution $p(c^*|k)$ by the classes our clustering algorithm assigns to cluster k. We also define the similarity between each

of the true classes c^* and all other hypothesized classes c by $p(c|c^*, k)$. Making the assumption that c and k are conditionally independent given c^*, then by definition $p(c|c^*, k) \approx p(c|c^*)$. $p(c|c^*)$ measures the probability a phoneme being hypothesized as c given that the true phoneme is c^*. The probability $p(c|c^*)$ is calculated by running a phonetic classification experiment on the TIMIT development set, as described in [37], and counting the number of times phoneme c^* is confused with phoneme c. Finally, to calculate $p(c|k)$ we sum over all the hidden variables c^*, as given by Equation 4.7.

$$p(c|k) = \sum_{c^*} p(c|c^*, k)p(c^*|k) = \sum_{c^*} p(c|c^*)p(c^*|k) \tag{4.7}$$

Intuitively, to calculate $p(c|k)$, Equation 4.7 computes the probability of each of the true classes assigned to cluster k (i.e., $p(c^*|k)$) and weights them by the similarity of these true classes c^* to class c (i.e., $p(c|c^*)$). $p(c|k)$ is computed in the same manner by observing the similarity between c and c^* as:

$$p(k|c) = \sum_{c^*} p(c^*|c, k)p(k|c^*) = \sum_{c^*} p(c^*|c)p(k|c^*) \tag{4.8}$$

Again the confusion probability $p(c^*|c)$ is derived from a phonetic classification confusion matrix. Equations 4.7 and 4.8 give more weight to classes which are spectrally similar, and Equations 4.2 and 4.4 are modified to reflect this as well.

4.3.3 V-Measure Cluster Analysis

In this section, we analyze the behavior of the V-measure and corresponding learned BACs across different noise types and SNRs.

Choosing Number of Clusters

First, we discuss how the V-measure allows us to choose an optimal number of clusters. Figure 4-4 left shows the V-measure as the number of clusters is varied from 2 to 50. The dendrogram on the right indicates the hierarchical clustering formed by merging classes in a bottom-up fashion. Notice that the class similarity V-measure shows a

broad peak around 8 clusters which represents the clustering solution which gives the best tradeoff between completeness and homogeneity. This is the number we choose as the optimal cluster number.

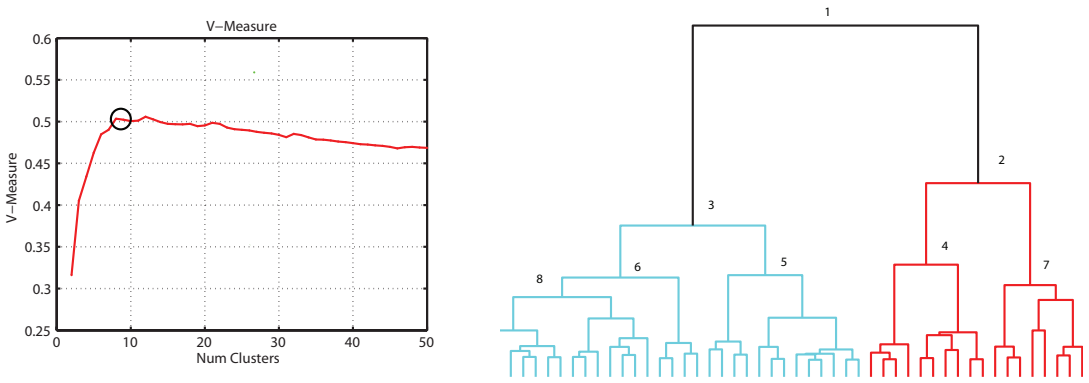

Figure 4-4: The figure on the left displays the number of clusters as the V-measure is varied. The peak in the V-measure, representing the optimal number of clusters, is indicated by a circled. The right figure illustrates a dendrogram formed by merging different clusters in a bottom-up fashion at different levels. The numbers at each level indicate the number of clusters at that level.

Next, we explore the benefits of our novel class similarity measure for better cluster selection at lower SNRs. Figure 4-5 shows the V-measure with and without the class similarity measure for (a) 30dB and (b) 10dB of babble noise as the number of clusters is varied.

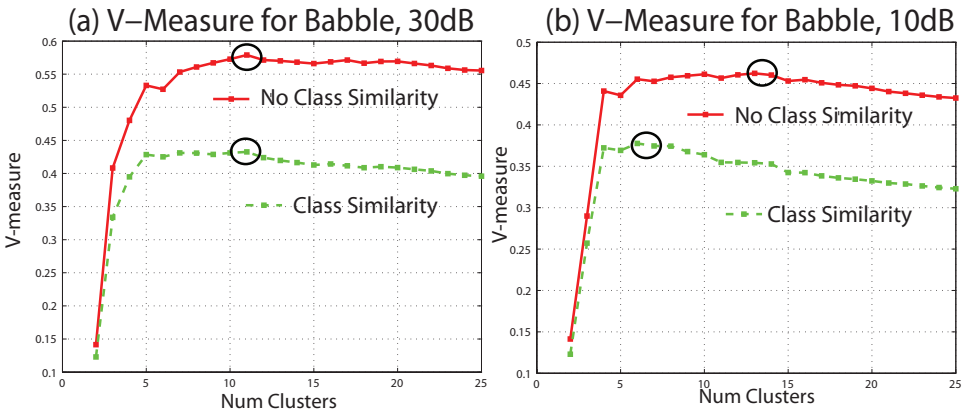

Figure 4-5: V-measure vs. number of clusters, with and without class similarity, at 30dB and 10dB of babble noise. The optimal number of clusters for each metric and noise level is indicated by a circle.

First, both measures show a broad peak, which defines the range for the optimal cluster number. In plot (a), both metrics peak at 11 clusters. In plot (b), the class similarity V-measure peaks at 6, while the other condition peaks at 13. As the SNR decreases and the number of confusions between broad classes increases, intuitively the number of broad classes should decrease. While the V-measure without class similarity seems to find a reasonable number of clusters at 30dB, the increase in clusters at 10dB indicates the clusters are not acoustically distinct. However, when similarity information is utilized the clusters are chosen based on spectral closeness, as reflected by a decrease in the number of clusters with decreasing SNR. While only babble is shown here, similar V-measure trends were observed for other noise types.

Choice of β

Referring to Equation 4.6, β controls the weight given to the model complexity completeness term and $\beta > 1$ weights completeness more than homogeneity. Figure 4-6 shows the behavior of the V-measure for three different values of β. We see that the larger we make β, the more weight given to model complexity and the smaller the number of optimal clusters.

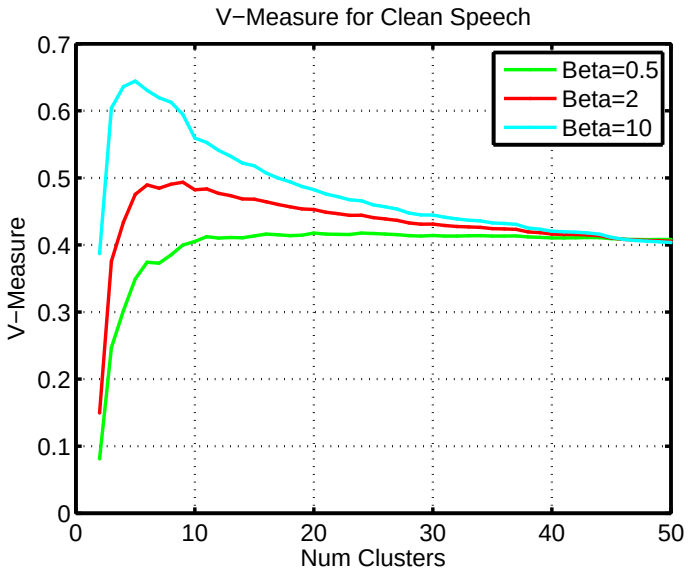

Figure 4-6: Behavior of V-measure vs. number of clusters for different values of β

In learning BACs, we justify our choice of β by defining our objective to learn

as many acoustic clusters such that the cluster distribution within the majority of individual phonemes is greater than 50%. In other words, 50% of the members within the majority of a specific phoneme will fall into the same BAC. We utilize this information in our choice of β because ideally all members of a specific phoneme should fall within the same BAC.

This idea is illustrated more clearly by Figure 4-7, which displays the cluster distribution within each phoneme for 7 learned BACs. This distribution was calculated by counting the number of training tokens within a phoneme group assigned to a specific class, and normalizing across all training tokens within that phoneme. Each distinct color in the figure represents a specific cluster and this colored bar within a phoneme group indicates the percentage of tokens within that phoneme assigned to that cluster. So, for example, more that 95% of phoneme tokens /h#/, the rightmost entry, belong to one cluster.

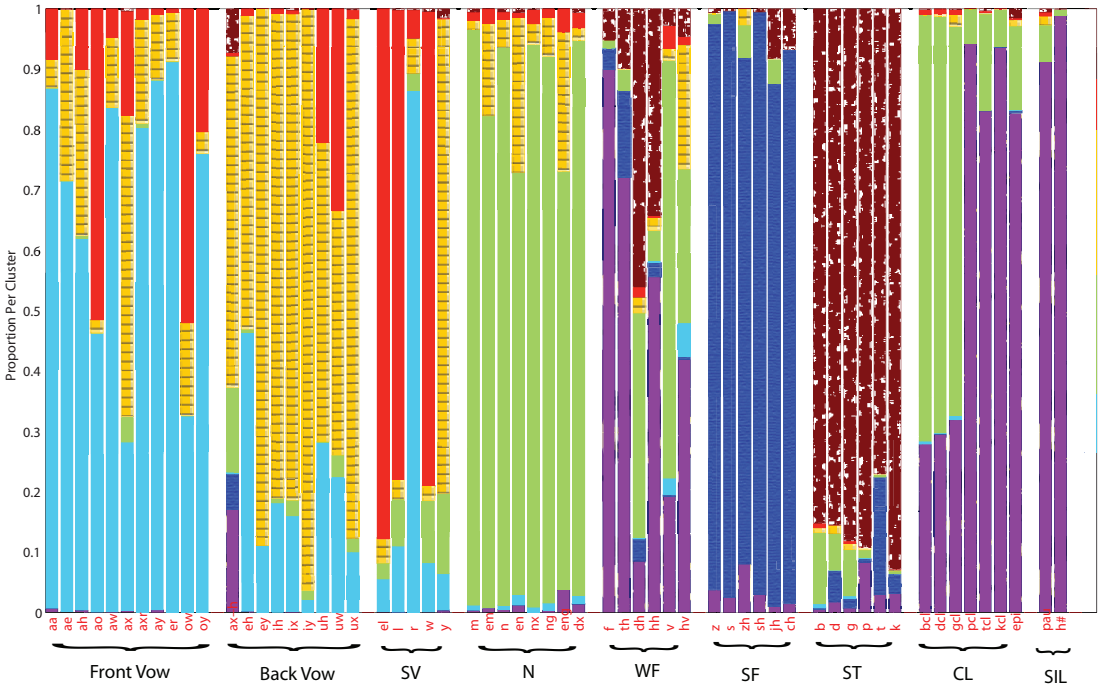

Figure 4-7: Cluster distribution for each phoneme. Each distinct color in the figure represents a specific cluster and this colored bar within a phoneme group indicates the percentage of tokens within that phoneme assigned to that cluster.

Broad Class Behavior in Noise

Now that we have described the behavior of the V-measure, we next turn to analyzing the learned clusters for each noise type. To gain a better understanding for the learned BACs, we first analyze the confusion of phonemes in the vowel class with all other phonemes. Again, these confusions are obtained from a phonetic classification experiment on the TIMIT development set, as described in [37], for each SNR and noise type. The confusions are calculated by counting all phonemes in TIMIT confused with each individual phoneme which is mapped to the vowel class. This mapping is defined in Table 3.1 from Section 3.4. We also explore this confusion for phonemes in the fricatives class as well. These two classes are chosen since the behavior of phoneme confusions in noise within these two classes is quite different.

Figure 4-8(a) shows the confusions for vowels in each noise type and SNR, normalized by the maximum vowel confusions over all noises. Notice that vowels have the least amount of confusions in stationary, non-harmonic pink noise, implying that harmonics are well-preserved in pink compared to non-stationary, non-harmonic babble, which has the most number of confusions. Non-stationary, non-harmonic factory noise retains harmonics better than babble but not as well as pink. Figure 4-8(b) plots the normalized confusions for fricatives, and indicates that fricatives have a common amount of confusions and thus behave similarly in all noises. This same trend is true for other non-harmonic broad classes such as stops and closures.

Cluster Evaluation in Noise

With a better understanding of broad class behavior in noise provided in the previous section, Table 4.1 shows the number of learned BACs for each SNR and noise type. In addition, to further compare the learned clusters in the three noise types, we analyze the quality of hypothesized clusters within various broad classes, i.e., vowels, weak fricatives, strong fricatives, etc. Figure 4-9 illustrates the distribution of various learned clusters within each broad class for clean speech. Each color represents a different cluster, while each pie slice color within a specific broad class illustrates the

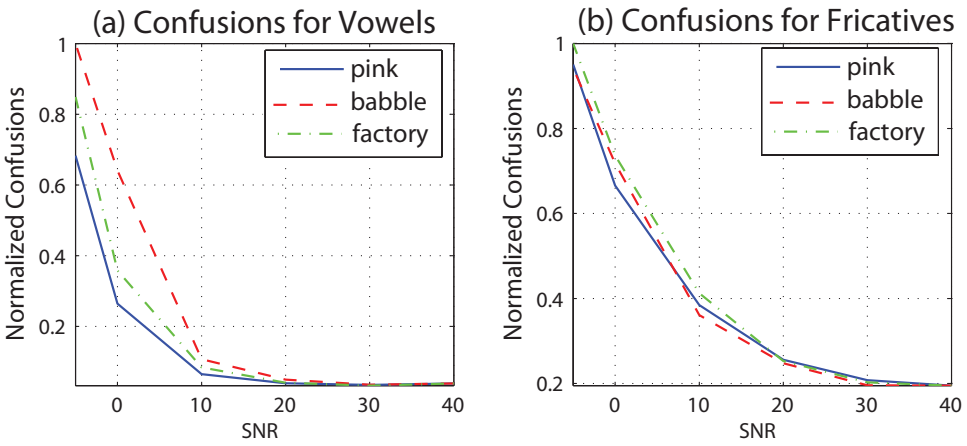

Figure 4-8: Normalized Confusions in Noise for Phonemes in Vowel and Fricative Classes.

percentage of that broad class that belongs to a specific cluster.

SNR	Clean	30dB	20dB	10dB	0dB	-5dB
Pink-Number of Clusters	11	12	7	6	5	4
Babble-Number of Clusters	11	10	6	5	5	3
Factory-Number of Clusters	11	10	7	6	4	4

Table 4.1: Number of Clusters Across SNRs

In clean speech, eleven broad acoustic clusters are learned. These clusters constitute the following main classes: front vowels, back vowels, /r/-like phonemes (i.e., /r/, /axr/, /er/), other semivowels, nasals, weak fricatives, strong fricatives, stops, voiced closures, unvoiced closures, silence. Figure 4-9 also indicates that there is quite a bit of homogeneity within each cluster, as each broad class tends to be concentrated in one main cluster. Notice from the figure that the vowel class has two main clusters, namely *clust1* and *clust3*, illustrating the split between front vowels and back vowels. In addition, notice the split between retroflexed semivowels (/r/-like phonemes) belonging to *clust9*, and other semivowels in the semivowel class, mainly concentrated in *clust5*.

As the noise level increases, the number of clusters decreases. The first merge occurs with the nasals, weak fricatives and closures being merged by 20dB, and then stops by 10dB. This is true for all noise types since many of the non-harmonic classes behave similarly in noise conditions, as demonstrated in Figure 4-8(b). Figure 4-

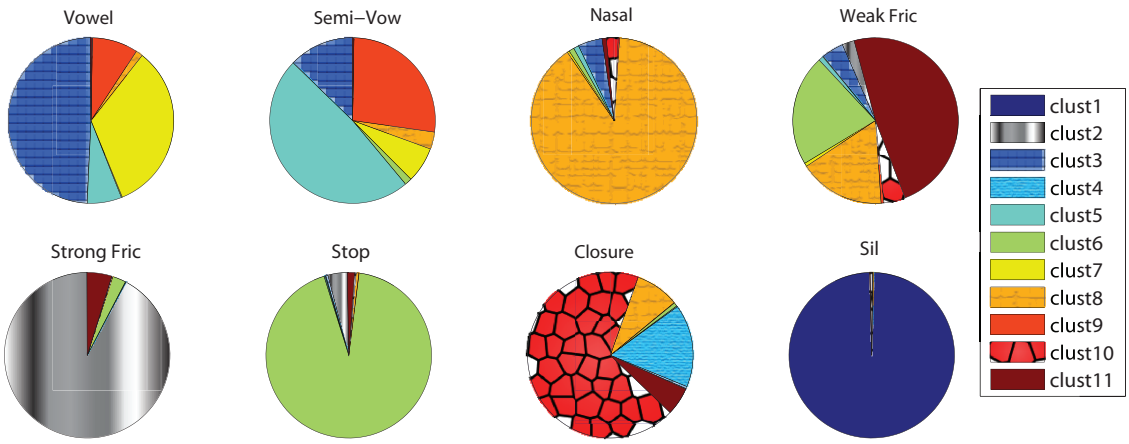

Figure 4-9: Distribution of learned clusters within each broad class for clean speech. Each color represents a different cluster, as illustrated by the legend, while each pie slice color within a specific broad class illustrates the percentage of that broad class that belongs to a specific cluster.

10 shows the cluster distribution within the closure class for each SNR and noise type. Notice that at each SNR, the cluster distribution is very similar for all three noise conditions, supporting the claim that closures behave similarly in different noise types. This fact was also verified for nasals and weak fricatives.

Between 20dB and 10dB, babble noise has one less class compared to both factory and pink noises. Because vowels are well preserved in pink and factory noise compared to babble, as shown by Figure 4-8(a), we find that separate clusters are formed for front and back vowels for these two noises. However, in babble noise just 1 cluster is formed for the vowel class. This is further verified in Figure 4-11, which shows the cluster distribution for vowels as a function of SNR and noise type. Notice that for lower than 30dB SNR, all vowels in babble noise fall into one cluster, with the exception of 0dB SNR, whereas two clusters are learned in pink and factory noises.

Finally, at -5dB, both pink and factory noises have 4 main clusters: vowels, strong fricatives, silence and the merged classes of weak fricatives, nasals, stops and closures. Babble noise has one less cluster, as the vowels are now merged with the weak fricatives, nasals, stops and closures.

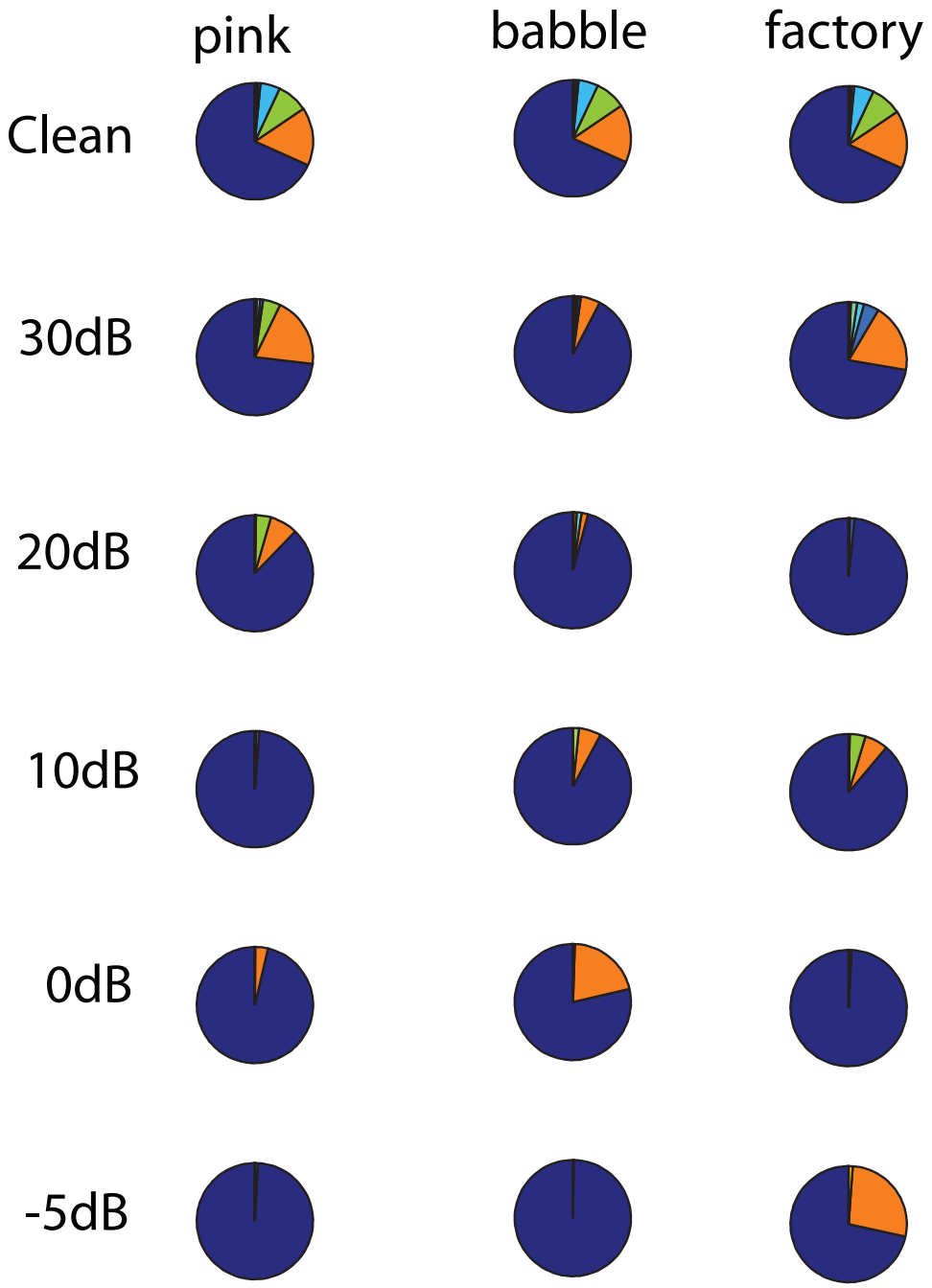

Figure 4-10: Distribution of learned clusters as a function of SNR and noise type for the closure class. Each color represents a different cluster, while each pie slice color within a circular pie illustrates the percentage of a specific cluster that belongs to the closure class.

4.4 Segmentation with Broad Classes

In the previous section, we described a method to learn broad classes (i.e., BPCs, BACs) in noise. Now, in this section, we examine how these broad classes can be

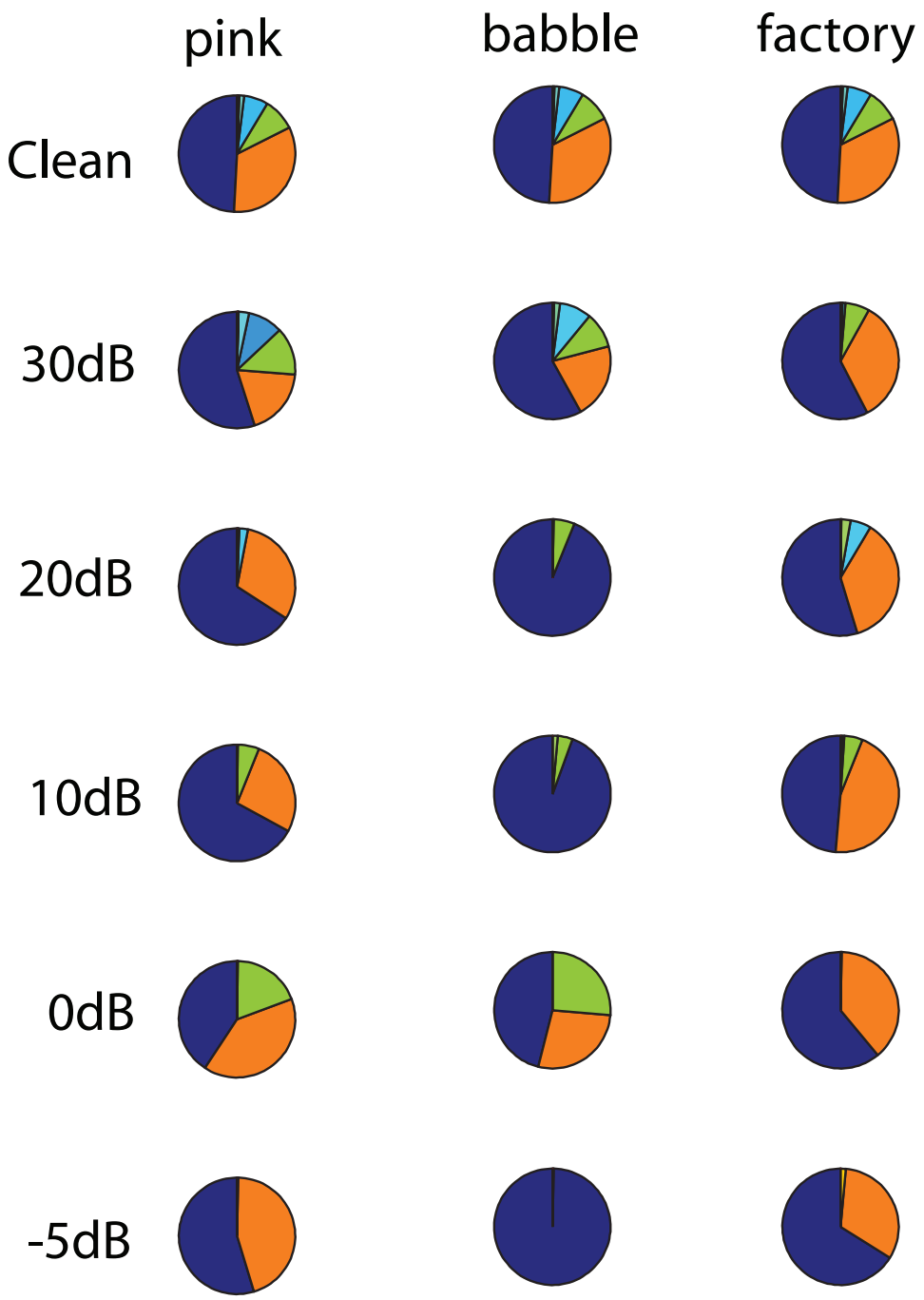

Figure 4-11: Distribution of learned clusters as a function of SNR and noise type for the vowel class. Each color represents a different cluster, while each pie slice color within a circular pie illustrates the percentage of a specific cluster that belongs to the vowel class.

used to design a robust landmark detection and segmentation algorithm for speech recognition. A block diagram of the system is shown in Figure 4-2. Given an input

utterance, we first detect the broad classes in the signal. Transitions between broad classes represent the places of largest acoustic change within an utterance. Recall that major landmarks in the baseline spectral change method are placed where the spectral change exceeds a specified global threshold. Since this threshold is static, oftentimes in noisy speech major landmark boundaries are poorly detected. Therefore, we explore using broad class transitions to aid in major landmark placement.

More specifically, we run multiple spectral change segmentations with different major landmark thresholds in parallel. We define a reasonable segmentation within an utterance as one where the broad class transitions are detected by a major landmark threshold setting. We can quantify how well a specific major landmark threshold detects a broad class (BC) and how many false alarms it produces by precision and recall, defined as:

$$\text{Precision} = \frac{\text{\# detections}}{\text{total \# true BC landmarks}}, \ \text{Recall} = \frac{\text{\# detections}}{\text{\# hyp. major landmarks}} \quad (4.9)$$

and can combine this into an F-measure score as:

$$\text{F-measure}_\alpha = \frac{(1+\alpha) \times \text{Precision} \times \text{Recall}}{\alpha \times \text{Precision} + \text{Recall}} \quad (4.10)$$

Here α controls how much emphasis we place on major landmark false alarms versus missed broad class. Since false alarms within a broad class could represent potential transitions between phonemes while a missed broad class detection may indicate not detecting acoustically distinct transitions, we allow for greater false alarms and fewer missed landmarks. Correspondingly, we tune α to be large to weight recall more than precision.

Therefore, for each broad class transition, we look at the major landmark setting which maximizes the F-measure. Formulating this idea mathematically, for a given utterance, let $S = \{s_1 \dots s_N\}$ represent a particular major landmark segmentation setting, T the list of all segmentation settings and $BC = \{BC_1 \dots BC_J\}$ the list of hypothesized broad class transitions. At each BC_i transition, the segmentation

parameter setting S_i^* which has the highest F-measure is the optimal segmentation setting chosen. In other words:

$$S_i^* = \{\arg\max_{S \in T} [\text{F-meas}(S|BC_i)]\} \tag{4.11}$$

Since each broad class conveys a distinct acoustic characteristic, we next look at setting a fixed density of minor landmarks specific to each broad class. For example, stops are more acoustically varying than vowels, and therefore we expect stops to have a greater density of minor landmarks. Finally, major and minor landmarks are connected together through an explicit set of segmentation rules.

In our connectivity method, we explore a partial-connectivity method similar to one explored in [79]. First we label each broad class major landmark as either hard or soft. Landmarks in which the spectral change across the landmark is above a specified threshold are defined to be hard landmarks, while soft landmarks have a spectral change below this threshold. Minor landmarks can be connected to other minor landmarks across soft major landmarks. However, minor landmarks cannot be connected across hard major landmarks. In addition, each major landmark is connected to the next two consecutive major landmarks, as in the regular spectral change method. Figure 4-12 illustrates the connectivity using hard and soft major landmarks more explicitly, while Figure 4-13 shows a graphical display of the partial-connectivity method from the SUMMIT recognizer.

Figure 4-14 shows a complete picture of the segmentation steps discussed in this section. First, a series of broad classes are detected. These broad classes are used as anchor points to aid in major landmark detection. Within each broad class, a set of minor landmarks are placed specific to that broad class. Landmarks are then connected together using the partial-connectivity method to form a segment network. A Viterbi search is then performed through this segment graph to find the best set of sub-word units.

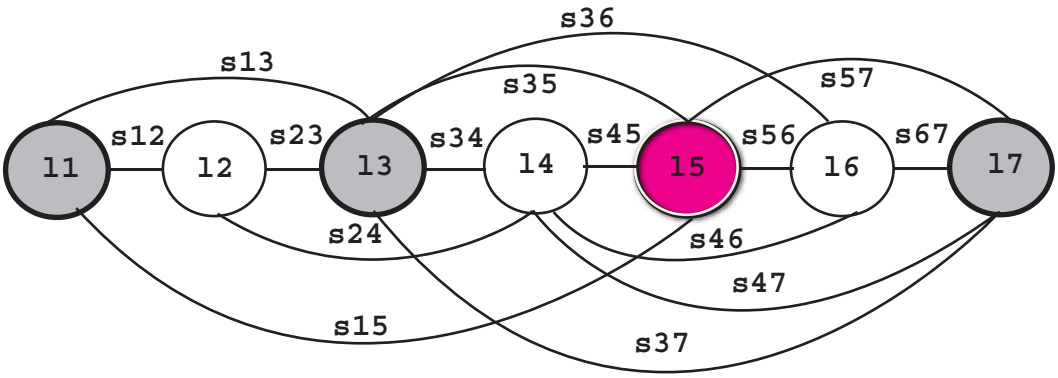

Figure 4-12: Segment network for Partial-Connection technique. Hard major landmarks are indicated by light shaded circles while soft major landmarks are indicated by dark shaded circles. Circles which are not shaded correspond to minor landmarks. Minor landmarks l_i are connected across soft major landmarks to other landmarks l_j which fall up to two major landmarks away via segments s_{ij}. However, minor landmarks cannot be connected across hard major landmarks. In addition, each major landmark is connected to the next two major landmarks.

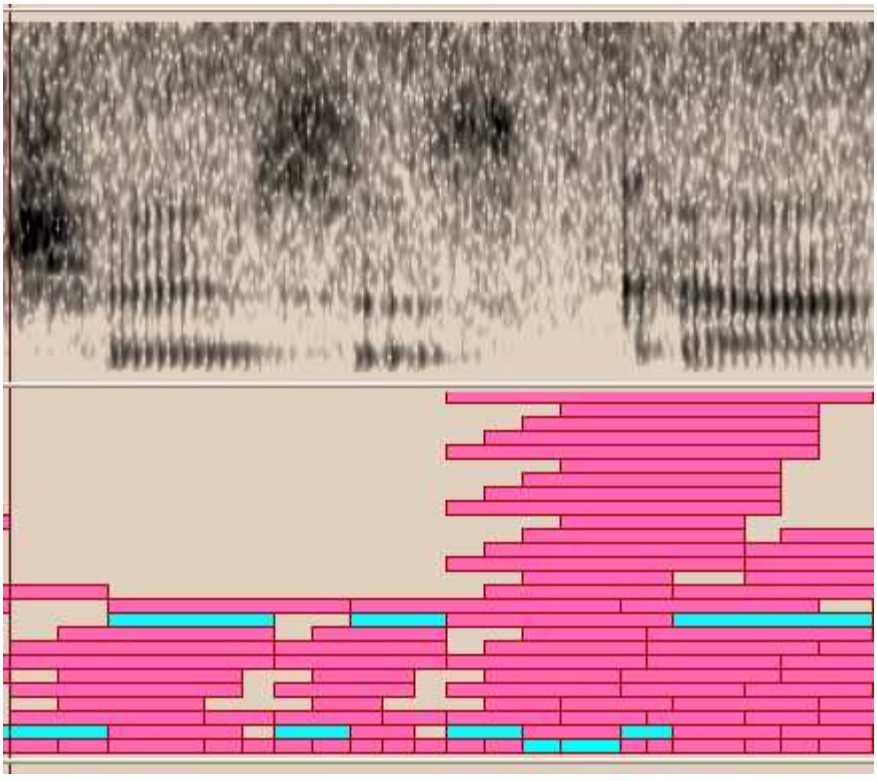

Figure 4-13: Graphical display from the SUMMIT recognizer. The top panel illustrates a spectrogram of the speech signal. The bottom panel shows the segmentation network for the partial connectivity method. The darker colored segments illustrate the segmentation with the highest recognition score during search.

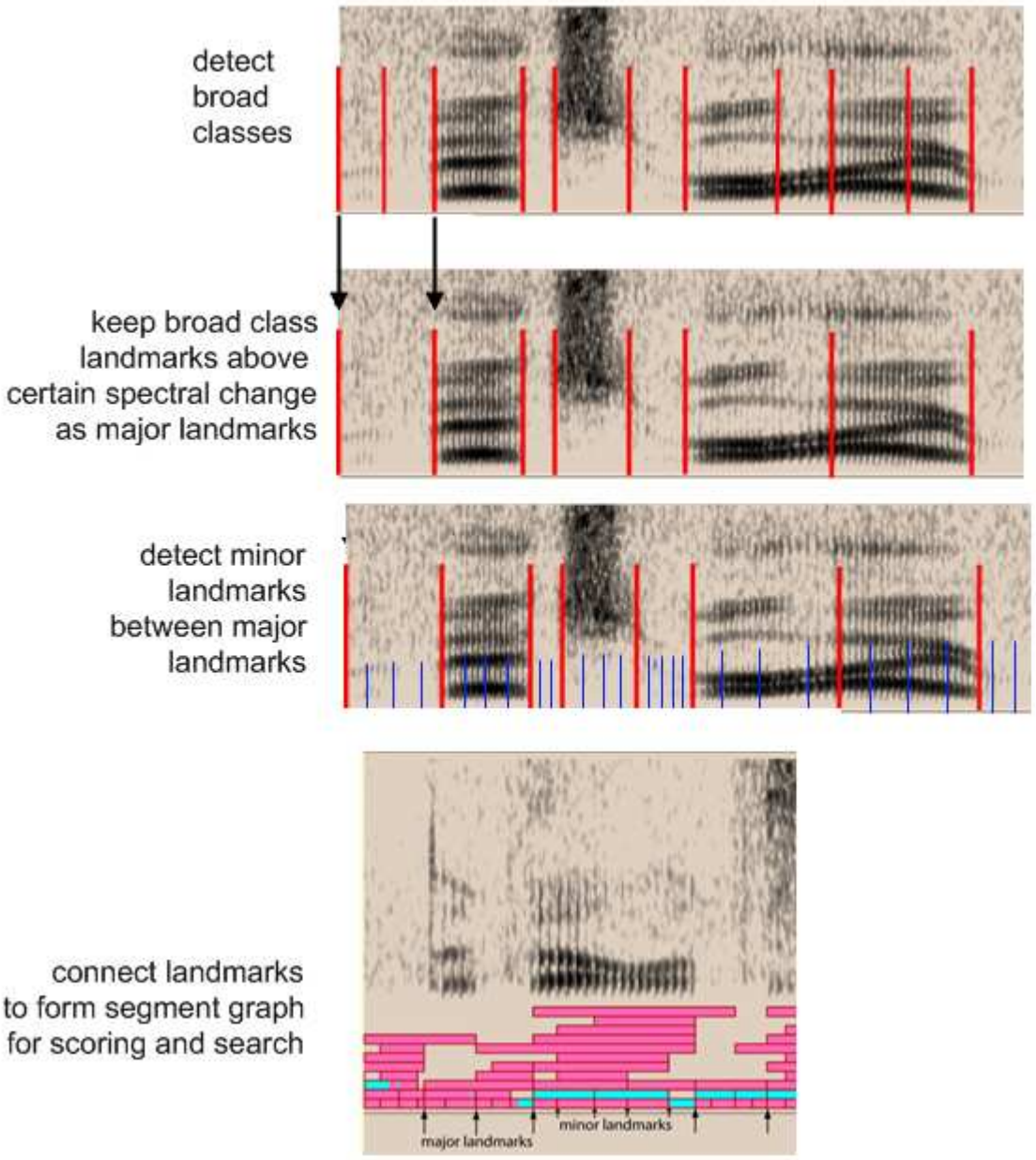

Figure 4-14: Steps of the broad class segmentation method. The first panel shows a spectrogram with broad classes delineated by lines. The second panel illustrates how broad classes are used for major landmark placement. The third panel depicts using broad classes for minor landmark placement, as indicated by the small lines. Finally, the last panel shows the segment network in SUMMIT formed using the partial-connectivity method.

4.5 Experiments

Phonetic recognition experiments are performed on the full 61-phoneme TIMIT corpus described in Section 2.4.1, which offers the benefit of a phonetically-rich context,

in addition to a hand-labeled and time-aligned transcription. We simulate noise on TIMIT by artificially adding pink, speech babble or factory noise, from the Noisex-92 database [93] at SNRs in the range of 30dB to -5dB.

Our experiments explore BPC/BAC units specific to each SNR and noise type. While the number of BPCs is fixed for each condition, the number of BACs varies based on the environment. Each broad class is modeled as a three-state, left to-right context-independent HMM, described in Chapter 3. A unigram language model is used. Broad class models are trained for each SNR and noise type using the TIMIT training set. For a given utterance, broad classes are detected with an HMM, and their transitions are used to aid in landmark detection, as discussed in Section 4.4. Phonetic recognition is then performed in SUMMIT using context-dependent triphone acoustic models to score and search the segment graph for the best recognition hypothesis. Acoustic models are trained specific to each SNR, noise type and segmentation method (i.e., BAC, BPC, sinusoidal and spectral change techniques). A bigram language model is used for phonetic recognition experiments. Recognition results are reported on the full test set.

First, we compare the phonetic error rate (PER) of the BAC and BPC segmentation methods to the baseline sinusoidal and spectral change approaches. Secondly, we analyze the minor landmark settings for the broad class segmentation method. Next, we investigate the recognition computation time of the BAC and BPC techniques for all utterances in the test set. This computation time is defined to be the total time, in seconds, spent strictly during the scoring and search phase. Finally, we explore the performance of the BAC method having a fixed number of acoustic clusters.

4.6 Results

4.6.1 Segmentation Error Rates

Table 4.2 shows the PER for each SNR, averaged across the three noise types, for the spectral change, sinusoidal, BPC and BAC methods. The best performing method

at each SNR is indicated in bold. Note that the number of BACs at each SNR for pink, babble and factory noise is also indicated in parentheses in the BAC column. In addition, Figure 4-15 shows the average duration difference between true phoneme boundaries and hypothesized landmarks for each method, also averaged across the three noise conditions. The durational difference is the absolute time difference between each true phonetic boundary in the TIMIT corpus and the landmark closest to this boundary. First, decreasing the SNR results in rapid degradation in performance for the spectral change method, as well as a large time deviation from the true phonetic boundaries. While the sinusoidal model approach is more robust at lower SNRs compared to the spectral change method, it does not perform as well at high SNRs, as landmarks are not as robust. The BAC and BPC methods provide the best performance of all methods, and have the most robust landmarks, as shown in Figure 4-15. A Matched Pairs Sentence Segment Word Error (MPSSWE) significance test [29] also indicates that the BAC and BPC results are statistically significant compared to the spectral change and sinusoidal methods, though not compared to each other. The only exception to this is -5dB of babble noise, where harmonics are very poorly preserved, leading to poor BACs. While the performance of these two methods is fairly similar across noise conditions, Section 4.6.3 will illustrate that their computation times are different.

TIMIT Average Phonetic Error Rates				
db	spec	sine	bpc	bac
Clean	28.7	30.6	27.7	**27.3** (11,11,11)
30dB	29.2	31.3	28.4	**28.3** (12,10,10)
20dB	32.5	34.3	**31.5**	31.7 (7,6,7)
10dB	42.1	43.3	41.1	**40.9** (6,5,6)
0dB	70.7	59.4	**57.9**	58.0 (5,5,4)
-5dB	91.8	68.5	**67.3**	69.6 (4,3,4)
Average	49.2	45.5	**42.3**	42.6

Table 4.2: PERs for Segmentation Methods on TIMIT Test Set, averaged across Pink, Babble and Factory noises at each SNR. The best performing method at each SNR is indicated in bold. In addition, the BAC method indicates the number of clusters at each SNR for pink, babble and factory noise in parentheses.

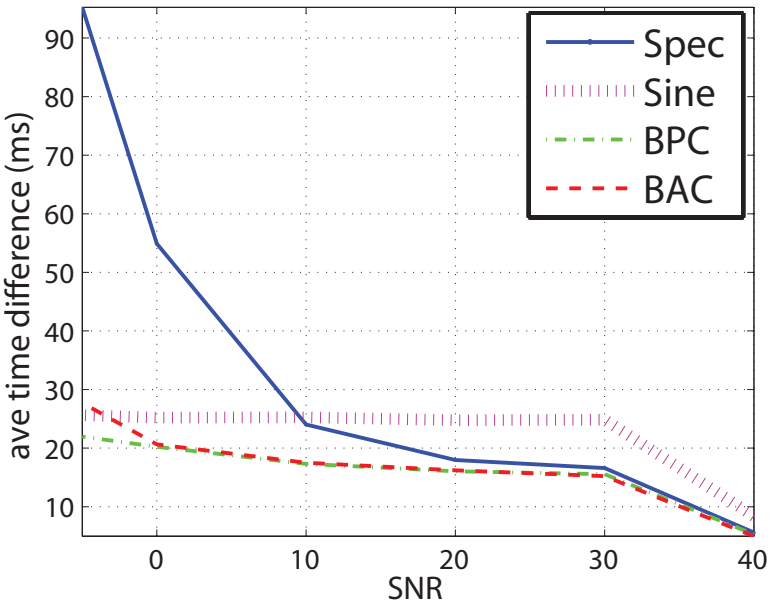

Figure 4-15: Average Time Difference between True Phonemes and Hypothesized Landmarks as a function of SNR for different segmentation methods. Results are averaged across the three different noise types.

4.6.2 Broad Class Landmark Tuning

Next, we explored the benefits to tuning the minor landmarks specific to each broad class. Table 4.3 shows the PER for the BPC Method in pink noise, with and without minor landmark tuning. Again, the best performing technique at each SNR is outlined in bold. Note that these results are shown for the TIMIT development set. One can observe that using BPC information to tune the minor landmarks per class has significant improvement compared to just tuning major landmarks. While not shown, similar results also hold for the BAC segmentation method.

4.6.3 Segmentation Computation Time

In this section, we use the V-measure to investigate the quality of the hypothesized BPC and BAC units, and show the direct correlation to computation time. The entire recognition process, as illustrated in Figure 4-2 involves detecting broad classes, creating a segmentation network, and finally performing a scoring and search through this network. In this thesis, we only explore computation time for the BPC and BAC

db	BPC Segmentation, Major Landmarks	BPC Segmentation, Major & Minor Landmarks
Clean	27.9	**27.1**
30dB	28.5	**27.5**
20dB	31.8	**31.1**
10dB	41.1	**40.1**
0dB	58.4	**56.0**
-5dB	65.5	**65.8**
Average	42.2	**41.3**

Table 4.3: PER on TIMIT Development Set for BPC Segmentation method comparing major landmark tuning vs. major and minor landmark tuning. Results are shown across different SNRs of pink noise. The best performing method at each SNR is indicated in bold.

methods during the scoring and search phases. We define computation time in this manner since the time to detect broad classes and form the segment graph is similar for the two approaches, and also negligible, compared to final computation time.

To assign a set of labeled classes to the broad units to compute the V-measure, we look at the true underlying phonemes which make up the different BPCs or BACs generated from the TIMIT transcription. Figure 4-16 shows the total V-measure, average V-measure for vowels, and computation time (CPU Time) as a percentage of real time, for the BAC/BPC units in the three noise conditions. Finally, the last column in Figure 4-16 illustrates the relative time difference between the BAC and BPC methods. For example, a relative time difference of 20% means that the BAC method is 20-percent faster relative to the BPC method (or the BPC method is 20-percent slower).

In pink noise, the total V-measure is higher for the BAC method across all SNRs, and gains are made particularly in the vowel class. As illustrated in Figure 4-8(a), pink noise tends to preserve harmonics well, resulting in a higher V-measure and better quality clusters for the BAC technique relative to BPC, which groups all vowels into one class. This leads to a faster CPU time for the BAC method, i.e., roughly between 0 to 20% faster relative to the BPC method. The segment graph in Figure 4-17(a) also indicates that the BAC method has more finer level hypothesized acoustic clusters compared to the BPC method in Figure 4-17(b), resulting in a smaller segment graph

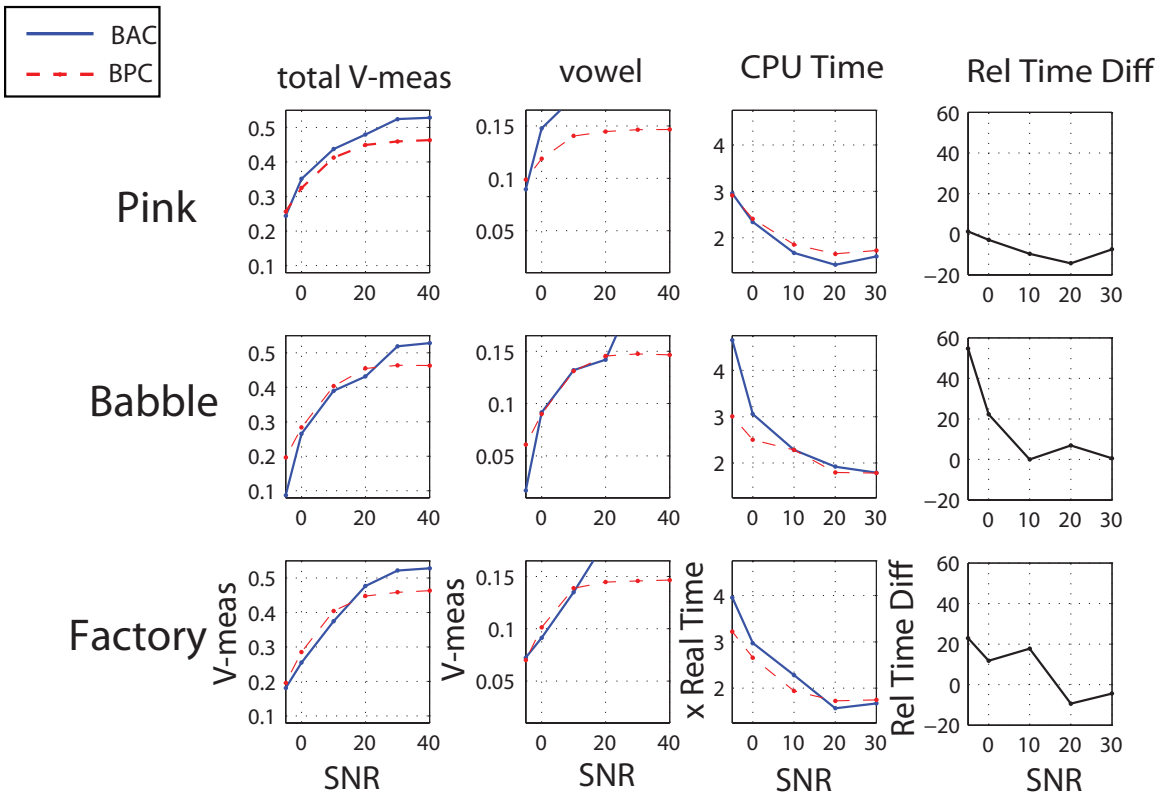

Figure 4-16: V-measures and CPU Times for BAC and BPC methods across different noise types and SNRs.

and faster CPU time.

In babble noise, harmonics are not well preserved at lower SNRs. This leads to greater confusions between broad classes, resulting in fewer BACs. Thus, in babble the BPC method has a higher V-measure and faster CPU time at lower SNRs. In fact at -5dB, the BAC method is roughly 60% slower relative to the BPC method.

Finally, for factory noise, at high SNRs, harmonics are well-preserved and the BAC method has a higher V-measure and faster CPU time. As the SNR decreases, harmonics are not as well preserved in factory compared to pink and the number of BACs decreases. Thus, the BPC method has finer level BPCs and is roughly 20% faster relative to the BAC method at lower SNRs. This is further confirmed by the smaller segment graph for BPC in Figure 4-17(c) compared to BAC in 5(d).

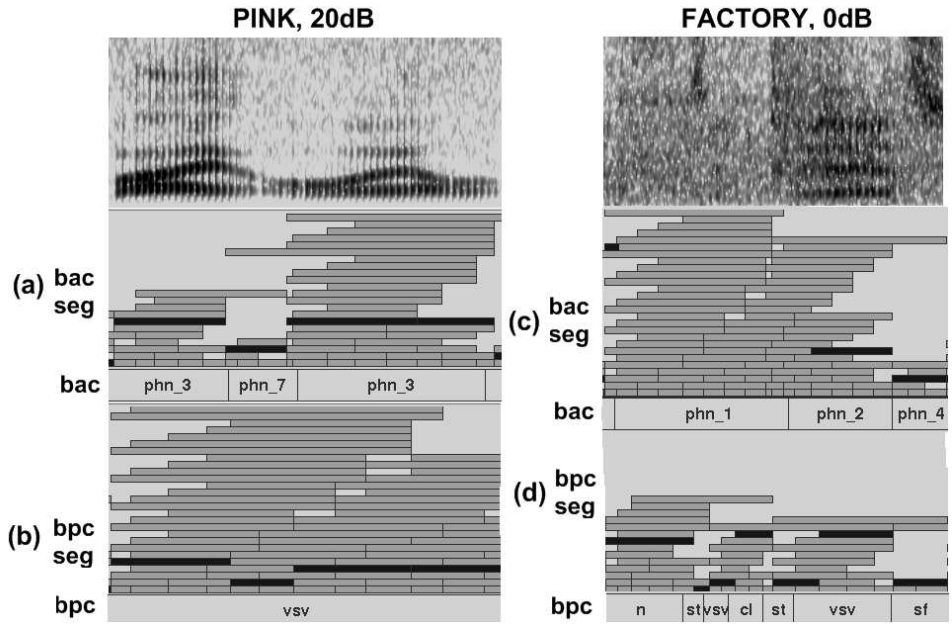

Figure 4-17: Graphical displays of BAC and BPC methods in SUMMIT. The top display contains speech spectrograms. Below that, (a) shows a segment-network for the BAC method in pink noise, and *bac* indicates the hypothesized BACs. Similarly, (b) shows the network for the BPC method in pink, and *bpc* are the hypothesized BPCs. The darker colored segments indicate the highest scoring segmentation achieved during search. (c) and (d) show the BAC and BPC methods in factory noise.

4.6.4 Broad Acoustic Segmentation with Fixed Classes

Thus far, we have observed that while there is little difference in PER for the BPC and BAC methods, the computation time is faster when there are more broad classes. In this section, we compare the BAC segmentation method when we choose just 7 clusters, to match the number of classes chosen by the BPC method. Table 4.4 shows the PER results averaged across the three noise conditions for the BAC method with 7 clusters as well as when clusters are chosen using the V-measure. Again the number of clusters for pink, babble and factory noise are shown in parentheses. The performance of the BPC segmentation method is listed as well for comparison.

At high SNRs, having fewer clusters results in front/back and semivowels merged into one class. As discussed in Section 4.6.3, merging front and back vowel classes leads to slower recognition computation time. In fact, at high SNRs, 7 clusters leads to the following classes: {vowels/semivowels, nasals, weak fricatives, strong fricatives,

TIMIT Average Phonetic Error Rates			
db	bpc	bac-variable clusters	bac-7 fixed clusters
Clean	27.7	**27.3** (11,11,11)	27.4
30dB	28.4	**28.3** (12,10,10)	**28.3**
20dB	**31.5**	31.7 (7,6,7)	31.6
10dB	41.1	**40.9** (6,5,6)	41.1
0dB	**57.9**	58.0 (5,5,4)	58.1
-5dB	**67.3**	69.6 (4,3,4)	68.0
Average	**42.3**	42.6	42.4

Table 4.4: PERs for BPC, BAC method with variable clusters and BAC method with fixed clusters on TIMIT Test Set, averaged across pink, babble and factory noises at each SNR. The bac-variable method indicates the number of clusters at each SNR for pink, babble and factory noise in parentheses. The best performing method at each SNR is indicated in bold.

stops, closures, silence} which are exactly the 7 BPCs.

At low SNRs, the extra number of BACs is mainly due to having two clusters for vowels and fricatives. In general, having 7 clusters does not change the segmentation performance much, as we have two clusters to predict certain classes rather than one. Furthermore, the locations of broad classes are used as "guides" to determine where to place major landmarks. If there is a broad class transition in the middle of the vowel where there is little spectral change, as is sometimes the case when using 7 BACs at low SNRs, these transitions are usually ignored since no major landmark setting identifies this transition. This explains why the segmentation performance in general is not sensitive to choosing an optimal number of BACs or 7 BACs.

The only exception to this is using 7 BACs at -5dB of babble noise, where we have 4 clusters to explain vowels, nasals, weak fricatives and closures instead of 1. However, the clusters are mixed between the broad classes, thus giving a low homogeneity score, which explains why the V-measure metric did not identify 7 as an optimal cluster number. Yet, because there are now 4 classes instead of 1 to explain a large portion of the audio signal, the broad class HMM decoder switches between these classes more frequently. Thus, there are more potential anchor points to detect segments, which explains why having 7 classes offers better performance.

In conclusion, matching the number of BACs to BPCs only changes the perfor-

mance of the BAC method in babble noise. While the results in Section 4.6.3 do indicate that having more classes leads to faster computation time, both Tables 4.2 and 4.4 verify the main message of this chapter - using a broad class pre-preprocessor for landmark detection and segmentation, whether acoustically or phonetically motivated, leads to significant recognition improvements in noisy environments compared to the spectral change and sinusoidal methods.

4.7 Chapter Summary

In this chapter, we explored using BPCs and BACs under different noise conditions to design a robust segment-based algorithm. We demonstrated that utilizing broad classes for both major and minor landmark placement offered improvements over the baseline spectral change and sinusoidal methods on the noisy TIMIT task. Also, we introduced a phonetic similarity metric into the V-measure, which allowed us to choose an appropriate number of distinct acoustic clusters and analyze under what noises the BAC or BPC method is preferred. We found that the BPC method has faster computation time in non-stationary noises, while BAC is faster in stationary conditions.

While utilizing broad classes as a pre-processor certainly improves the segmentation and corresponding recognition performance, notice from Figure 4-16 that the size of the search space and the corresponding search computation time grow as the SNR decreases and the environment becomes more noisy. In essence, more time is spent during search, and one might argue unnecessarily, when the signal becomes less reliable. In the next chapter, we tackle this problem by exploring broad class knowledge to identify reliable regions in our input signal. We then utilize the reliable regions to guide the search such that more time is spent in reliable regions and less time in unreliable ones.

Chapter 5

Broad Class Knowledge for Island-Driven Search

5.1 Introduction

In Chapter 3 we introduced an instantaneous adaptation technique using the EBW Transformations to recognize broad classes in noisy speech. This broad class knowledge was utilized as a pre-processor to aid in landmark detection in a segment-based speech recognition system in Chapter 4. We found that taking advantage of broad class information improved the segmentation and corresponding recognition performance in a variety of noise conditions. However, we also observed that, when the SNR decreases and the signal becomes more unreliable, more computational effort is spent during the search.

In this chapter, we address this problem by utilizing broad class knowledge as a pre-processor to first identify reliable regions in the input signal, which we refer to as islands. Portions of the signal which are not identified as islands are defined to be gaps, and represent areas of the signal which are unreliable. Reliable island regions are then used to develop a noise-robust island-driven search strategy. Specifically, we alter our search such that more effort is spent in reliable, information-bearing parts of the signal and less time in unreliable gap regions. We will demonstrate that, by utilizing regions of reliability during search, we can not only reduce the amount of

computation spent during search, but also improve recognition accuracy as well.

5.1.1 Motivation

Many speech scientists believe that human speech processing is done first by identifying regions of reliability in the speech signal and then filling in unreliable regions using a combination of contextual and stored phonological information [3], [88]. However, most current decoding paradigms in speech recognition consist of a left-to-right scoring and search component, and an optional right-to-left component, without utilizing knowledge of reliable speech regions. More specifically, speech systems often spend the bulk of their computational efforts in unreliable regions when in reality most of the information in the signal can be extracted from the reliable regions [103]. In the case of noisy speech, if phrases are unintelligible, this may even set the search astray and make it impossible to recover the correct answer [73]. This is particularly a problem in large vocabulary speech systems, where pruning is required to limit the size of the search space. Pruning algorithms generally do not make use of the reliable portions of the speech signal, and hence may prune away too many hypotheses in unreliable regions of the speech signal and keep too many hypotheses in reliable regions [56].

Island-driven search [12] is an alternative method to better deal with noisy and unintelligible speech. This strategy works by first hypothesizing islands from regions in the signal which are reliable. Further recognition works outwards from these anchor points to hypothesize unreliable regions. Island-driven search has been applied in a variety of areas, for example in parsing [17] and character recognition [73], though has been relatively unexplored in probabilistic automatic speech recognition (ASR). The goal of this chapter is to explore an island-driven search strategy for modern-day ASR.

The incorporation of island-driven search into continuous ASR poses many challenges which previous techniques have not addressed. First, the choice of island regions is a very challenging and unsolved problem [99]. Kumaran et al. have explored an island-driven search strategy for continuous ASR [56]. In [56], the authors perform a first-pass recognition to generate an N-best list of hypotheses. A word-confidence

score is assigned to each word from the 1-best hypothesis, and islands are identified as words in the 1-best hypothesis which have high confidence. Next, the words in the island regions are held constant, while words in the gap regions are re-sorted using the N-best list of hypotheses. This technique was shown to offer a 0.4% absolute improvement in word error rate on a large vocabulary conversational telephone speech task. However, we argue that, if the motivation behind island-driven search is to identify reliable regions of the signal which might be thrown away during pruning, identifying these regions from an N-best list generated from pruning is not an appropriate choice.

The use of acoustic information to identify islands is explored by Park in [71]. Specifically, Park introduces a probabilistic landmark algorithm which assigns probability scores to all possible acoustic landmarks. A Viterbi search is performed through this landmark probability network to determine the best set of landmarks. Because landmarks have probabilities assigned to them, an N-best sequence of landmarks can also be hypothesized. Park identifies islands as landmarks which do not change from one N-best hypothesis to the next. While this algorithm does introduce a technique to identify island and gap regions from the input speech signal, subsequent use of this information in continuous speech recognition was not explored.

The use of island information for parsing has also been explored in the BBN HWIM speech understanding system [98]. In this system, parsing works outwards from island regions to parse a set of gap regions. While this type of approach has shown promise for small grammars, it has not been explored in large vocabulary speech recognition systems due to the computational complexities of the island parser.

Finally, [73] explores island-driven search for isolated-word handwriting recognition. The authors identify reliable islands in isolated words to obtain a small filtered vocabulary, after which a second-pass more detailed recognition is performed. This technique is similar to that explored by Tang et al. in [89] for improving lexical access using broad classes. However, these solutions cannot be directly applied to continuous speech recognition. Thus, our first goal is to develop a methodology to identify reliable island regions that can be applied to continuous ASR.

Second, the nature of speech recognition poses some constraints on the type of

island-driven search strategy preferred. While island searches have been explored both unidirectionally and bi-directionally, the computational and on-line benefits of unidirectional search in speech recognition make this approach more attractive. Furthermore, if reliable regions are identified as sub-word units and not words, a bidirectional search requires a very complex vocabulary and language model. Unidirectional island-driven search strategies, which have been explored in [12] and [21], typically make use of a heuristic strategy to significantly decrease the number of nodes expanded. Therefore, our second goal is to explore the use of island/gap regions in a unidirectional framework to decrease the number of nodes expanded during search. Specifically we look to use island/gap knowledge to efficiently prune the search space and decrease the amount of computational effort spent in unreliable regions. We hope that, by increasing the efforts spent in reliable islands, the recognition accuracy will also improve.

5.1.2 Proposed Approach

In this chapter, we look to develop a method of island-driven search which can be incorporated into a modern probabilistic ASR framework. Specifically, we look to alter the typical left-to-right search such that more computational effort is given to reliable island regions compared to gap areas. A block diagram of the proposed system is illustrated in Figure 5-1.

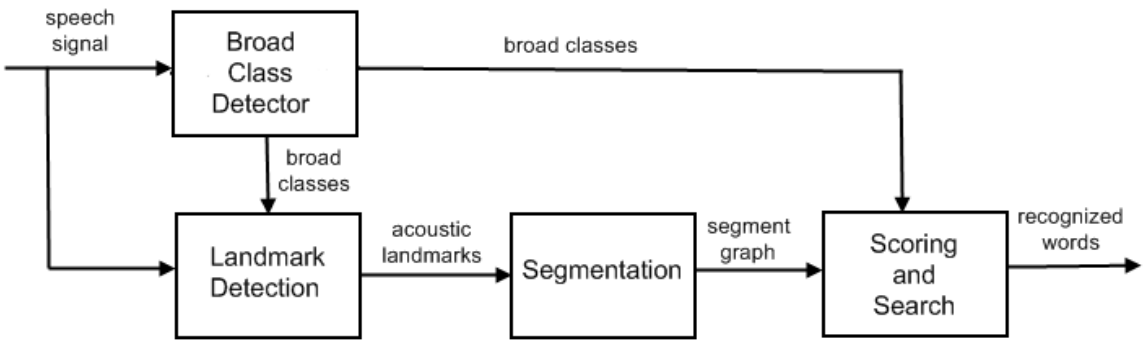

Figure 5-1: Block Diagram of Broad Class Pre-Processor within SUMMIT Framework Utilized for Island-Driven Search

First, we explore utilizing broad class knowledge to identify reliable island regions[1]. The EBW-HMM broad class recognizer, described in detail in Chapter 3, is used to detect broad classes. A confidence score is then assigned to each hypothesized broad class, and island regions are identified as those broad classes with high confidence. We then utilize the island/gap knowledge to better guide our search and limit our search methods from unnecessarily spending too much computation time in unreliable regions.

In Chapter 4 we observed that while the broad class segmentation method offered improvements in noisy conditions, the size of the search space and corresponding recognition computation time increased as the signal became more unreliable. Therefore, we explore utilizing island information to prune the segmentation graph generated from the broad class segmentation. In Chapter 4 we also observed that there was very little difference in performance if the broad class segmentation was motivated along acoustic *vs.* phonetic dimensions. Therefore, in this chapter, we focus on applying island-pruning to the broad phonetic class (BPC) segmentation technique.

In addition, we investigate utilizing island information in the scoring and search component of the recognition process. Specifically, to limit spending time unnecessarily in gap regions, we look at scoring less detailed models in gap regions in the form of broad classes and more detailed acoustic models in island regions.

First, we explore the proposed island-driven search strategy on the small vocabulary Aurora-2 noisy digits task. We will demonstrate that our island-based segment pruning method offers improvements in both performance and computation time over the broad class segmentation method. In addition, further usage of island information during final recognition offers additional improvements in both performance and computation time.

We then investigate the extension of these island-driven methods to the large vocabulary CSAIL-info corpus. We will illustrate that, on the CSAIL-info task, island-driven techniques offer comparable performance to the broad class segmentation method, though still provide faster computation time.

[1]Note that in this chapter, the term broad class will refer to just broad phonetic classes (BPCs).

5.1.3 Overview

The rest of this chapter is broken down as follows. We discuss our method for detecting islands in Section 5.2. Utilization of island information for pruning of the search space and during final recognition are discussed in Sections 5.3 and 5.4 respectively. Section 5.5 discusses the experiments performed, while Sections 5.6 and 5.7 discuss the results on Aurora-2 and CSAIL-info tasks respectively. Finally, Section 5.8 summarizes the main findings in this chapter.

5.2 Identifying Islands

In this section, our method of identifying island regions is discussed. Island identification is motivated from acoustic information itself rather than using language model information, as done in [56], so as to ensure that islands are not detected from a pruned search space. Specifically we look to detect reliable islands from broad class knowledge for three reasons. First, in Chapters 3 and 4, we have illustrated that broad classes are much more spectrally distinct and more robustly detected in noisy environments compared to the underlying phonemes. Second, as [88] discusses, when humans process speech, they utilize articulator-free broad classes (i.e., vowels, nasals, fricatives, etc) as one source of information when identifying reliable regions in the signal to help in further processing of unreliable regions. Third, island-driven search experiments are conducted on the broad class segment-based recognizer discussed in Chapter 4. This method uses broad class knowledge in designing a robust landmark detection and segmentation algorithm. As we will show, utilization of this same broad class knowledge to further reduce the search space during island-driven search allows us to make direct use of the broad class segmentation and does not require further reliability detectors (i.e., syllables, etc.), therefore minimizing system complexity.

Thus, to detect reliable island regions from broad class knowledge, we define reliable areas to be those broad classes which are detected with high confidence. To determine confidence scores for hypothesized broad classes, we explore a broad class-level acoustic confidence scoring technique, as discussed in [50]. The confidence

scoring method is discussed in more detail below.

5.2.1 Confidence Features

First, we derive a series of features for each hypothesized broad class based on frame-level acoustic scores generated from the HMM broad class recognizer described in Chapter 3. The most common acoustic score for broad class confidence is the maximum *a posteriori* (MAP) probability, as given by Equation 5.1. Here c_i is the hypothesized broad class and o_t is the observed feature vector at a specific frame t. The value of $p(c_i|o_t)$ varies from 0 to 1, and is closer to 1 the higher the confidence in the hypothesized model c_i.

$$C_{map}(c_i|o_t) = p(c_i|o_t) = \frac{p(o_t|c_i)p(c_i)}{p(o_t)} \tag{5.1}$$

The other acoustic score proposed in [50] is the normalized log-likelihood (NLL) score $p(c_i|o_t)$, as given by Equation 5.2. This confidence measure is based purely on the acoustic score $p(o_t|c_i)$ and does not incorporate the prior class probability $p(c_i)$ like the MAP score. The NLL confidence score ranges from $-\infty$ to $\log p(c_i)$, with increased confidence in class c_i reflected by a more positive NLL value.

$$C_{nll}(c_i|o_t) = p(c_i|o_t) = \log\left(\frac{p(o_t|c_i)}{p(o_t)}\right) \tag{5.2}$$

Using these frame-level acoustic confidence features, we can derive broad class-level features for each hypothesized broad class by taking various averages across the frame-level features[2]. Table 5.1 shows the features used. A more detailed mathematical description of these features can be found in [50].

A complete illustration of the steps taken to obtain broad class confidence features is displayed in Figure 5-2. The first panel shows the broad class recognition output from the HMM. In the second panel, frame-level acoustic confidence features are extracted at each frame o_t. Then in the third panel, broad class-level features, f_1 and

[2]Note that if the same broad class is hypothesized twice in a row, a separate broad class-level feature is extracted for each broad class.

Feature
Arithmetic Mean of C_{map} scores
Arithmetic Mean of C_{nll} scores
Geometric Mean of C_{map} scores
Geometric Mean of C_{nll} scores
Standard Deviation of C_{map} scores
Standard Deviation of C_{nll} scores
Catch All Model scores

Table 5.1: Broad Class-Level Features

f_2, are computed from the frame-level confidence features.

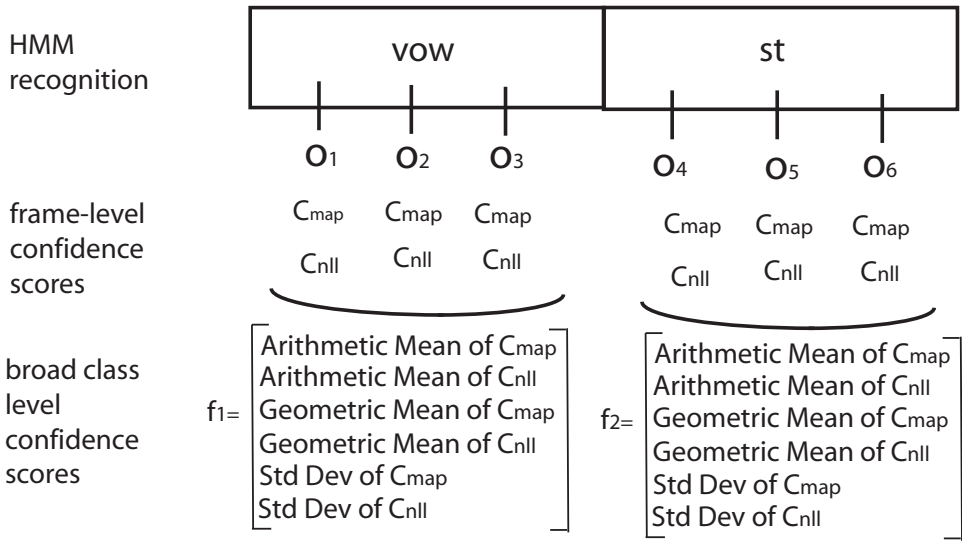

Figure 5-2: Diagram of various steps in obtaining broad class confidence features. The first panel shows the broad class recognition output from the HMM. In the second panel, frame-level acoustic confidence features are extracted at each frame o_t. Finally, in the third panel, broad class-level features, f_1 and f_2, are computed from the frame-level features.

5.2.2 Confidence Scores from Features

After broad class-level features are extracted from each hypothesized broad class, a Fisher Linear Discriminant Analysis (FLDA) projection [22] is applied to reduce the set of broad class-level features f into a single dimension confidence score. The goal of the FLDA is to learn a projection vector w to reduce dimensionality of f while achieving maximal separation between two classes. Typically, these two classes are

correctly and incorrectly hypothesized sub-word units (i.e., [50]). However, the goal of our work is to identify reliable island regions, not correctly hypothesized broad classes. More intuitively, a silence or stop closure could be hypothesized correctly but generally provides little reliability information on the actual word spoken relative to a voiced sound, such as a vowel. Therefore, a 2-class unsupervised k-means [22] clustering algorithm is applied to the feature vectors f to learn a set of set of two classes, denoted as $class_0$ and $class_1$.

To analyze the behavior of these two learned classes, Figure 5-3 shows a histogram of the arithmetic mean of the C_{map} scores, one of the acoustic features extracted from the broad class recognition results, for $class_0$ and $class_1$. The figure indicates that there is good separation between the C_{map} scores for the two different classes. In addition, the C_{map} scores are much higher for $class_0$ relative to $class_1$, showing that there is higher confidence in this class. In addition, Figure 5-4 shows a histogram of the standard deviation of the C_{map} scores for $class_0$ and $class_1$. Not only is there good separation between the two classes, but the standard deviation of scores for $class_0$ is also smaller than the scores for $class_1$, another indication of higher confidence in $class_0$. Similar trends were observed for the other broad class-level features listed in Table 5.1. The trends illustrated in Figures 5-3 and 5-4 indicate that there is higher confidence in $class_0$, and thus we will refer to this class as the "reliable" class, while $class_0$ will be called the "unreliable" class.

The trends in $class_0$ and $class_1$ are further confirmed by analyzing at the concentration of broad classes belonging to $class_0$ and $class_1$, as illustrated in Figure 5-5. The figure shows that most of the reliable broad-classes, i.e., nasals, vowels and semi-vowels, belong to $class_0$. However, unreliable classes such as closures, strong fricatives, silence, stops, and weak-fricatives, have a higher concentration in $class_1$.

After a set of two classes is learned, the FLDA is then used to learn a linear projection w. The projection vector is then applied to a newly hypothesized broad class feature vector to produce a single acoustic confidence score, namely $F_{score} = w^T f$.

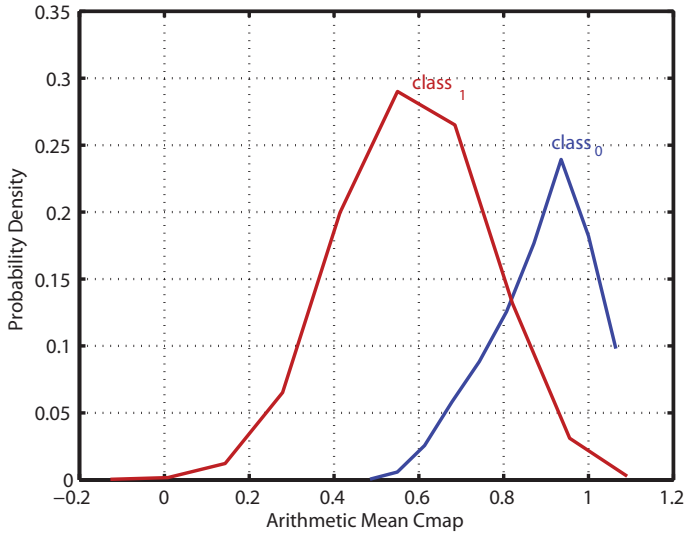

Figure 5-3: Histogram of the arithmetic mean of the C_{map} scores for $class_0$ and $class_1$

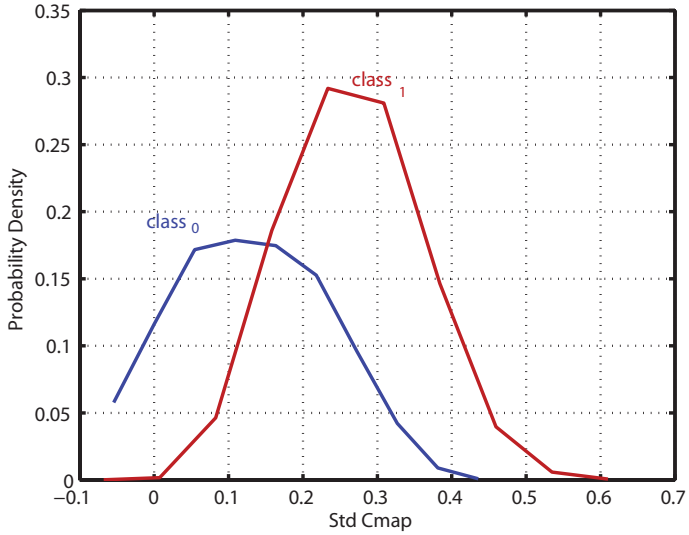

Figure 5-4: Histogram of the standard deviation of the C_{map} scores for $class_0$ and $class_1$

5.2.3 Detecting Island Regions

After confidence scores are defined for each hypothesized broad class, an appropriate confidence threshold to accept the broad class as a reliable island region must be determined. Ideally, we would like island regions to include reliable broad classes, that is vowels, semivowels and nasals. Furthermore, we would like transitions between is-

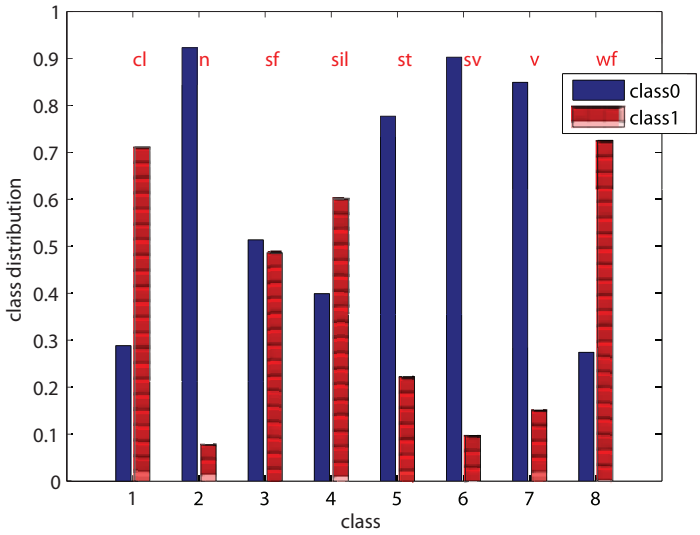

Figure 5-5: Distribution of broad classes belonging to $class_0$ and $class_1$

lands and gaps to occur at true boundaries between reliable/unreliable broad classes in the utterance, but would like to minimize the transitions that occur in the middle of sequences of reliable or unreliable broad classes. Figure 5-6 shows an example of a set of hypothesized broad classes, along with two island/gap hypotheses. Notice that the first island/gap hypothesis detects transitions between reliable/unreliable broad classes. However, the second island/gap hypothesis is poor as an island/gap transition, delineated by an 'X', is hypothesized in the middle of an unreliable sequence of broad classes.

Figure 5-6: A hypothesized set of broad classes, along with two examples illustrating a set of good and poor island/gap detections. The second island/gap hypothesis is poor as an island/gap transition, delineated by an 'X', is hypothesized in the middle of an unreliable sequence of broad classes

Thus, we define our goal of detecting reliable broad classes as those broad classes that provide a high probability of detecting the true reliable/unreliable transitions with a low false alarm probability. The probability of detection is calculated by looking at the percentage of true reliable/unreliable transitions that are detected by a particular island/gap transition. Similarly, the probability of false alarm is calculated by the percentage of island/gap transitions that do not detect a true reliable/unreliable transition.

To find an appropriate confidence threshold setting, we calculate a Receiver Operating Characteristic (ROC) [22] curve, which is a common tool used to find a suitable tradeoff between detection and false alarms as the confidence threshold setting is varied. Figure 5-7 shows a graphical display of this ROC curve for different confidence threshold settings. The optimal confidence threshold, as indicated by the rectangular box, offers a high probability of detection and low false alarm rate.

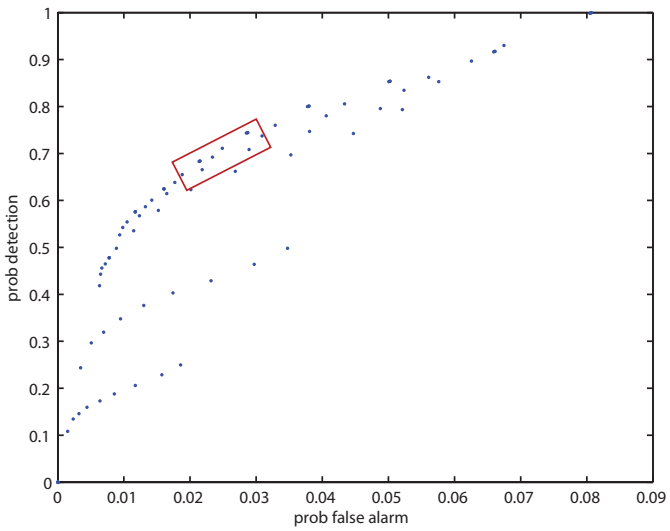

Figure 5-7: ROC Curve for different confidence threshold settings. The optimal confidence threshold is indicated by a rectangular box.

After an appropriate setting is determined to define island regions, we then use this information in our island-driven search methods. In Section 5.3 we discuss a method to prune the search space while in Section 5.4 we explore a technique to reduce computation time during model scoring.

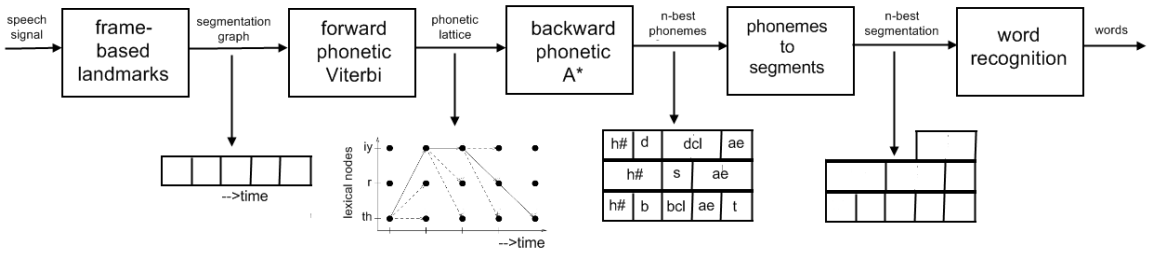

Figure 5-8: Block Diagram of Segmentation by Recognition

5.3 Island-Driven Segmentation

Segment-based recognizers can often be computationally expensive, as the number of possible segmentations can grow exponentially with the number of segments [13]. We have observed in Chapter 4 that the size of the search space and number of segmentations also grows as speech is subjected to noisier environments. Therefore, we first explore utilizing island/gap information to prune the size of the segment graph.

5.3.1 Segmentation by Recognition

Segmentation by recognition has been explored in [13] and [61] as a means of producing a smaller segment graph with more meaningful segments. A block diagram of segmentation by recognition in shown in Figure 5-8.

In this method, landmarks are first placed at a fixed frame rate independent of acoustic change. Then, a forward phonetic Viterbi search is performed to produce an phonetic lattice with corresponding acoustic scores. Next, a backwards A^* search [44] is carried out on this lattice to produce an N-best list of phonemes. This N-best list is then converted into a new pruned N-best segment graph. A second-pass word recognition is then performed over this pruned segment graph.

Segmentation by recognition is attractive for two reasons. First, the pruned segment graph is produced from phonetic recognition and therefore the segments are much better aligned to the phonemes we want to match during word recognition. Second, the segment graph is much smaller, allowing more paths to be kept alive

during recognition and subsequently reducing the chances of throwing away potentially good paths. In fact, improvements with segmentation by recognition over the acoustic segmentation method were found for both phonetic [13] and word recognition tasks [61]. We take advantage of this segmentation by recognition idea in utilizing island/gap knowledge to prune the search space.

5.3.2 Island-Based Segmentation

Many dynamic beam width approaches (i.e., [2], [23]) have resulted in greater recognition speed-up but not an improvement in recognition performance. One explanation for this is that the dynamic pruning strategies prune the gap and island regions independently without incorporating acoustic confidence information. Therefore, to try and better utilize the reliable regions for dynamic pruning, we explore using island/gap contexts to prune the segment graph, an idea that is similar to the segmentation by recognition idea discussed in the previous section.

More specifically, we first use the broad classes to define a set of island/gap regions as presented in Section 5.2, and to define a set of variable frame rate acoustic landmarks discussed in Chapter 4. We look at using the variable frame rate landmarks rather than fixed frame rate landmarks due to the computational benefits of having fewer landmarks, an idea which was first explored in [61].

Island/gap knowledge is then used to chunk an utterance into smaller sections at islands of reliability. This not only allows us to vary the amount segment pruning in island *vs.* gap regions, but also allows the future potential opportunity to parallelize the forward/backward search done in each region, similar to [61], therefore not requiring exactly two full recognition passes.

In each island region, a forward *phonetic* Viterbi search is done to produce an phonetic lattice. A backwards A^* search over this lattice then generates a smaller list of N-best segments, after which a new pruned segment graph is created in the island regions. Here N, the number of allowed paths, is chosen to be the N which optimizes the recognition performance on a held out development set.

Next, the pruned segment graphs in the island regions are used to influence seg-

ment pruning in the gap regions. More specifically, another forward Viterbi/backward A^* is performed across each gap-island-gap region. Here the pruned island segment graph from the island pruning is inserted in the island regions. Again, N is chosen to optimize performance on the development set. We chose N in the gap regions to be smaller than in the island regions to allow for fewer segments in areas we are less confident about and more detailed segments in confident island regions.

Finally, the N-best segments from the island and gap regions are combined to form a pruned segment graph [3]. Then, given the new segmentation by recognition graph, a second-pass full *word* recognition is done over this pruned search space. We will refer to this segment-pruning technique described above as an island-driven segmentation, as fewer segments are permitted in areas of reliability and denser segmentation is allowed during regions of less confidence.

A pictorial view of the island-based segmentation idea from the SUMMIT recognizer is illustrated more clearly in Figure 5-9. Item A in the figure shows a spectrogram and corresponding segment graph in SUMMIT. Item B indicates the detected island and gap regions. C illustrates a forward Viterbi and backward A^* search in the island regions, while D shows a forward Viterbi and backward A^* search over a gap-island-gap region. Finally E depicts a pruned segment graph.

5.4 Utilization of Island Information During Final Recognition

In the previous section, we demonstrated that by utilizing island/gap information, the size of the segmentation graph could be reduced. In this section, we explore the utilization of island/gap regions during the final recognition search component to further differentiate between the search effort in islands *vs.* gaps.

[3]Note that the island-gap transitions are hard boundaries, so segments in island and gap regions are not connected together.

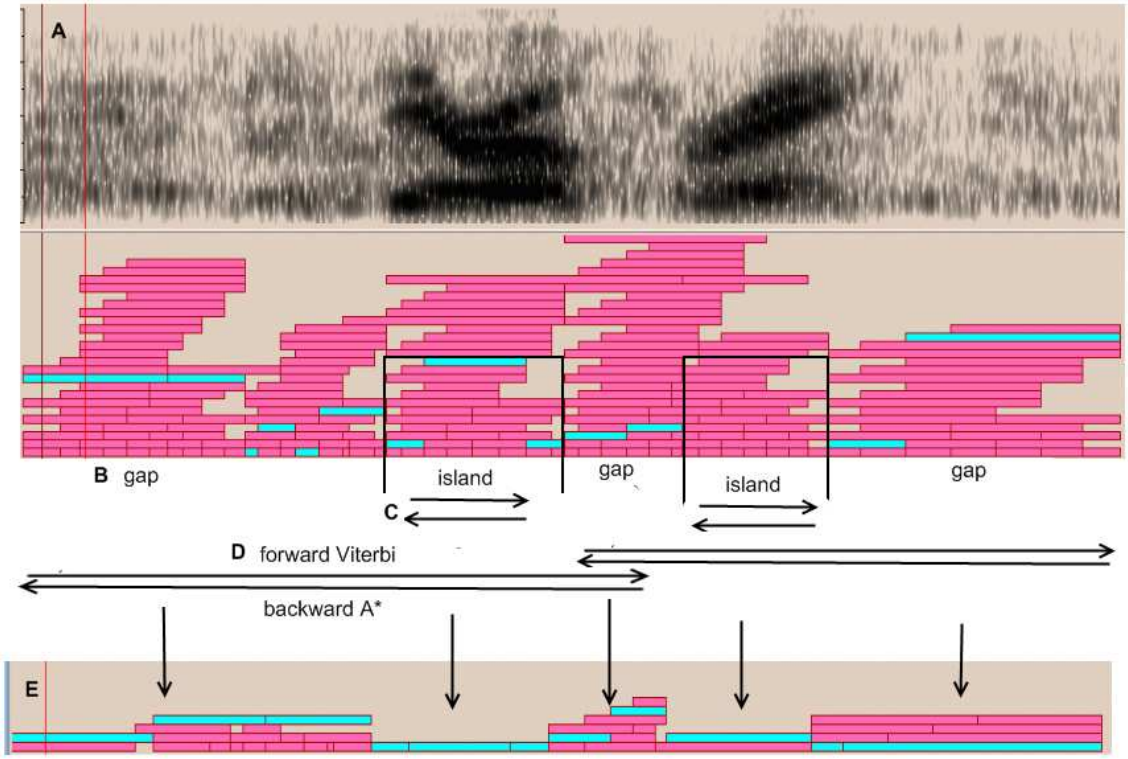

Figure 5-9: A view of island-based segmentation from the SUMMIT recognizer.*A* shows a spectrogram and corresponding segment graph in SUMMIT. *B* illustrates island and gap regions. *C* shows a forward Viterbi and backward A^* search in the island regions, while *D* illustrates a forward Viterbi and backward A^* search over a gap-island-gap region. Finally *E* depicts the resulting pruned segment graph.

5.4.1 Overview

Many speech systems often spend the bulk of their computational efforts in unreliable regions when in reality most of the information in the signal can be extracted from the reliable regions [103]. This trend is illustrated in Figure 5-10, which shows the average number of active viterbi nodes within each phoneme for all 11 words in Aurora-2 digit task. The number of active viterbi nodes is also a measure of beam width. Notice that the number of active counts is much higher at the beginning of a word, where the phonemes are unvoiced and unreliable. However, after knowledge of reliable phonemes, such as vowels and semi-vowels, the number of active counts drops. Similar behavior can also be observed in Figure 5-11, which depicts the percentage of acoustic models requested by the search for each phoneme in a word. Again, notice

that the number of models evaluated is higher for unreliable phonemes compared to reliable phonemes. Both figures confirm the fact that the search component of a speech recognition system spends most of its effort in unreliable regions.

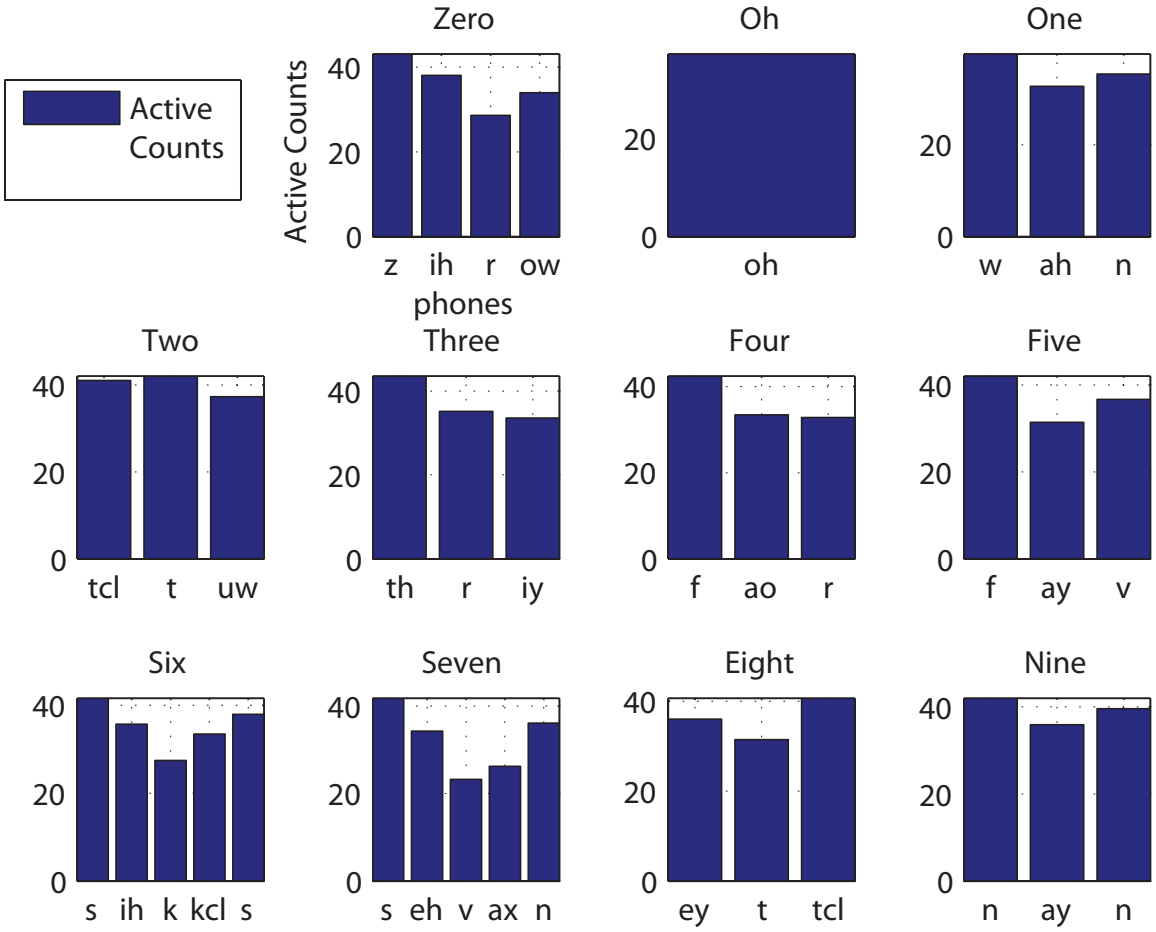

Figure 5-10: Average Number of Active Viterbi Nodes within each phoneme of a word. Plots are shown for all 11 digits in the Aurora-2 task.

Thus, to reduce the computation in unreliable regions, we explore a technique to score less detailed acoustic models in gap regions and more detailed models in island regions. For example, the Aurora-2 corpus contains 28 phones, and therefore effectively scores 157 diphone acoustic models (after clustering) for each possible segment. If less detailed broad class models are scored for each segment, this can reduce the number of acoustic models to approximately 49, roughly one-third. Therefore, we investigate scoring broad class acoustic models in the gap regions and more detailed full phonetic acoustic models in islands. Figure 5-12 gives a graphical illustration of

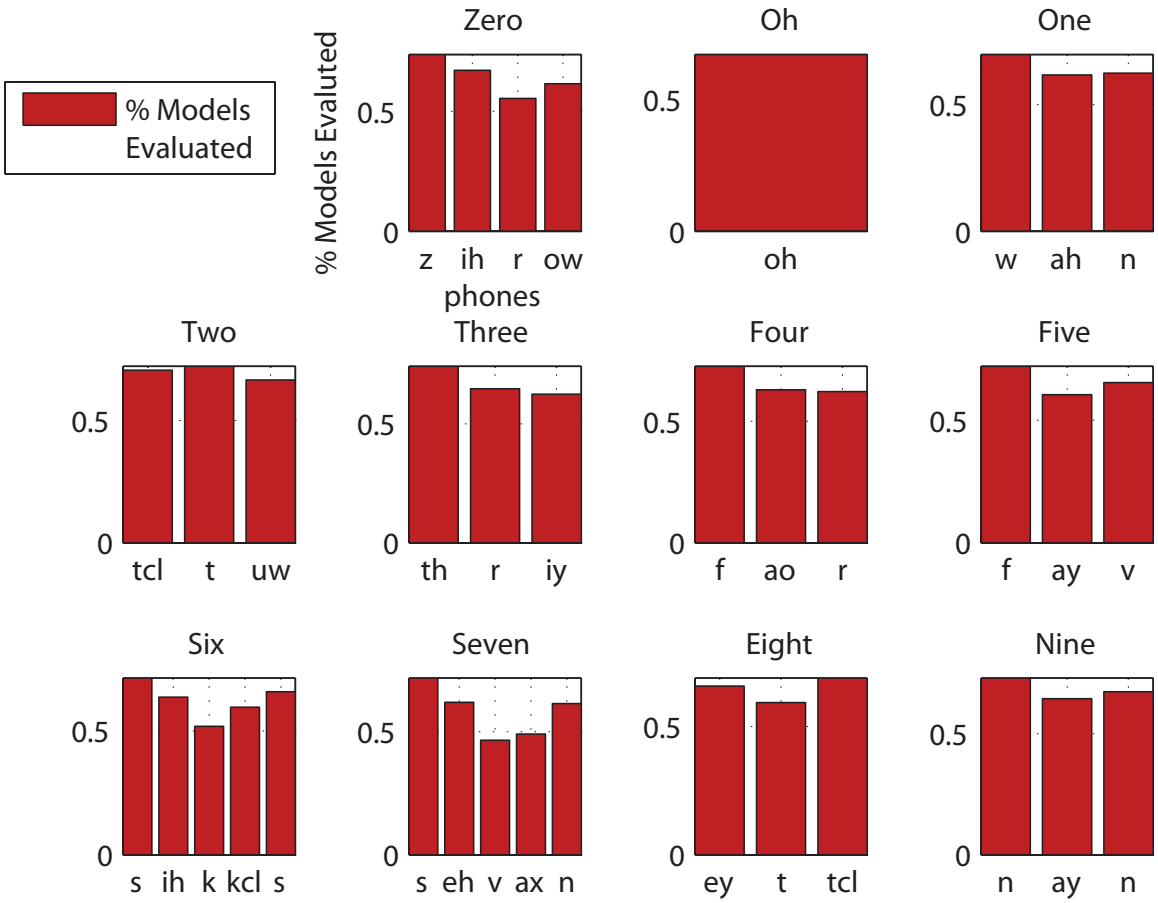

Figure 5-11: Average Number of Models Evaluated within each phoneme of a word. Plots are shown for all 11 digits in the Aurora-2 task.

this process. In order to implement this joint broad class/phonetic recognizer, we make changes to both the Finite State Transducer (FST) search space and acoustic models, both of which are discussed below.

5.4.2 Finite State Transducer Formulation

The SUMMIT recognizer utilizes an FST framework [33] to represent the search space. The benefit of using the FST network is that a wide variety of search networks can be constructed by utilizing basic mathematical FST operations, such as composition. In order to allow for broad class models in the search space, we represent the FST network R as being composed of the following components:

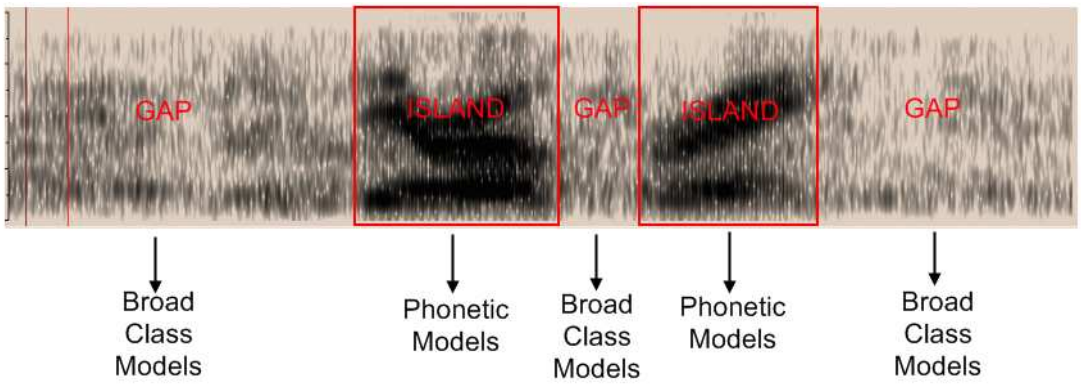

Figure 5-12: Graphical illustration of joint phonetic-broad class model scoring. First, island and gap regions are identified in the waveform. Secondly, broad class models are scored in gap regions and phonetic models are scored in island regions.

$$R = C \circ B \circ P \circ L \circ G$$

C typically represents the mapping from context-dependent (CD) phonetic labels to context-independent (CI) phonetic labels. Our CD labels include both phonetic and broad class labels, so C now represents the mapping from CD joint broad class/phonetic labels to CI broad class/phonetic labels. We next compose C with B, which represents a mapping from joint broad class/phonetic labels to CI phonetic labels. So, for example, given a broad class label for the stop class as *st* and corresponding phonetic labels for phones which make up the stop class, namely [t] and [k], the B FST representation for stop labels is given by Figure 5-13, where the mapping is given by <input label>:<output label>. Intuitively, B takes all broad class/phonetic CI labels and maps them into CI phonetic labels.

The rest of the composition is standard, with P representing the phonological rules, L the word lexicon and G the grammar. Thus, the full composition R maps input context-dependent broad class/phonetic labels directly to word strings.

Therefore each word in the lexicon is represented as a combination of broad class and phoneme sub-word units. For example, one sub-word representation of the word "three" could be:

$$\text{three} : /WF \;\; r \;\; iy/$$

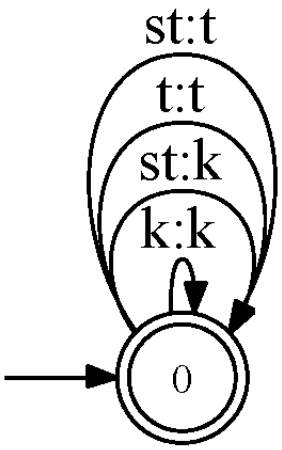

Figure 5-13: FST illustrating mapping from joint broad class/phonetic labels to context-independent phonetic labels. The mapping is given by <input label>:<output label>. Here *st* corresponds to the stop broad class label, while [t] and [k] are phonetic labels which belong to the stop class.

Intuitively, the island information */r iy/* is really characterizing the word "three" and there is no need to do a detailed search of the weak fricative */th/*.

5.4.3 Acoustic Model

The acoustic model calculates the probability of an observation o_t given sub-word unit u_n as $P(o_t|u_n)$. In island regions, the sub-word unit u_n is a phonetic model Phn and the acoustic model is scored as $P(o_t|Phn)$ for each Phn. In the gap region, the sub-word unit is a broad class model BC. We calculate $P(o_t|BC)$ by taking the average of all the phonetic model scores which make up the broad class. The expression for the broad class acoustic model score is given more explicitly by Equation 5.3. Here N is the number of Phn models which belong to a specific broad class (BC).

$$P(o_t|BC) = \frac{1}{N}\left(\sum_{Phn \in BC} P(o_t|Phn)\right) \tag{5.3}$$

This approach is chosen for two reasons. First, if a separate set of broad class and phonetic models were trained, the observation spaces used during training would be different. Therefore, the scores for $P(o_t|Phn)$ and $P(o_t|BC)$ would be in different ranges, making it difficult to combine both scores during full word recognition. While

an appropriate scale factor could potentially be learned to account for this, the second reason we choose to calculate the broad class scores via Equation 5.3 is for model simplicity. Having a separate set of broad class models would also require training up phonetic-broad class diphone models which occur at island-gap transitions, making the set of total acoustic models trained more than three times the size of the number of phonetic models.

5.5 Experiments

Island-driven search experiments are first conducted on the small vocabulary Aurora-2 database [45]. The Aurora-2 task consists of clean TI-digit utterances with artificially added noise at levels of -5db to 20db in 5db increments. We utilize this corpus for experiments because the simple nature of the corpus allows us to explore the behavior of the proposed island-driven search techniques in noisy conditions. Results are reported on Test Set A, which contains noise types similar to those in the training data, namely subway, babble, car, and exhibition hall noise. For the broad class recognizer, a set of context-independent broad class acoustic models, discussed in Chapter 3, are trained for each SNR and noise condition in the Aurora-2 training set. The same broad phonetic classes described in Section 4.2 are used, though the vowel and semi-vowel broad classes are now separated, in part due to the simple nature of the vocabulary. A unigram language model is used for each broad class. For subsequent word recognition experiments, global multi-style diphone acoustic models are used. Acoustic models are trained specific to each segmentation described, namely the spectral change, BPC segmentation and island-driven segmentation techniques. The language model gives equal weight to all digits, and allows digits to by hypothesized in any order.

Experiments are then conducted on the CSAIL-info corpus, which contains information about people, rooms, and events in the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT. The large vocabulary nature of the task, coupled with the various non-stationary noises which contaminate the speech utter-

ances, motivate us to further explore island techniques on this task. Results are reported on the development set. For the broad class recognizer, a global broad class acoustic model is trained using the CSAIL-info training set for the 7 broad phonetic classes defined in Section 4.2, and again a unigram language model is used. For word recognition experiments, diphone acoustic models are trained using data collected from the telephone-based Jupiter weather system at MIT [33]. The acoustic models are trained using only the spectral change segmentation method. A trigram language model is used.

A variety of experiments are conducted to analyze the behavior of the island-driven strategy proposed. First, we explore the robustness of the technique discussed in Section 5.2 to identify island and gap regions. Second, we analyze the word error rate (WER) of the island-based segment pruning and joint broad class/phonetic model scoring techniques. Third, the computational benefits of the island-driven techniques are investigated. This analysis is done on both the Aurora-2 and CSAIL-info tasks.

5.6 Results on Aurora

5.6.1 Island Quality Investigation

First, we investigate the robustness of the technique to hypothesize islands and gaps proposed in Section 5.2. This is achieved by analyzing the concentration of island and gap regions within each phonemic unit of each digit word. Ideally, a robust island will have a high concentration of vowels, semi-vowels and nasals, which correspond to more reliable, robust parts of the speech signal.

Figure 5-14 illustrates for each phoneme in a word, the distribution of islands and gaps within that phoneme. This distribution is shown for all 11 digits in the Aurora corpus. The distribution is normalized across each phoneme, so, for example a distribution of 0.3 for the island region for /z/ in "zero" means that 30% of the time /z/ is present in an island region and 70% of the time it is contained in a gap region. Each plot shows the same behavior, i.e., most of the vowels, semi-vowels and nasals

in each word, containing the information-bearing parts of the signal, are concentrated in the island regions. However, most of the non-harmonic classes belong to the gap regions. Now that we have illustrated the robustness of our island detection method, in the next section we analyze the performance of the island-driven search methods proposed in Sections 5.3 and 5.4 respectively.

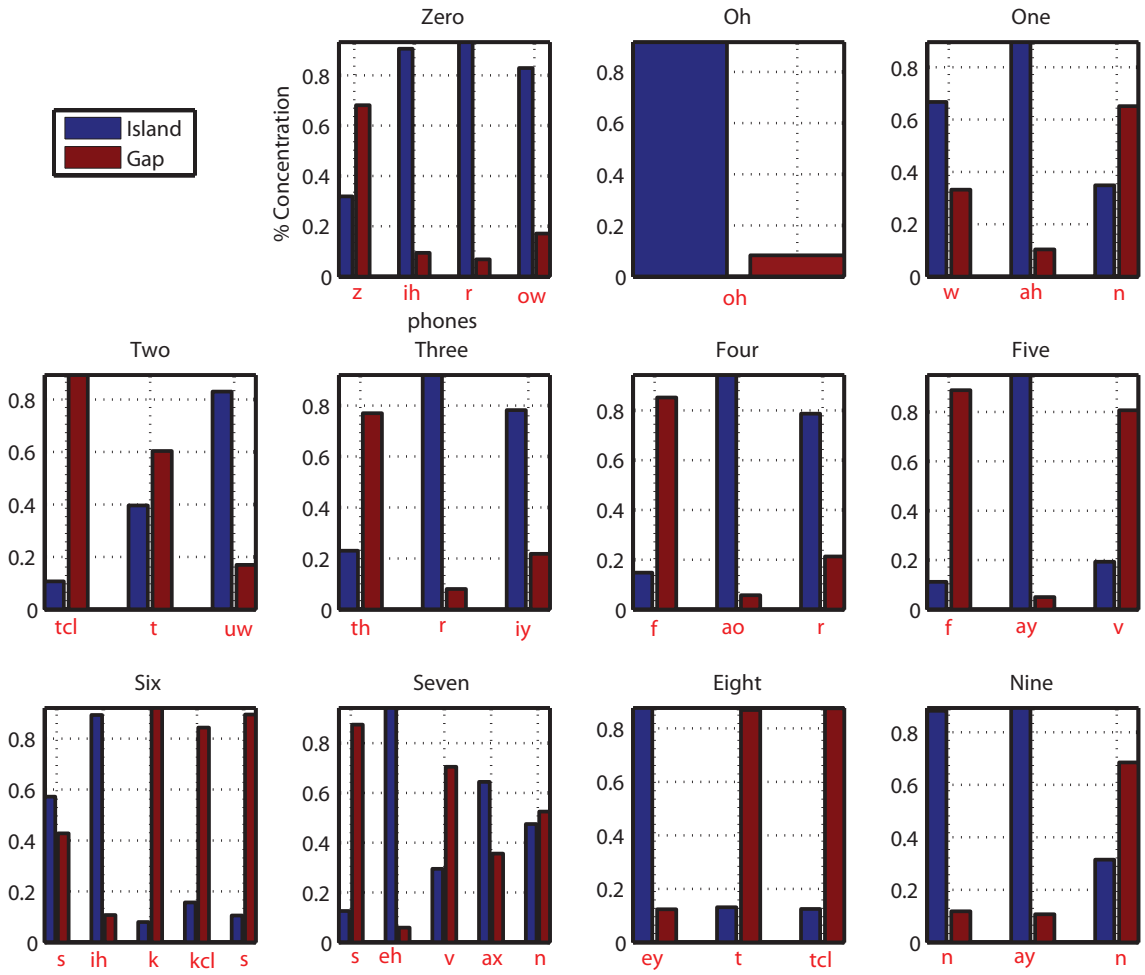

Figure 5-14: Concentration of Islands and Gaps within each phoneme of a word. Plots are shown for all 11 digits in the Aurora-2 task.

5.6.2 Error Rates

Island-Based Segment Pruning

First, we explore the island-based segment pruning on the Aurora-2 Test set. Table 5.2 compares the baseline spectral change segmentation, the BPC segmentation discussed in Chapter 4 and the island-based segmentation method. The results are averaged across all SNRs and noise types in Test Set A.

Notice that the BPC segmentation method outperforms the spectral change method. This illustrates that using a broad class pre-processor for landmark detection also offers improvements for word recognition tasks[4]. In addition, Table 5.2 indicates that the island segmentation method outperforms both the spectral change and BPC segmentation techniques, and a Matched Pairs Sentence Segment Word Error (MPSSWE) significance test [29] indicates that the island segmentation result is statistically significant from the other two approaches.

These results verify that recognition results can be improved by using the island/gap regions to reduce the segmentation graph and keep the most promising segments, thereby reducing the number of paths searched. This increases the chances that reliable parts of the signal are not thrown away, and prevents the search from scoring and keeping poor segments. A more detailed error analysis exploring the benefits of the island-driven approach is presented later in this section.

Segmentation Method	WER
Baseline Spectral Change Segmentation	31.9
BPC Segmentation Baseline	22.8
Island-Based Segmentation	**22.3**

Table 5.2: WER for Segmentation Methods on Aurora-2 Test Set A. The best performing method is indicated in bold.

[4]Note that the success of the broad class segmentation method discussed in Chapter 4 was only demonstrated on the TIMIT phonetic recognition task.

Utilization of Island Information During Final Recognition

In this section, we explore the utilization of island/gap regions during the final search to further decrease the number of nodes expanded. The first question explored is how many broad class models are necessary to score in the gap regions. Figure 5-15 shows the change in WER on the development set for the joint broad class/phonetic method as the number of broad class models is varied. Here, the additional broad class chosen at each point on the graph is picked to give the maximum decrease in WER. We also compare the WER of the joint method to scoring only phonetic models in both island and gap regions, as indicated by the flat line in the figure. *Point A* in the figure corresponds to the location where the WER of the joint broad class/phonetic approach equals that of the phonetic approach. This corresponds to 8 broad classes, which are indicated in Table 5.3.

silence	vowel
semi-vowel	nasal
closure	stop
weak fricative	strong fricative

Table 5.3: Broad Classes in Gap Region Corresponding to Point A in Figure 5-15

If the number of broad classes is increased, and particularly if additional splits are made in the strong and weak fricative classes, the WER continues to decrease. The best set of broad class models is depicted by *Point B* in Figure 5-15, with the following broad classes shown in Table 5.4. There is no extra benefit to increasing the number of broad classes past 10, as illustrated by the increase in WER.

silence	vowel
semi-vowel	nasal
closure	stop
voiced weak fricative	unvoiced weak fricative
voiced strong fricative	unvoiced strong fricative

Table 5.4: Broad Classes for Gap Region Corresponding to Point B in Figure 5-15

Using these 10 broad class models to score the gap regions, Table 5.5 compares the WER when only phonetic models are scored *vs.* using island/gap information to

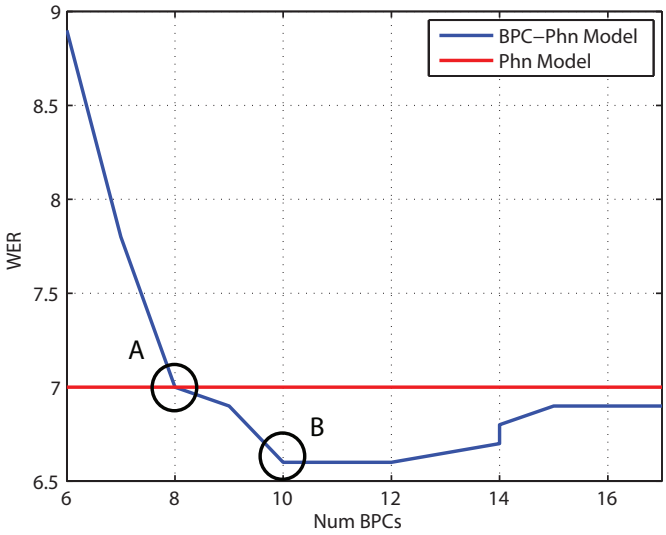

Figure 5-15: WER vs. Number of Broad Class Models when joint broad class/phonetic models are scored. The WER when only phonetic models are scored is also indicated.

score broad class/phonetic models. There is a slight improvement when broad class models are scored in gap regions, showing that doing a less detailed evaluation in unreliable regions does not lead to a degradation in performance.

Scoring Method	WER
Island-Based Segmentation, Full Phonetic Models	22.3
Island-Based Segmentation, Broad Class/Phonetic Combination	**22.1**

Table 5.5: WER for Island-Based Segmentation Methods on Aurora-2 Test Set A. The best performing method is indicated in bold.

Error Analysis

To better understand the improvement in error rate offered by the island-driven techniques, in this section we perform a detailed error analysis. Table 5.6 breaks down the WER for the BPC segmentation, island segmentation method scoring phonetic models and the island segmentation method scoring joint broad class/phonetic models, and lists the corresponding substitution, deletion and insertion rates.

Notice that the island segmentation causes an increase in substitution and deletion errors. By making cruder segment and modeling approximation, the slight increase

in the substitution and deletion rates is no surprise. However, the main advantage to the island based approach is the large decrease in insertion rate.

Scoring Method	WER	Subs	Del	Ins
BPC Segmentation	22.8	9.9	6.8	6.1
Island Seg, Phonetic Models	22.3	10.8	7.6	3.9
Island Seg, Broad Class/Phonetic Models	**22.1**	**11.1**	**8.0**	**3.0**

Table 5.6: Breakdown of Error Rates for Segmentation Methods on Aurora-2 Test Set A. The best performing method is indicated in bold.

A closer investigation of these insertion errors is illustrated in Figure 5-16a, which displays the number of insertion errors for the above three methods, when errors occur purely in island region, gap regions, or span over a combined island&gap region. In addition, Figure 5-16b illustrates the relative reduction in insertion errors over the BPC segmentation for each of these regions. The following observations can be made:

- Most of the insertions occur in gap only and island&gap regions where the signal is not as reliable compared to a pure island region.
- The biggest reduction in insertions with the island segmentation approaches occurs in the gap only region. For example, the broad class/phonetic combination has approximately a 66% reduction in insertion rate in the gap region.
- The broad class/phonetic combination has approximately a 40% reduction in insertion rate in the island and island&gap regions.

This insertion rate reduction in the gap region shows one of the strengths of island driven search. Having a detailed segmentation and phonetic model scoring in unreliable gap regions, particularly in noisy conditions, can throw the search astray without taking into account future reliable regions, resulting in a large insertion of words.

This point is illustrated more clearly in Figure 5-17 which shows the absolute reduction in insertions for the three segmentation methods for each word in the Aurora corpus. The two words that have the highest number of insertion reductions are

oh and *eight*. These two words can have very short vowels so any slight degree of voicing in gaps due to noise or pre/post voicing from words can cause these insertions. However, when we take advantage of islands of reliability to retain only the most promising segments and score less detailed models in gaps, we limit the number of unwanted insertions.

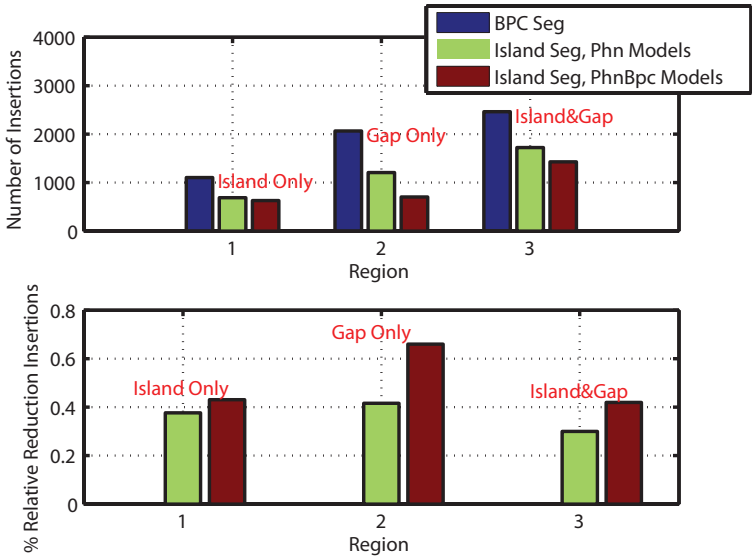

Figure 5-16: Insertion Errors

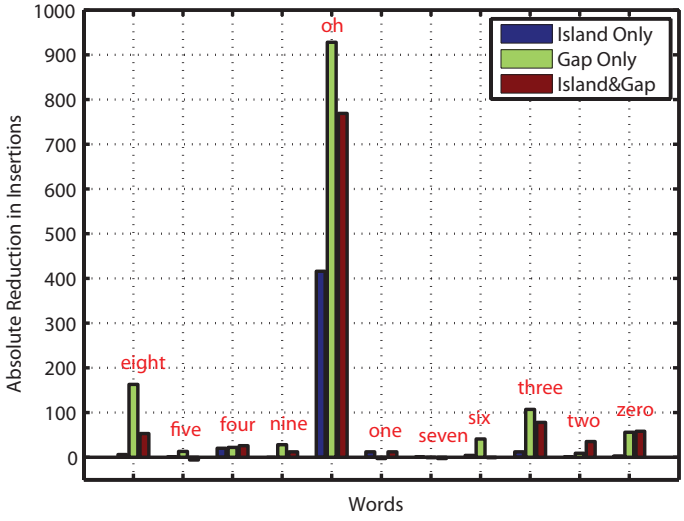

Figure 5-17: Inserted Words

5.6.3 Computational Efficiencies

The last section presented the benefits of the island-based approach for improving WER. In this section, we explore some of the computational efficiencies to the island-based approach.

First, one advantage of the island-driven segment pruning is the reduction in segment graph density. Figure 5-18 compares the average number of segments per second for the BPC segmentation and the island segmentation techniques. This is computed by calculating the number of segments produced per time (in seconds) for each utterance, and averaging this across all utterances at a specific SNR in Test Set A. Notice that the number of segments produced in the island method appears to be much less sensitive to an increase in SNR compared to the BPC segmentation approach, and it produces on average about 2.5 times fewer segments per second.

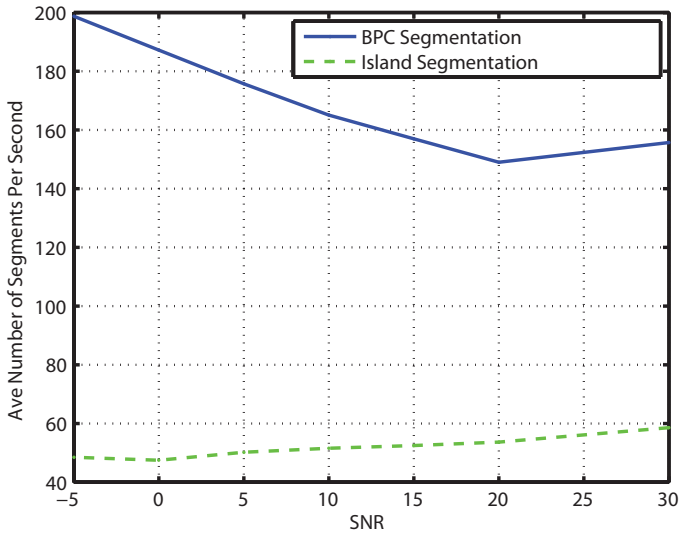

Figure 5-18: Average number of segments per second vs. SNR for BPC and Island segmentation approaches on the Aurora-2 Test Set A.

Next, we explore the Viterbi path extensions for the BPC segmentation and island segmentation approaches. The number of Viterbi path extensions is computed by counting the number of paths extended by the Viterbi search through the length of the utterance. Figure 5-19 shows a histogram of the Viterbi extensions on all utterances in Test Set A for the two approaches. Notice that the island segmentation

extends fewer paths and has an average path extension of about 9.5 (in log scale), compared to the BPC segmentation which extends roughly 10.4 paths (log scale).

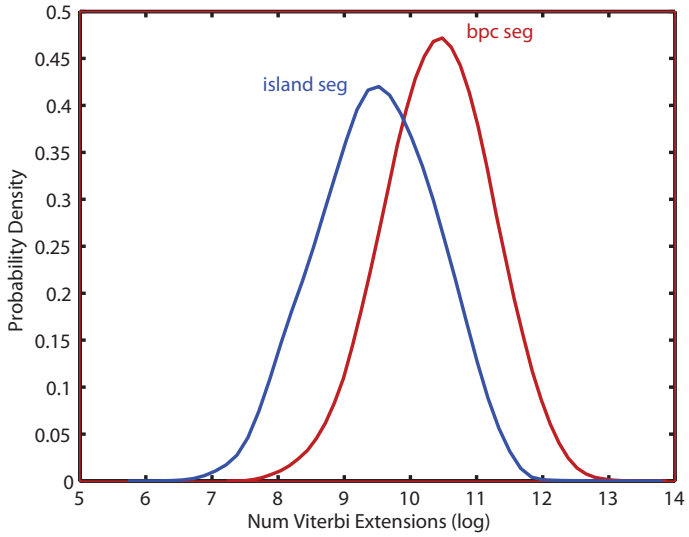

Figure 5-19: Histogram of Number of Viterbi Extensions (log scale) for the BPC and Island segmentation approaches on the Aurora-2 Test Set A.

Finally, to evaluate the benefit in computational effort with the joint broad class/phonetic recognizer, we explore the number of models requested by the search during recognition. Every time paths are extended at each landmark, the search requests a set of models to extend these paths. The number of models evaluated per utterance is computed by calculating the total number of models requested through the length of an utterance. Figure 5-20 illustrates a histogram of the number of models evaluated (in log scale) for all utterances in Test Set A, in both the island and gap regions. The joint broad class/phonetic method is much more efficient, particularly in the gap region, and evaluates fewer models compared to the phonetic method.

5.7 Results on CSAIL-info

In this section, we analyze the performance of the island-driven techniques on the CSAIL-info task.

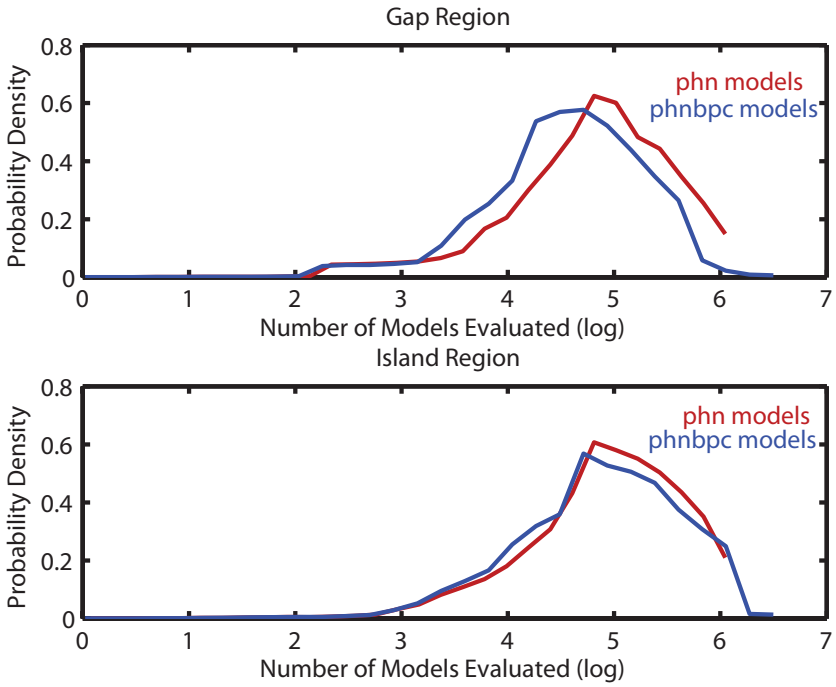

Figure 5-20: Histogram of Number of Models Evaluated in Island and Gap Regions

5.7.1 Island Quality Analysis

As in the Aurora-2 task, we first explore the quality of the island detection technique. It has been suggested that stressed syllables in English carry more acoustically discriminatory information than their unstressed counterparts and therefore provide islands of reliability [58].

To analyze the behavior of stressed syllables, the vocabulary in the CSAIL-info training and development sets were labeled with stress markings. These stress markings were obtained by looking at the IPA stress marking in the Merriam-Webster dictionary [1]. Both primary and secondary stressed syllables were marked.

Two different techniques to define islands are investigated. In [96], it was determined that only identifying stressed syllables from nucleus vowels offered more reliability than also using stress information for non-vowel segments. Thus, we first explore using the broad class island-detection technique discussed in Section 5.2 such that islands are identified to maximize the detection of true stressed vowels.

Second, instead of running a broad class pre-processor to detect islands, we in-

vestigate using a stressed/unstressed syllable detector to detect stressed syllables. In addition to including stressed vowels as part of the stressed syllable, we also include pre-syllable consonant clusters, which could potentially carry reliable information and therefore could occur in island regions. Because a pre-syllable cluster could include more than one consonant before the stressed vowel (i.e., *star*), the probability of the entire pre-syllable cluster occurring in an island is less than a cluster which contains just one consonant before the stressed vowel (i.e., *seven*). Therefore, we only assign stressed syllable markings to pre-syllable clusters which contain just one consonant before the stressed vowel. A stressed/unstressed syllable detector is trained similar to the broad class detector described in Chapter 3 and islands are identified using the method described in Section 5.2. Below, we compare the behavior of identifying islands via broad classes *vs.* stressed syllables.

First, we analyze the distribution of just stressed vowels in islands and gaps. Figure 5-21 shows the distribution of stressed vowels per utterance in the island and gap regions for islands identified via broad classes *vs.* stressed syllables. Each point in the island distribution shows that for all the stressed vowels per utterance, $x\%$ of the time $y\%$ of these stressed vowels are found solely in island regions. First, the figure illustrates that there is very little difference between the distribution of stressed vowels when islands are identified via broad classes or stressed syllables. In addition, both graphs illustrate that a significantly higher number of stressed vowels appear in island regions compared to gaps. For the broad class method, approximately 84% of stressed vowels appear in island regions, while for the stressed syllable method approximately 83% appear in islands. In the gap region, the broad class technique contains only about 16% of stressed vowels, while the stressed syllable approach has 17%. Both figures confirm that most of the information-bearing parts of the signal are found in the island regions for both methods, while the impoverished parts of the signal are found in the gaps.

Because stressed vowels should ideally represent stable portions of the signal, they should also be recognized with high probability. Therefore, we also analyze the recognition accuracy of stressed vowels in the island and gap regions for both

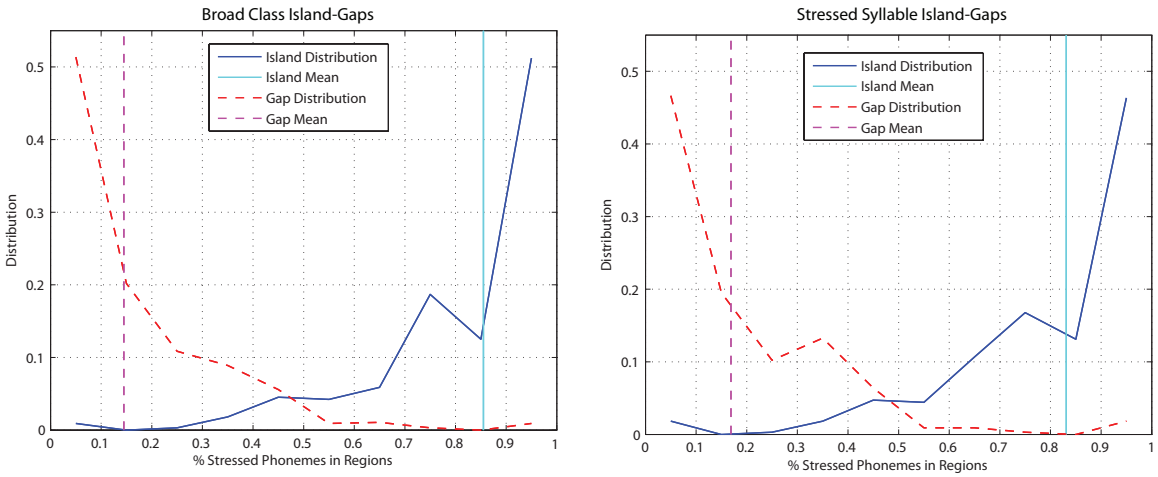

Figure 5-21: Distribution of Stressed Vowels Calculated Per Utterance on the CSAIL-info task in Island and Gap Regions

methods. Figure 5-22 shows a distribution of the percentage of correct stressed vowels in the island and gap regions. Again notice that there is little difference between the distributions for the two island detection approaches. The broad class figure indicates that approximately 84% of the stressed vowels found in island regions are correct while 81% of stressed vowels occurring in gaps are correct. The island/gap behavior for the syllable method is similar, with approximately 86% of the stressed vowels correct in island regions, while 80% in gaps are correct. From this graph and Figure 5-21 we can conclude that not only are most stressed vowels found in island regions for both island-detection techniques, but also that most of these stressed vowels are correctly hypothesized.

Finally, Figure 5-21 indicated that about 16% of stressed syllables in the broad class method occur in gap regions, while 17% occur in gaps for the stressed syllable method. Since we would expect most stressed syllables to occur in islands, we observe the length of the words which contain stressed syllables in island and gap regions. Figure 5-23 shows the distribution of the length of words, measured by the number of letters contained in the word, in island and gap regions, for both techniques. For the broad class method, note that over 51% of words in gaps have a length less than two, while more than 79% have a length less than four. Again the numbers are similar for the stressed syllable technique, with approximately 54% of words with length less

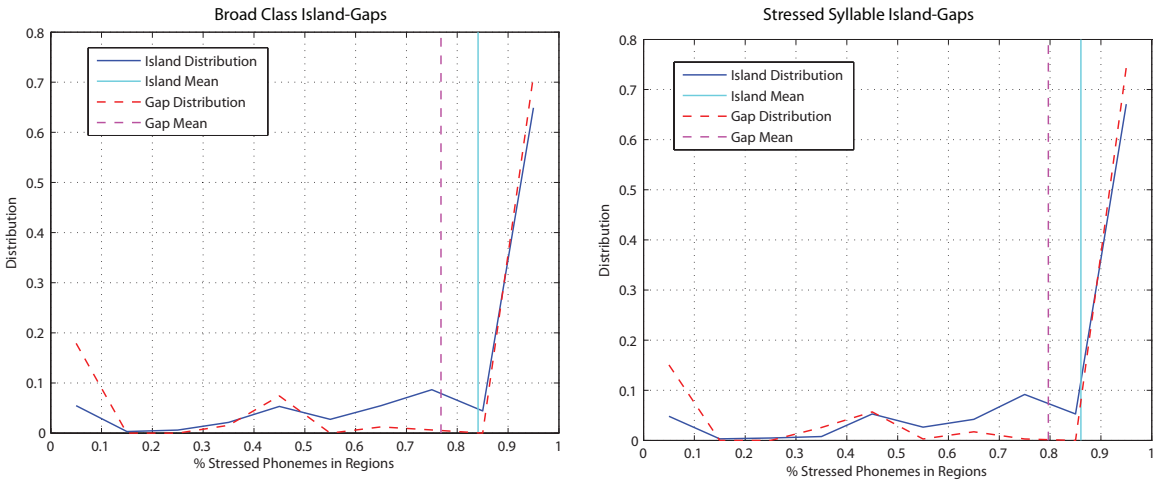

Figure 5-22: Distribution of Correct Stressed Vowels on the CSAIL-info task in Island and Gap Regions

than two and 80% with length less than four. This illustrates that many of the words found in gaps are monosyllabic function words (i.e., *a*, *it*, *is*, *the*) which are typically spoken in reduced form.

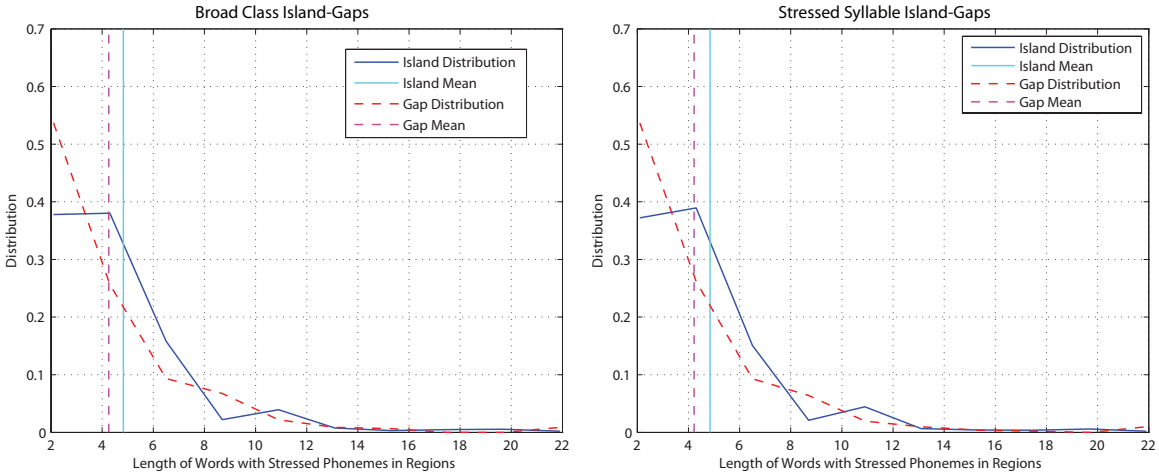

Figure 5-23: Distribution of Length of Words with Stressed Syllables on the CSAIL-info task in Island and Gap Regions

5.7.2 Error Analysis

Having confirmed that the detected islands for the CSAIL-info task are indeed reasonable, in this section we analyze the performance of the island-based segment pruning and joint broad class/phonetic model scoring methods.

Island-Based Segment Pruning

Table 5.8 shows the results for the baseline spectral change segmentation, as well as the BPC segmentation method discussed in Chapter 4 and the island-based segment pruning technique, both when islands are detected with broad classes and stressed syllables. First, notice again that the BPC Segmentation is more robust than the baseline spectral change method. In addition, both island-based segmentation techniques offer similar performance, which is no surprise given the similar behaviors illustrated in Section 5.7.1. However, both island-based techniques offer slightly worse performance than the BPC method, though a MPSSWE significance test indicates that the difference in errors rates for the two methods is not statistically significant. Since the behavior of the two island-techniques is similar, we will just focus on analyzing the island-driven broad class method further.

Method	WER
Baseline Spectral Change Segmentation	26.5
BPC Segmentation	**24.3**
Island-Based Segmentation - Broad Classes	24.8
Island-Based Segmentation - Stressed Syllables	24.8

Table 5.7: WER for Different Segmentation Techniques on CSAIL-info Task. The best performing method is indicated in bold.

One hypothesis for the slight deterioration in performance in the island-driven technique is that acoustic models are trained on the Jupiter weather system using the spectral segmentation method, which behaves more similarly to the BPC segmentation method compared to the island-based segmentation approach. We have observed in the Aurora-2 task that retraining acoustic models specific to each segmentation method offered greater improvements rather than using acoustic models trained only on the spectral change segmentation method. However, due to the limited data in the CSAIL-info training set, better performance was found using Jupiter acoustic models, rather than training acoustic models specific to each segmentation.

Taking a closer look at the segmentations, Figure 5-24 shows a cumulative distribution of the time difference between actual phonetic boundaries and landmarks for

the island and spectral change segmentation methods. The true phonetic boundaries were determined by performing a forced transcription using the BPC Segmentation technique[5]. Notice that the island technique hypothesizes a larger percentage of segments with a time difference of less than 0.05 seconds to the true phonetic boundaries relative to the spectral change method. In addition, Figure 5-25 shows a distribution of the average segments per second. Here we see that the island method has the least dense segment graph, as we would expect, while the BPC segmentation has the densest segment graph. While the island method is not as dense as either the baseline or BPC segmentation, the fact that it comes closer to detecting true phonetic landmarks compared to the spectral change segmentation gives more justification for the fact that the acoustic model training is limiting the island method.

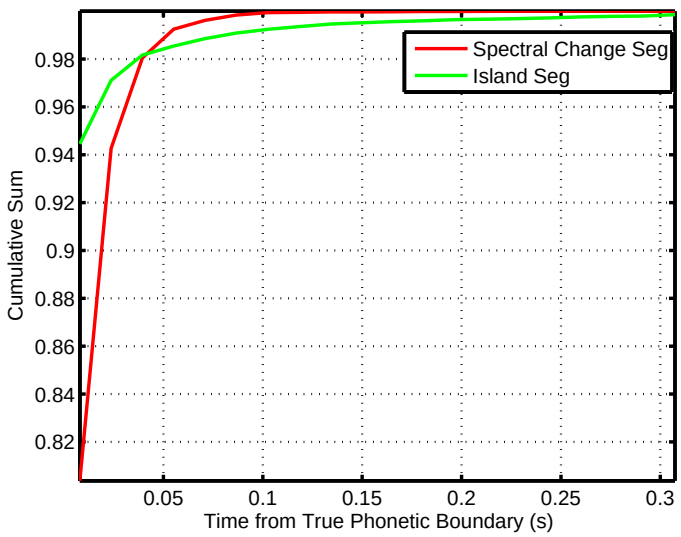

Figure 5-24: Cumulative Distribution of Time Difference Between Phonetic Boundaries and Landmarks on the CSAIL-info task

Joint Phonetic/Broad Class Models

Next, we explore the behavior of the joint broad class/phonetic approach on the CSAIL-info task. Table 5.8 shows the performance of the joint broad class/phonetic

[5]A distribution is not shown for the BPC Segmentation approach as the time difference would be zero since this segmentation was used to generate the forced transcription.

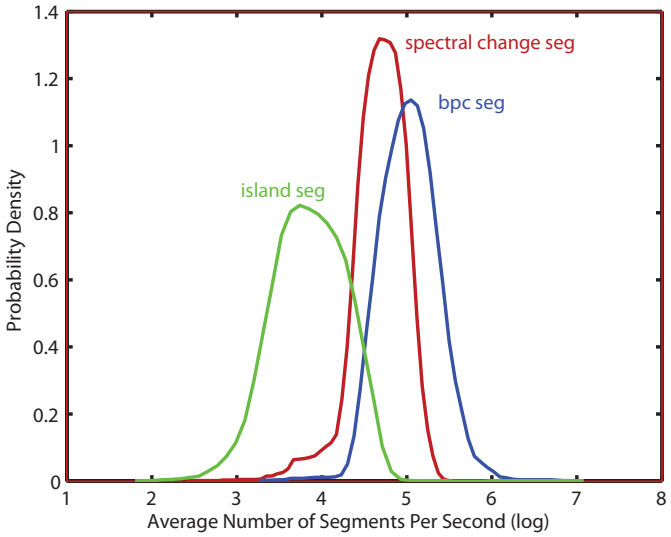

Figure 5-25: Distribution of Average Segments Per Second for Different Segmentation Methods on the CSAIL-info task

approach for various broad class splits. First, notice that using noise and nasal broad classes leads to a slight improvement in performance. However, as the number of clusters is increased past the nasal class, the error rate increases. Because of the large scale nature of the CSAIL-info task, scoring less detailed broad class models increases the confusability among words. For example, consider the words "bat" and "pat", which have the same broad class transcription. To address this issue, in the future, we would like to consider exploring a lexical access technique similar to [90], where a first pass recognition is performed to determine an N-best list of broad class/phonetic hypotheses, after which a second-pass word recognition is done over this cohort of words.

Broad Classes	WER (development)
No Broad Classes-Phonetic Models	24.8
Noise Classes (Laughter, Cough, Babble)	24.7
+Nasal	24.8
+Alveolar Closures + Labial Closures +Dental Closures	25.1
+Voiced Stops + Unvoiced Stops	25.2
+Voiced Weak Frics + Unvoiced Weak Frics	25.5

Table 5.8: WER for Different Broad Classes in Gap Region on CSAIL-info Task

5.8 Chapter Summary

In this chapter, we explored an island-driven search method which we incorporated into a modern probabilistic ASR framework. More specifically, we utilized broad class information to identify a set of island and gap regions. We illustrated that this proposed method to identify islands was able to identify vowels (in the Aurora-2 task) and stressed syllables (in the CSAIL-info corpus), typically representing information-bearing parts of the signal, with high probability.

On the Aurora-2 noisy digits task, we demonstrated that utilizing island/gap information to prune the segmentation graph resulted in an improvement in both word error rate and computation time. Furthermore, utilizing island/gap information during final recognition by scoring less detailed broad class models in gap regions resulted in further improvements in both performance and timing.

Finally on the CSAIL-info task, we showed that utilizing island information for segment pruning offered comparable performance to the BPC segmentation approach. However, further utilization of broad class knowledge in gap regions during final search resulted in a slight degradation in performance.

Chapter 6

Contributions and Future Work

6.1 Contributions

In this thesis, we explored the use of broad speech units for noise robust speech recognition. We first explored a technique to robustly recognize broad classes in noise. Then, we utilized a broad class pre-processor to develop a robust landmark detection and segmentation algorithm. Finally, we investigated the use of broad classes in island-driven search. The main contributions of the thesis are summarized in more detail in the following subsections.

6.1.1 Instantaneous Adaptation for Broad Class Detection

In Chapter 3, we introduced a novel instantaneous adaptation technique using a gradient steepness measurement derived from the Extended Baum-Welch (EBW) transformations. We incorporated this instantaneous adaptation technique into an HMM framework and we illustrated that this gradient metric allowed for a simple and effective adaptation technique which did not suffer from the data and computational intensities of other adaptation methods such as Maximum a-Posteriori (MAP), Maximum Likelihood Linear Regression (MLLR) and feature-space Maximum Likelihood Linear Regression (fMLLR).

We explored the EBW gradient metric for broad phonetic class (BPC) recognition

on the TIMIT corpus. We found that the EBW gradient method outperformed the standard likelihood technique, both when initial models are adapted via MLLR and without adaptation. In addition, we demonstrated the EBW metric captures the difference between the likelihood of an observation given the initial model and the likelihood given a model estimated from the current observation being scored, while the likelihood metric just calculates the former. We showed that this extra model re-estimation step is a main advantage of the EBW technique.

Finally, we investigated the benefits of the EBW technique in noisy conditions. We demonstrated that, when models are trained on clean speech and used to decode noisy speech, the model re-estimation inherent in the EBW algorithm allows for significant improvement over the likelihood method.

6.1.2 Utilization of Broad Class Knowledge For Landmark Detection

Segment-based speech recognition systems [31], [68] have been observed to be quite sensitive in noisy conditions [80]. Thus, in Chapter 4, we explored using the broad class HMM developed in Chapter 3 as a pre-processor to develop a robust landmark detection and segmentation algorithm using the SUMMIT segment-based speech recognition system [31]. Specifically, we explored using broad class transitions, which represent large areas of acoustic change in the audio signal, to aid in landmark detection, specifically in noisy conditions. We also compared whether these broad classes should be motivated along acoustic or phonetic dimensions, known as broad phonetic classes (BPCs) and broad acoustic classes (BACs) respectively.

We demonstrated that using either BPCs or BACs as a pre-processor for segmentation offered significant improvements in recognition performance over the baseline SUMMIT segmentation method in a variety of noise conditions on the TIMIT corpus. While the BPC and BAC methods provided similar recognition accuracy across various noise conditions and SNRs, we discovered that the BPC method provides faster computation time in non-stationary noises, while BAC is faster in stationary

conditions.

6.1.3 Utilization of Broad Class Knowledge for Island-Driven Search

Finally, in Chapter 5 we explored an island-driven search method which we incorporated into a modern probabilistic ASR framework. More specifically, we utilized broad class information to identify a set of "reliable" island and "unreliable" gap regions. We illustrated that this proposed method to identify islands was able to identify vowels (in the Aurora-2 task) and stressed syllables (in the CSAIL-info corpus), typically representing information-bearing parts of the signal, with high probability.

On the Aurora-2 noisy digits task, we showed that utilizing island/gap information to prune the segmentation graph resulted in an improvement in both word error rate and computation time. Furthermore, utilizing island/gap information during the final recognition by scoring less detailed broad class models in gap regions resulted in additional improvements in both performance and timing. Finally on the CSAIL-info task, we illustrated that utilizing island information for segment pruning offered comparable performance to the BPC segmentation approach. However, further utilization of broad class knowledge in gap regions resulted in a slight degradation in performance.

6.2 Future Work

In this section, we discuss various ideas for future work centered around the major contributions in this thesis.

6.2.1 Instantaneous Adaptation

Given the success of our gradient metric in for broad class recognition via HMMs, and the widespread use of HMMs in the speech recognition community, we would like to expand the use of this gradient metric for large vocabulary tasks. Recently, we

have applied EBW decoding to a Large Vocabulary Continuous Speech Recognition (LVCSR) task, namely transcription of English Broadcast News in the distillation portion of the Global Autonomous Language Exploitation (GALE) [15] evaluation. Some of the issues related to the choice of D and normalization on a per state basis, are being explored in this context. Our work on broad class recognition provided a good understanding of the behavior of the EBW transformations and serves as a precursor to understand the issues in the Gale large vocabulary continuous speech recognition (LVCSR) task better. We are hoping that demonstrating success of the EBW gradient metric on a larger scale task will introduce a new decoding metric into HMMs which can be applied for general speech recognition tasks.

In addition, we have also been exploring other gradient steepness metrics. For example, in [53] we provide a theoretical formulation showing that, for any technique where a distance metric can be provided, a corresponding set of model updates and gradient steepness measure can also be derived. We then derived an explicit set of model update rules and gradient steepness for the Kullback-Leibler [22] distance metric. Thus, we are also interested in comparing the behavior of various gradient metrics for numerous pattern recognition tasks, such as HMM decoding.

6.2.2 Landmark Detection and Segmentation

The broad class landmark detection method presented in Chapter 4 took advantage of broad class transitions to find a set of major landmarks and minor landmarks, but utilized spectral change knowledge to determine these landmarks. We would like to explore if landmarks can be hypothesized without acoustic information. More specifically, we are interested in hypothesizing major landmarks solely based on broad class transitions. Within a broad class, a more detailed phonetic search can be performed to determine a set of minor landmarks, which correspond to hypothetical transitions between phonemes.

In addition, in Chapter 5, we presented an island-segmentation technique which first divided the utterance into island and gap regions. A forward Viterbi and backwards A^* phonetic recognition was performed independently in each region to generate

a pruned segment graph, over which a second pass word recognition was done. We are looking at optimizing this current method by parallelizing the forward/backward search, and performing corresponding word recognition in each region, similar to [61], therefore not requiring exactly two full recognition passes.

6.2.3 Island-Driven Search

While we demonstrated the effectiveness of our proposed island-driven search technique in Chapter 5, the improvements over doing a regular search were small, particularly for the large vocabulary CSAIL-info task. We suspect that one of the main reasons is that the search remained left-to-right in the proposed technique, which still leaves the possibility for good hypotheses to be pruned away. Therefore, in the future, we are interested in exploring a search method which first starts in the reliable island regions and works outwards to the gaps, thereby utilizing the island regions more heavily and decreasing the risk of pruning away good hypotheses.

The technique presented in Chapter 5 for utilizing broad classes during final word recognition scored each broad class model by averaging all the acoustic model scores of phonemes belonging to that broad class. This approach was chosen for ease of implementation, though, when running such a system in real-time, this type of technique to score broad class models would actually increase recognition computation time. Therefore, we are interested in exploring the use of a separate set of broad class and phonetic models. Since the observation spaces used during training would be different, an appropriate scale factor would need to be empirically determined so that the ranges for the two scores would be similar.

In addition, in Chapter 5 we observed on the CSAIL-info task that using a joint broad class/phonetic model approach led to a degradation in performance. One hypothesis for this is that, on a large scale task, scoring less detailed broad class models increases the confusability among words. For example, consider the words "bat" and "pat", which have the same broad class transcription. To address this issue, in the future, we would like to consider exploring a lexical access technique similar to [90], where a first pass recognition is performed to determine an N-best

list of broad class/phonetic hypotheses, after which a second-pass word recognition is done over this cohort of words.

Finally, Chapter 5 explored using both broad class and stressed syllable information to define islands. We suspect that one of the reasons for the minimal recognition improvements with our island-driven techniques was due to our definition of islands. Thus, in the future, we would like to explore a better definition of what constitutes an island. For example, as [88] discusses, when humans process speech, they first identify distinct acoustic landmarks to segment the speech signal into articulator-free broad classes. Acoustic cues are extracted at each segment to come up with a set of features for each segment, which make use of both articulator-bound and articulator-free features. Finally, knowledge of syllable structure is incorporated to impose constraints on the context and articulation of the underlying phonemes. We would like to explore a combination of articulator-free and articulator-bound cues, in conjunction with syllable knowledge, to better define islands.

6.2.4 Broad Classes for Multi-Lingual Speech Recognition

Broad classes have been shown to provide a set of language-independent units. For example, [11] illustrates that various languages use the lexical space in a similar fashion when represented by a set of broad classes. Furthermore, [8] shows that there is a large set of phonemes which are similar across languages (i.e., poly-phonemes), while most of the language dependent information is captured by a smaller set of phonemes specific to certain languages. Therefore, in language identification experiments, clustering poly-phonemes into a set of broad classes allows for similar performance to using language-specific phonemes, while reducing computational effort. In this thesis, we explored using broad classes for acoustic landmark detection and island-driven search on English-only corpora. However, in the future, we are interested in exploring how these techniques behave in other languages.

Appendix A

Properties of Extended Baum-Welch Transformations

In this Appendix, we elaborate on various properties of the Extended Baum-Welch (EBW) transformations, which we discussed in Chapter 3.

A.1 Mathematical Understanding of EBW Transformations

In Section 3.2, we presented formulas for the EBW mean and variance update formulas, given by Equations 3.2 and 3.3 respectively. Below, we describe in more detail the intuitive meaning of these equations.

A.1.1 Linearization of EBW Mean

In [53] we explored the deeper underlying meaning of EBW these transformations, by linearizing the mean and variance parameters. First, let us rewrite Equation 3.2 from Chapter 3 as follows:

$$\hat{\mu}_j = \frac{\frac{\sum_{i=1}^{M} c_{ij} x_i}{D} + \mu_j}{\frac{\sum_{i=1}^{M} c_{ij}}{D} + 1} \tag{A.1}$$

Furthermore, we assume the following Taylor series expansion for the denominator, where terms with $1/D^2$ are combined together.

$$\frac{1}{\frac{\sum_{i=1}^{M} c_{ij}}{D} + 1} = 1 - \frac{\sum_{i=1}^{M} c_{ij}}{D} + o\left(\frac{1}{D^2}\right) \tag{A.2}$$

Substituting Equation A.2 into A.1, we get the following:

$$\hat{\mu}_j = \left(\frac{\sum_{i=1}^{M} c_{ij}}{D}\right)\mu_j + \left(1 - \frac{\sum_{i=1}^{M} c_{ij} x_i}{D}\right)\mu_j + o\left(\frac{1}{D^2}\right) \tag{A.3}$$

Assuming $\alpha = \frac{\sum_{i=1}^{M} c_{ij}}{D}$, Equation A.3 can be re-written as:

$$\hat{\mu}_j = \alpha\left(\frac{\sum_{i=1}^{M} c_{ij} x_i}{\sum_{i=1}^{M} c_{ij}}\right) + (1-\alpha)\mu_j + o\left(\frac{1}{D^2}\right) \tag{A.4}$$

Intuitively, we see that the EBW update for $\hat{\mu}_j$ is a weighted combination of the initial mean μ_j and the extremum of the associated function. Here α controls the weight given to the initial model *vs.* the model estimated by taking the extremum of the associated function.

A.1.2 Linearization of EBW Variance

Let us derive a similar linearization for the EBW variance given in Equation 3.3 from Chapter 3. Assuming the same Taylor series expansion given in Equation A.2, we can rewrite Equation 3.3 as follows:

$$\hat{\sigma}_j^2 = \alpha\left(\frac{\sum_{i=1}^{M} c_{ij} x_i^2}{\sum_{i=1}^{M} c_{ij}}\right) + (1-\alpha)(\mu_j^2 + \sigma_j^2) - \hat{\mu}_j + o\left(\frac{1}{D^2}\right) \tag{A.5}$$

Now, rewriting the linearization for the updated mean $\hat{\mu}_j^2$ as

$$\hat{\mu}_j^2 = \mu_j^2 + \alpha 2\mu_j \frac{\sum_{i=1}^{M} c_{ij}(x_i - \mu_j)}{\sum_{i=1}^{M} c_{ij}} \tag{A.6}$$

and substituting this into Equation A.5, gives the following equation for the updated variance after simplification:

$$\hat{\sigma}^2 = \alpha \left(\frac{\sum_{i=1}^{M} c_{ij} x_i^2 - 2\mu_j \sum_{i=1}^{M} c_{ij}(x_i - \mu_j) - \sum_{i=1}^{M} c_{ij}\mu_j^2}{\sum_{i=1}^{M} c_{ij}} \right) + (1-\alpha)\sigma_j^2 + o(\frac{1}{D^2}) \tag{A.7}$$

Given Equation A.7, we can also rewrite the EBW update for $\hat{\sigma}_j^2$ as a weighted combination of the initial variance σ_j^2 and the extremum of the associated function, as similarly done for $\hat{\mu}_j$.

$$\hat{\sigma}_j^2 = \alpha \left(\frac{\sum_{i=1}^{M} c_{ij}(x_i - \mu_j)^2}{\sum_{i=1}^{M} c_{ij}} \right) + (1-\alpha)\sigma_j^2 + o\left(\frac{1}{D^2}\right) \tag{A.8}$$

Again, we can observe that the EBW update equation for the variance $\hat{\sigma}_j^2$ is also a weighted combination of the initial variance σ_j^2 and the extremum of the associated function.

A.2 Behavior of EBW Adaptation Term D

A.2.1 Behavior of D in EBW-F Metric

In [84] we compared the behavior of the EBW-F, EBW-T and Likelihood metrics in classifying audio samples from the CHIL corpus [95]. In this paper, we also explored the behavior of the EBW-F classifier for various values of D, which controls the rate at which updated models are trained.

Figure A-1 shows the classification accuracy for the EBW-F metric for various values of D. Recall that this metric is calculated using Equation 3.14 from Chapter 3. The accuracy for the EBW-T (shown in Equation 3.7) and Likelihood (Equation

3.12) classifiers, which are both independent of D, are also shown for comparison.

First, for very large D the EBW-F classifier accuracy approaches that of the EBW-T. This verifies the statement given in Equation 3.5, that for large D, the EBW-F metric approaches EBW-T. As we make D smaller and train the updated model more quickly, an appropriate estimate for the updated model is still achievable and the objective function still increases with model re-estimation. It is particularly beneficial to quickly update the initial model if the slope of the objective function is relatively flat. This results in an increase in classification accuracy for the EBW-F metric.

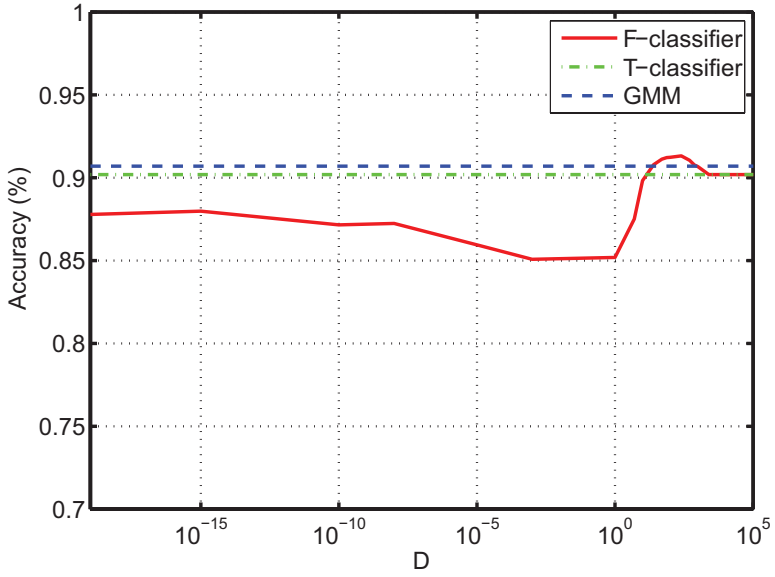

Figure A-1: Classification accuracy of EBW-F Classifier for various values of D. The EBW-F and Likelihood accuracies are also shown for comparison.

If D is too small then we train our models too quickly and do not increase the value of the objective function on each iteration. We would expect that the EBW-F accuracy should continue to decrease for smaller D. However, Figure A-1 shows that accuracy decreases for small D but then increases for very small D. To explain this factor, if we take D very small, then we can re-write Equations 3.2 and 3.3 independently of D as:

$$\hat{\mu}_j = \hat{\mu}_j(D) = \frac{\sum_{i=1}^{M} c_{ij} x_i}{\sum_{i=1}^{M} c_{ij}} \tag{A.9}$$

$$(\hat{\sigma}_j)^2 = \hat{\sigma}_j(D)^2 = \frac{\sum_{i=1}^{M} c_{ij} x_i^2}{\sum_{i=1}^{M} c_{ij}} - (\hat{\mu}_j)^2 \tag{A.10}$$

As D becomes smaller, the re-estimated model $\hat{\lambda}_j(D)$ is less influenced by the original model λ_j. However, the updated means and variances are weighted by c_{ij} from Equation 3.6, and those c_{ij} which have higher likelihood z_{ij} are weighted more. Thus, for very small D, the classifier accuracy increases as we put less weight on the poor model re-estimation and more emphasis on the initial likelihood z_{ij}. In addition, the EBW-F score moves closer to the likelihood classifier, which is influenced entirely by z_{ij}.

A.2.2 EBW Adaptive D

As Section A.2.1 illustrated, there is no mathematical rigorous method to determine the value of D, and therefore an appropriate choice for D is often accomplished via hand-tuning. To minimize the work needed to tune D, various approaches have been explored. For example, [75] explores setting D in MMI training of GMMs to be proportional to the number of data points assigned to that GMM. Intuitively, the more data assigned to a specific model, the larger D is and the less the updated model needs to be trained.

In this section, we explore a very similar idea to [75]. Conceptually, the better our original models, the less we want to train our updated models and the larger we want D. And similarly, the better our original models, the larger the log-likelihood will be. Thus, we investigate adapting the rate of model training at each frame based on the likelihood. Specifically, we explore the following linear relationship between D and log-likelihood shown in Figure A-2.

Here LL_{max} and LL_{min} are the upper and lower limits of the log likelihood determined from training data, and D_{max} and D_{min} the corresponding limits that we allow D to take. Between the likelihood limits, D is set linearly proportional to the likelihood. Intuitively, we can think of the log-likelihood as a confidence measure to

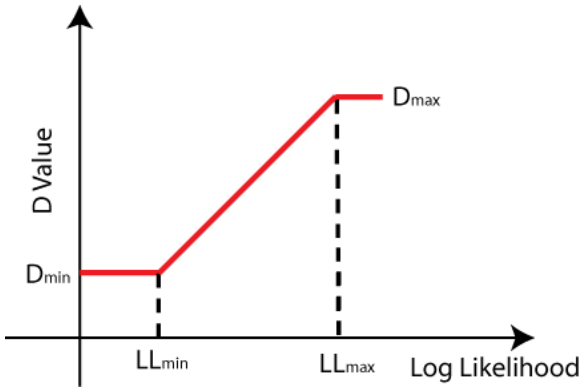

Figure A-2: Linear Transformation of Likelihood used to determine D.

determine how quickly we need to estimate the updated model.

We explore the benefits of the Adaptive D metric on the TIMIT broad phonetic class (BPC) recognition task discussed in Section 3.4. Figure A-3 shows the performance of the EBW-F Norm Global D and Adaptive D classifiers on the TIMIT development set as we globally vary D.

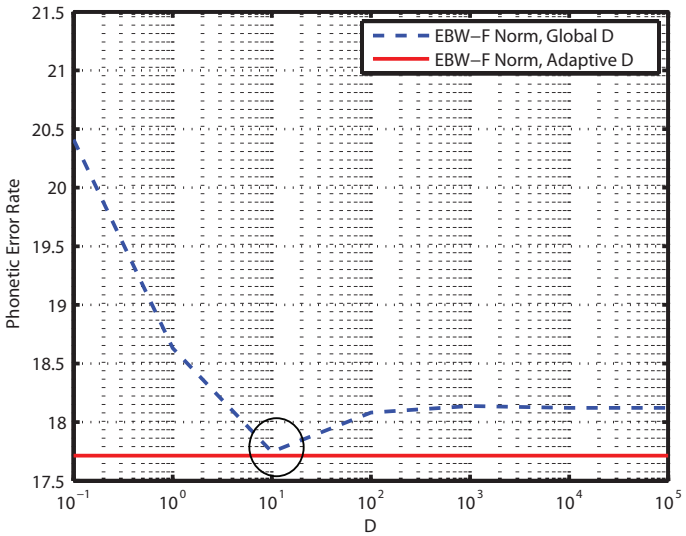

Figure A-3: Change in Phonetic Error Rate for Different EBW Metrics as D is varied. Note the large change in PER for the Global D method as a function of D. As indicated by the circle, the Adaptive D technique is able to achieve the same performance as the best Global D choice without having to heuristically tune it.

First, notice that the performance of the Global D method is quite sensitive to

the choice of D, as the PER varies by over 3% as we vary the rate at which we re-estimate updated models between 10^{-1} and 10^{5}. If D is made smaller and the updated models are trained more quickly, we are still able to get an appropriate estimate for the updated model while allowing the objective function to increase. However, if we take D to be too small and train our models too quickly, the value of the objective function is not guaranteed to increase on each iteration. Therefore, the performance of the Global D metric decreases.

However, as indicated by the circle in Figure A-3, if the likelihood scores are used as a confidence measure to linearly adapt D, the Adaptive D metric has similar performance to the best performing value of D in the global D method, without having to heuristically tune D.

Appendix B

Phonetic Symbols

In this Appendix, we provide the IPA, ARPAbet and Broad Phonetic Class (BPC) symbols for phones in the English Language, as listed in Table B.1.

IPA	ARPA	BPC	Example	IPA	ARPA	BPC	Example
[ɑ]	aa	vowel	b*o*b	[ɨ]	ix	vowel	deb*i*t
[æ]	ae	vowel	b*a*t	[i]	iy	vowel	b*ee*t
[ʌ]	ah	vowel	b*u*t	[j]	jh	strong fricative	*j*oke
[ɔ]	ao	vowel	b*ough*t	[k]	k	stop	*k*ey
[ɑʷ]	aw	vowel	b*ou*t	[k˺]	kcl	closure	k closure
[ə]	ax	vowel	*a*bout	[l]	l	semi-vowel	*l*ay
[əʰ]	ax-h	vowel	p*o*tato	[m]	m	nasal	*mom*
[ɚ]	axr	vowel	butt*er*	[n]	n	nasal	*noon*
[ɑʸ]	ay	vowel	b*i*te	[ŋ]	ng	nasal	si*ng*
[b]	b	stop	*b*ee	[ɾ̃]	nx	nasal	wi*nn*er
[b˺]	bcl	closure	b closure	[o]	ow	vowel	b*oa*t
[č]	ch	strong fricative	*ch*oke	[ɔʸ]	oy	vowel	b*oy*
[d]	d	stop	*d*ay	[p]	p	stop	*p*ea
[d˺]	dcl	closure	d closure	[▫]	pau	closure	pause
[ð]	dh	weak fricative	*th*en	[p˺]	pcl	closure	p closure
[ɾ]	dx	weak fricative	mu*dd*y	[ʔ]	q	stop	glottal stop
[ɛ]	eh	vowel	b*e*t	[r]	r	semi-vowel	*r*ay
[l̩]	el	semi-vowel	bott*le*	[s]	s	strong fricative	*s*ea
[m̩]	em	nasal	bott*om*	[š]	sh	strong fricative	*sh*e
[n̩]	en	nasal	butt*on*	[t]	t	stop	*t*ea
[ŋ̍]	eng	closure	Wash*ing*ton	[t˺]	tcl	closure	t closure
[▯]	epi	closure	epenthetic silence	[θ]	th	weak fricative	*th*in
[ɝ]	er	semi-vowel	b*ir*d	[ʊ]	uh	vowel	b*oo*k
[e]	ey	vowel	b*ai*t	[u]	uw	vowel	b*oo*t
[f]	f	weak fricative	*f*in	[ü]	ux	vowel	t*oo*t
[g]	g	stop	*g*ay	[v]	v	weak fricative	*v*an
[g˺]	gcl	closure	g closure	[w]	w	weak fricative	*w*ay
[h]	hh	weak fricative	*h*ay	[y]	y	weak fricative	*y*acht
[ɦ]	hv	weak fricative	a*h*ead	[z]	z	strong fricative	*z*one
[ɪ]	ih	vowel	b*i*t	[ž]	zh	strong fricative	a*z*ure
-	h#	silence	utterance initial and final silence				

Table B.1: IPA, ARPAbet and Broad Phonetic Class (BPC) symbols for the phones in the English Language with sample occurrences

Bibliography

[1] Merriam-Webster Online Dictionary. http://www.merriam-webster.com, May 2009.

[2] S. Abdou and M. S. Scordilis. Beam Search Pruning in Speech Recognition Using a Posterior Probability-Based Confidence Measure. *Speech Communication*, 42(3-4):409–428, 2004.

[3] J. B. Allen. How Do Humans Process and Recognize Speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):567–577, 1994.

[4] B. S. Atal. Effectiveness of Linear Prediction Characteristics of the Speech Waveform for Automatic Speaker Identification and Verification. *Journal of the Acoustical Society of America*, 55(6):1304–1312, 1974.

[5] M. Bacchiani and M. Ostendorf. Joint Lexicon, Acoustic Unit Inventory and Model Design. *Speech Communication*, 29(2-4):99–114, 1999.

[6] L. E. Baum and J. A. Eagon. An Inequality with Applications to Statistical Prediction for Functions of Markov Processes and to a Model of Ecology. *Bulletin of the American Mathematical Society*, 73:360–363, 1963.

[7] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A Maximization Technique Occuring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[8] K. Berkling, T. Arai, E. Barnard, and R. A. Cole. Analysis of Phoneme-Based Features for Language Identification. In *Proc. ICASSP*, pages 289–292, 1994.

[9] S. Boll. Suppression of Acoustic Noise in Speech Using Spectral Subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 2(2):113– 120, 1979.

[10] G. Box and D. Cox. An Analysis of Transformations. *Journal of Royal Statistical Society*, 26(2):211–246, 1964.

[11] R. Carlson, K. Elenius, O. E. Granström, and S. Hunnicutt. Phonetic and Orthographic Properties of the Basic Vocabulary of Five European Languages. In *Proc. ICASSP*, pages 2763–2766, 1986.

[12] P. P. Chakrabarti, S. Ghose, and S. C. DeSarkar. Heuristic Search Through Islands. *Artificial Intelligence*, 29(3):339–347, 1986.

[13] J. Chang and J. R. Glass. Segmentation and Modeling in Segment-Based Recognition. In *Proc. ICASSP*, pages 1199–1202, 1997.

[14] N. Chomsky and M. Halle. *Sound Pattern of English*. Harper & Row, 1968.

[15] J. Cohen. The GALE Project: A Description and an Update. In *Proc. ASRU*, pages 237–237, 2007.

[16] M. Cooke, P. Green, L. Josifovski, and A. Vizinho. Robust Automatic Speech Recognition with Missing and Unreliable Acoustic Data. *Speech Communication*, 34(3):267–285, 2001.

[17] A. Corazza, R. De Mori, R. Gretter, and G. Satta. Computation Probabilities for an Island-Driven Parser. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):936–950, 1991.

[18] X. Cui, M. Afify, and Y. Gao. MMSE-Based Stereo Feature Stochastic Mapping for Noise Robust Speech Recognition. In *Proc. ICASSP*, pages 4077–4080, 2008.

[19] S. Das, R. Bakis, A. Nadas, D. Nahamoo, and M. Picheny. Influence of Background Noise and Microphone on the Performance of the IBM Tangora Speech Recognition System. In *Proc. ICSASP*, pages 71–74, 1993.

[20] S. Davis and P. Mermelstein. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366, 1980.

[21] J. F. Dillenburg. *Techniques for Improving the Efficiency of Heuristic Search*. PhD thesis, Univeristy of Illinois at Chicago, 1993.

[22] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley Interscience, 2nd edition, 2001.

[23] T. Fabian, R. Lieb, G. Ruske, and M. Thomae. A Confidence-Guided Dynamic Pruning Approach - Utilization of Confidence Measurement in Speech Recognition. In *Proc. Interspeech*, pages 585–588, 2005.

[24] G. Fant. *Acoustic Theory of Speech Production*. Mouton and Co., 's-Gravenage, Netherlands, 1960.

[25] M. J. F. Gales. The Generation and Use of Regression Class Trees for MLLR Adaptation. Technical report, Cambridge University, 1996.

[26] M. J. F. Gales and S. Young. Robust Continuous Speech Recognition using Parallel Model Combination. *IEEE Transactions on Speech and Audio Processing,*, 4(5):352 – 359, 1996.

[27] Y. Gao and J-P. Haton. Noise Reduction and Speech Recognition in Noise Conditions Tested on LPNN-Based Continuous Speech Recognition System. In *Proc. Eurospeech*, pages 1035–1038, 1993.

[28] J. L. Gauvain and C. H. Lee. Maximum A-Posteriori Estimation for Multivariate Gaussian Mixture Observations on Markov Chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, 1994.

[29] L. Gillick and S. Cox. Some Statistical Issues in the Comparison of Speech Recognition Algorithms. In *Proc. ICASSP*, pages 532–535, 1989.

[30] J. R. Glass. *Finding Acoustic Regularities in Speech: Applications to Phonetic Recognition*. PhD thesis, MIT, 1988.

[31] J. R. Glass. A Probabilistic Framework for Segment-Based Speech Recognition. *Computer Speech and Language*, 17(2-3):137–152, 2003.

[32] J. R. Glass, T. J. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay. Recent Progress in the MIT Spoken Lecture Processing Project. In *Proc. Interspeech*, 2007.

[33] J. R. Glass, T. J. Hazen, and I. L. Hetherington. Real-time Telephone-Based Speech Recognition in the JUPITER Domain. In *Proc. ICASSP*, pages 61–64, 1999.

[34] Y. Gong. Speech Recognition in Noise Environments: A Survey. *Speech Communication*, 16(3):261–291, 1995.

[35] P. S. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo. A Generalization of the Baum Algorithm to Rational Objective Functions. In *Proc. ICASSP*, number 631-634, 1989.

[36] A. K. Halberstadt. *Heterogeneous Acoustic Measurements and Multiple Classifiers for Speech Recognition*. PhD thesis, Massachusetts Institute of Technology, 1998.

[37] A. K. Halberstadt and J. R. Glass. Heterogeneous Acoustic Measurements for Phonetic Classification. In *Proc. Eurospeech*, pages 401–404, 1997.

[38] A. K. Halberstat and J. R. Glass. Heterogeneous Measurements and Multiple Classifiers for Speech Recognition. In *Proc. ICSLP*, pages 995–998, 1998.

[39] T. J. Hazen. Visual Model Structures and Synchrony Constraints for Audio-Visual Speech Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 14(3):1082–1089, 2006.

[40] T. J. Hazen, I. L. Hetherington, H. Shu, and K. Livescu. Pronunciation Modeling using a Finite-State Transducer Representation. *Speech Communication*, 46(2):189–203, 2005.

[41] T. J. Hazen and V. W. Zue. Automatic Language Identification using a Segment-Based Approach. In *Proc. Eurospeech*, 1993.

[42] H. Hermansky. Perceptual Linear Predictive (PLP) Analysis of Speech. *Journal of the Acoustic Society of America*, 87(4):1738–1752, 1990.

[43] H. Hermansky and N. Morgan. RASTA Processing of Speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578 – 589, 1994.

[44] I. L. Hetherington, M. Phillips, J.R. Glass, and V. W. Zue. A* Word Network Search for Continuous Speech Recognition. In *Eurospeech*, pages 1533–1536, 1993.

[45] H. G. Hirsch and D. Pearce. The AURORA Experimental Framework for the Performance Evaluations of Speech Recognition Systems under Noisy Condidions. In *ISCA ITRW ASR2000 "Automatic Speech Recognition: Challenges for the Next Millennium"*, pages 181–188, 2000.

[46] A. S. House and E. P. Neuburg. Toward Automatic Indentification of the Language of an Utterance I. Preliminary Methodological Consideration. *JASA*, 62(3):708–713, 1977.

[47] X. Huang, A. Acero, and H. W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development* . Prentice Hall, 2001.

[48] D. P. Huttenlocher and V. W. Zue. A Model of Lexical Access for Partial Phonetic Information. In *Proc. ICASSP*, pages 391–394, 1984.

[49] B. H. Juang. Speech Recognition in Adverse Environments. *Computer Speech and Language*, 5(33):275–294, 1991.

[50] S. O. Kamppari and T. J. Hazen. Word and Phone Level Acoustic Confidence Scoring. In *Proc. ICASSP*, pages 1799–1802, 2000.

[51] D. Kanevsky. Extended Baum Transformations for General Functions. In *Proc. ICASSP*, pages 17–21, 2004.

[52] D. Kanevsky. Extended Baum Transformations For General Functions, II. Technical report, Human Language Technologies, IBM, 2005.

[53] D. Kanevsky, T. N. Sainath, and B. Ramabhadran. A Generalized Family of Parameter Estimation Techniques. In *Proc. ICASSP*, pages 1725–1728, 2009.

[54] K. Kirchhoff, G. A. Fink, and G. Sagerer. Combining Acoustic and Articulatory Feature Information for Robust Speech Recognition. *Speech Communication*, 37(3-4):303–319, 2002.

[55] F. Kubala, J. Bellegarda, J. Cohen, D. Pallett, D. Paul, M. Phillips, R. Rajasekaran, F. Richardson, M. Riley, R. Rosenfeld B. Roth, and M. Weintraub. The Hub and Spoke Paradigm for Continuous Speech Recognition. In 37-42, editor, *Proc. ARPA Human Language Technology Workshop*, 1994.

[56] R. Kumaran, J. Bilmes, and K. Kirchhoff. Attention Shift Decoding for Conversational Speech Recognition. In *Proc. Interspeech*, pages 1493–1496, 2007.

[57] L. Lamel, R. Kassel, and S. Seneff. Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus. In *Proc. of the DARPA Speech Recognition Workshop*, pages 61–70, 1986.

[58] W. A. Lea. *Trends in Speech Recognition*. Prentice Hall, Englewood Cliffs, NJ, 1980.

[59] C. H. Lee, B. H. Juang, F. K. Soong, and L. R. Rabiner. Word Recognition Using Whole Word and Subword Models. In *Proc. ICASSP*, pages 683–686, 1989.

[60] C. H. Lee, F. K. Soong, and B. H. Juang. A Segment Model Based Approach to Speech Recognition. In *Proc. ICASSP*, pages 501–504, 1988.

[61] S. C. Lee and J. R. Glass. Real Time Probabilistic Segmentation for Segment-Based Speech Recognition. In *Proc. ICSLP*, pages 1803–1806, 1998.

[62] C. J. Leggetter and P. C. Woodland. Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. *Computer Speech and Language*, 9(2):171–185, 1995.

[63] Y. Li, H. Erdogan, Y. Gao, and E. Marcheret. Incremental Online Feature Space MLLR Adaptation for Telephony Speech Recognition. In *Proc. ICSLP*, pages 1417–1420, 2002.

[64] R. P. Lippmann. Speech Recognition by Machines and Humans. *Speech Communication*, 22(1):1–15, 1997.

[65] G. A Miller and P. E. Nicely. An Analysis of Perceptual Confusions Among Some English Consonants. *Journal of the Acoustical Society of America*, 27(2):338–32, 1955.

[66] P. J. Moreno and R. M. Stern. Sources of Degradation of Speech Recognition in the Telephone Network. In *Proc. ICASSP*, pages 109–112, 1994.

[67] Y. Normandin, R. Cardin, and R. De Mori. High-Performance Connected Digit Recognition using Maximum Mutual Information Estimation. *IEEE Transactions on Speech and Audio Processing*, 2(2):299–311, 1994.

[68] M. Ostendorf, V. Digalakis, and O. Kimball. From HMMs to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition. *IEEE Transactions on Speech and Audio Proceesing*, 4(5):360–378, 1996.

[69] M. Padmanabhan and S. Dharanipragada. Maximum-Likelihood Nonlinear Transformation for Acoustic Adaptation. *IEEE Transactions on Speech and Audio Processing*, 12(6):572 – 578, 2004.

[70] D. Pallet, W. Fisher, and J. Fiscus. Tools for the Analysis of Benchmark Speech Recognition Tests. In *Proc. ICASSP*, pages 97–100, 1990.

[71] C. Park. *Consonant Landmark Detection for Speech Recognition*. PhD thesis, Massachusetts Institute of Technology, 2008.

[72] H. L. Pick, G. M. Siegel, and P. W. Fox. Inhibiting the Lombard Effect. *Journal of the Acoustical Society of America*, 85(2):894–900, 1989.

[73] J. F. Pitrelli, J. Subrahmonia, and B. Maison. Toward Island-of-Reliability-Driven Very-Large-Vocabulary On-Line Handwriting Recognition Using Character Confidence Scoring. In *Proc. ICASSP*, pages 1525–1528, 1991.

[74] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah. Boosted MMI for Model and Feature-Space Discriminative Training. In *Proc. ICASSP*, pages 4057–4060, 2008.

[75] D. Povey and B. Kingsbury. Evaluation of Proposed Modifications to MPE for Large Scale Discriminative Training. In *Proc. ICASSP*, pages 321–324, 2007.

[76] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE*, 77(2):257–286, 1989.

[77] Y. Rabner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[78] A. Rosenberg and J. Hirschberg. V-Measure: A Conditional Entropy-Based External Cluster Evaluation. In *Proc. EMNLP-CoNLL*, pages 410–420, 2007.

[79] T. N. Sainath. Acoustic Landmark Detection and Segmentation using the McAulay-Quatieri Sinusoidal Model. Master's thesis, MIT, 2005.

[80] T. N. Sainath and T. J. Hazen. A Sinusoidal Model Approach to Acoustic Landmark Detection and Segmentation for Robust Segment-Based Speech Recognition. In *Proc. ICASSP*, pages 525–528, 2006.

[81] T. N. Sainath, D. Kanevsky, and G. Iyengar. Unsupervised Audio Segmentation using Extended Baum-Welch Transformations. In *Proc. ICASSP*, pages 209–212, 2007.

[82] T. N. Sainath, D. Kanevsky, and B. Ramabhadran. Broad Phonetic Class Recognition in a Hidden Markov Model Framework using Extended Baum-Welch Transformations. In *Proc. ASRU*, pages 306–311, 2007.

[83] T. N. Sainath, D. Kanevsky, and B. Ramabhadran. Gradient Steepness Metrics Using Extended Baum-Welch Transformations for Universal Pattern Recognition Tasks. In *Proc. ICASSP*, pages 4533–4536, 2008.

[84] T. N. Sainath, V. W. Zue, and D. Kanevsky. Audio Classification using Extended Baum-Welch Transformations. In *Proc. Interspeech*, 2969-2972, 2007.

[85] P. Scanlon, D. Ellis, and R. Reilly. Using Broad Phonetic Group Experts for Improved Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):803–812, 2007.

[86] F. Sha and L. Saul. Comparison of Large Margin Training to Other Discriminative Methods for Phonetic Recognition by Hidden Markov Models. In *Proc. ICASSP*, pages 313–316, 2007.

[87] K. N. Stevens. *Acoustic Phonetics*. The MIT Press, Cambridge, MA, 1998.

[88] K. N. Stevens. Toward A Model for Lexical Access Based on Acoustic Landmarks and Distinctive Features. *Journal of the Acoustic Society of America*, 111(4):1872–1891, 2002.

[89] M. Tang, S. Seneff, and V. W. Zue. Modeling Linguistic Features in Speech Recognition. In *Proc. Eurospeech*, pages 2585–2588, 2003.

[90] M. Tang, S. Seneff, and V. W. Zue. Two-Stage Continuous Speech Recognition Using Feature-Based Models: A Preliminary Study. In *Proc. ASRU*, pages 49–54, 2003.

[91] V. Valtchev, J. J. Odell, P. C. Woodland, and S. J. Young. MMIE Training of Large Vocabulary Speech Recognition Systems. *Speech Communication*, 22(4):303–314, 1997.

[92] A. Varga. Assessment for Automatic Speech Recognition: II. Noisex-92: A Database and an Experiment to Study the Effect of Additive Noise on Speech Recognition Systems. *Speech Communication*, 12(3):247–251, 1993.

[93] A. Varga, H. J. M. Steeneken, M. Tomlinson, and D. Jones. The Noisex-92 Study on the Effect of Additive Noise on Automatic Speech Recognition. Technical report, DRA Speech Research Unit, 1992.

[94] A. J. Viterbi. Error Bounds for Convolution Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[95] A. Waibel, H. Steusloff, R. Stiefelhagen, and the CHIL Project Consortium. CHIL: Computers in the Human Interaction Loop. In *5th International Workshop on Image Analysis for Multimedia Interactive Services*, 2004.

[96] C. Wang and S. Seneff. Lexical Stress Modeling for Improved Speech Recognition of Spontaneous Telephone Speech in the JUPITER Domain. In *Proc. Eurospeech*, pages 2761–2765, 2001.

[97] K. Wang, S. A. Shamma, and W. J. Byrne. Noise Robustness in the Auditory Representation of Speech Signals. In *Proc. ICASSP*, pages 335–338, 1993.

[98] J. Wolf and W. Woods. The HWIM Speech Understanding System. In *Proc. ICASSP*, pages 784–787, 1977.

[99] W. A. Woods. *Language Processing for Speech Understanding*. Prentice Hall International (UK) Ltd., 1985.

[100] S. J. Young, J. J. Odell, and P. C. Woodland. Tree-Based State Tying for High Accuracy Acoustic Modelling. In *Proc. HLT*, pages 307–312, 1994.

[101] Q. Zhu and A. Alwan. On the Use of Variable Frame Rate Analysis in Speech Recognition. In *Proc. ICASSP*, pages 1783–1786, 2000.

[102] V. W. Zue. The Use of Speech Knowledge in Automatic Speech Recognition. *IEEE Special Issue on Human-Machine Communication by Speech*, 73(11):1602–1615, 1985.

[103] V. W. Zue. On Organic Interfaces. In *Proc. Interspeech*, pages 1–8, 2007.