VOICE QUERY-BY-EXAMPLE FOR RESOURCE-LIMITED LANGUAGES USING AN ERGODIC HIDDEN MARKOV MODEL OF SPEECH

A Dissertation
Presented to
The Academic Faculty

By

Asif Ali

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2013

VOICE QUERY-BY-EXAMPLE FOR RESOURCE-LIMITED LANGUAGES USING AN ERGODIC HIDDEN MARKOV MODEL OF SPEECH

Approved by:

Professor Mark A. Clements, Advisor
Professor, School of ECE
Georgia Institute of Technology

Professor Chin-Hui Lee
Professor, School of ECE
Georgia Institute of Technology

Professor David V. Anderson
Professor, School of ECE
Georgia Institute of Technology

Professor John Copeland
Professor, School of ECE
Georgia Institute of Technology

Professor Alexandar Lerch
Asst. Professor, School of Music
Georgia Institute of Technology

Date Approved: 14 November 2013

To family and good friends

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my thesis advisor, Professor Mark A. Clements, from whom I have learned so much, about life and research. He has been a true inspiration, and I thank him for his guidance in helping me develop as a researcher and as an individual.

Next, I would like to thank the members of my thesis committee: Prof. Chin-Hui Lee, Prof. David V. Anderson, Prof. John Copeland, and Prof. Alexander Lerch, for their many insightful comments and suggestions. I would also like to acknowledge and thank Nexidia Inc. for their contributions and help in this research.

I would also like to thank the Fulbright Scholarship Commission, and its funding agencies, the U.S. Department of State's Bureau of Educational and Cultural Affairs, and Higher Education Commission, Pakistan, for supporting this research. I would also like to express my gratitude to the scholarship administrators at Institute of International Education and United States Education Foundation, Pakistan.

I would like to thank Jennifer Lunsford at Center for Signal and Information Processing, and Dr. Daniela Staiculescu at ECE graduate office. I wish to thank my fellow graduate students and researchers, Hrishikesh Rao, Ahmad Beirami, Mehrez Souden, Samir Rustamov, Jonathan Kim and many others, for their friendship, help, and comraderie.

Most importantly, I wish to thank my family: my parents, my lovely wife, my brothers and sisters, for loving me, for believing in me, and for supporting me every step of the way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

An ergodic hidden Markov model (EHMM) can be useful in extracting underlying structure embedded in connected speech without the need for a time-aligned transcribed corpus. In this research, we present a query-by-example (QbE) spoken term detection system based on an ergodic hidden Markov model of speech. An EHMM-based representation of speech is not invariant to speaker-dependent variations due to the unsupervised nature of the training. Consequently, a single phoneme may be mapped to a number of EHMM states. The effects of speaker-dependent and context-induced variation in speech on its EHMM-based representation have been studied and used to devise schemes to minimize these variations. Speaker-invariance can be introduced into the system by identifying states with similar perceptual characteristics. In this research, two unsupervised clustering schemes have been proposed to identify perceptually similar states in an EHMM. A search framework, consisting of a graphical keyword modeling scheme and a modified Viterbi algorithm, has also been implemented. An EHMM-based QbE system has been compared to the state-of-the-art and has been demonstrated to have higher precisions than those based on static clustering schemes.

CHAPTER 1
INTRODUCTION

The goal of developing an automatic, unsupervised speech recognition system has eluded researchers for the last 50 years. Although significant progress has been made in this period, training a speech recognition system still requires considerable human supervision, and as yet, the trained systems are not comparable in accuracy to human perception.

The development of a speech recognition system is a resource-intensive process, dependent on the availability of a large amount of time-aligned labeled speech for training. In traditional approaches to speech recognition, separate statistical models are trained for each unit of speech. These units of speech can be whole words or subword units such as phonemes. For each acoustic entity, exemplars are isolated from a large training set and labeled accordingly. These isolated exemplars are then used to train the corresponding left-right hidden Markov models for each acoustic entity.

There are nearly seven thousand different languages spoken around the world today. The majority of the seven billion inhabitants of the world speak just a few of these languages, e.g., English, Mandarin, Spanish, or Hindi. According to some estimates, half of the languages in use today have less than a thousand native speakers.

Out of the thousands of spoken languages, speech recognition systems have been developed for a small number of widely spoken languages. The majority of languages lack the linguistic resources required for training a new speech recognition system. Even if the linguistic structure of a language is known, there may not be enough trained linguists available to transcribe the huge amount of speech data required for training. It is clear that current methodologies for training speech recognition systems cannot be extended to new resource-limited languages.

Different languages generally have a different set of building blocks or phonemes with different morphological rules controlling the composition of these units. For instance, the phonetic units of a tonal language, such as Mandarin, are significantly different from those in Germanic languages. As a result, statistical models trained for one language generally have limited potential for migration to other languages.

The goal of this research is to design a voice query-by-example (VQbE) system for resource-limited languages using an ergodic hidden Markov model (EHMM) of speech. EHMMs have been used successfully in a number of different applications including language identification [1], gender identification [2], and speech coding [3]. However, there has been limited success in their use for speech recognition [4, 5].

In the next section, a brief overview of the speech generation mechanism is presented, and common sources of variation in speech are described. The state-of-the-art in KWS systems is covered in Section 1.2, leading to the presentation of the proposed system based on a single, large EHMM of speech in Chapter 2.

1.1 Origin and History of the Problem

The design of a keyword spotting system (KWS) for resource-limited languages has been the focus of many research efforts in recent years. Since the orthographic transcription of speech is generally not available for these languages, KWS systems take the form of a VQbE system where the keyword search algorithm accepts a query in the form of a speech waveform.

The goal of a KWS system is to recognize all the occurrences of an utterance in a body of voice data. To do so, the system must look for those acoustic features that characterize an utterance while safely ignoring those that do not alter its meaning. A brief overview of the process of speech production to highlight the mechanisms responsible for generating the different features of speech is provided in the following sections.

This introduction continues with an overview of the different KWS systems proposed over the years.

1.1.1 Speech Production

Speech contains a wealth of information. In addition to the acoustic features that define an utterances, speech also encode non-linguistic cues containing context and speaker-specific information. Human beings are acutely perceptive of these cues and can use them to identify a speaker's identity (if known), gender, social and emotional state, and even general health. In addition to the speaker-dependent variations, speech also exhibits contextual variations such as prosodic variations in loudness or pitch in different situations.

The focus of this section is to introduce the characteristic features of speech and to contrast these with the contextual and speaker-dependent variations that do not change the perceptual qualities of a speech segment. The various features of speech have their origin in the speech production mechanism. At its most basic, speech involves the generation of sound when air is pushed through the vocal cords, and the manipulation of this sound by the articulators to form distinct units of speech.

The articulation process is initiated in the glottis which contains the vocal folds. The glottis changes the characteristics of speech using a process known as phonation. The rhythmic opening and closing of the vocal folds gives rise to voiced speech. Voiced speech has a harmonic quality that distinguishes it from unvoiced speech.

In unvoiced or voiceless speech, the glottis is wide open and sound is produced primarily by the constrictions created by the articulators. Phonemes such as /V/[1] and /F/ are respective examples of voiced and voiceless phonemes. These phonemes have the articulators in the same configuration and can only be distinguished by the difference in the phonation process. The phoneme /F/ has a noise-like quality while /V/ is characterized by the presence of harmonics, generated by the oscillating vocal folds.

The phonation process determines the spectral character of a speech segment. The input excitation, in voiced segments, can be modeled as a series of periodic pulses and manifests itself as a series of evenly spaced peaks in the spectrogram of speech. The frequency of the

[1] ARPABET notation has been used to denote phonemes in this paper.

glottal pulse is called the fundamental frequency or $F0$ and is related to the perceived pitch of the speech signal.

In the case of unvoiced speech, the emitted sound is mostly influenced by the articulators with the glottis being wide open. The spectrum of the speech signal is wider and lacks the characteristic harmonic structure of voiced speech. Since the vocal cords are wide open, unvoiced speech segments generally have lower energy and shorter duration compared to the voiced phonemes.

The articulators in various configurations, determine the unique identity of each phoneme within the broad classes of voiced and voiceless speech. The oral and nasal cavities act as a resonating chamber while the velum, tongue, teeth, and the lips manipulate the structure of this resonating chamber.

The opening or closing of the nasal cavity by the velum changes the size and shape of the vocal tract, and consequently, alter the spectral characteristics of the vocal tract. In addition to controlling the resonance frequencies, the articulators, such as lips, teeth, and the tongue, can also be used to form constrictions along the vocal tract to produce turbulence required to form some consonants.

The resonances produced by the articulators appear as peaks in the frequency spectrum. Compared to the closely spaced harmonic peaks associated with the pitch, these resonant peaks are wider and form the envelope of the harmonics of the fundamental frequency. These resonant peaks are known as formants and are denoted as $F1$,$F2$,$F3$ and so on.

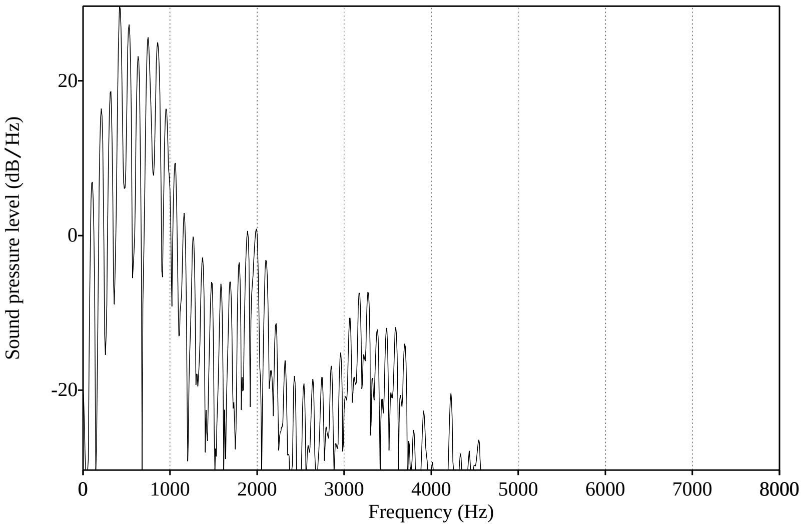


Figure 1: The spectrogram of the vowel /AO/.

The spectrogram for a sample of the vowel /AO/ is shown in Figure1 . The voiced nature of the speech segment is evident by the presence of closely spaced harmonics in the spectrogram. These harmonics lie at integer multiples of the fundamental frequency. The fundamental frequency can be estimated by calculating the distance between the consecutive peaks in the spectrogram. There are roughly nine peaks in a 1000 Hz wide segment of spectrogram, leading to an estimate of 110 Hz for the fundamental frequency.

The formants, on the other hand, form the spectral peaks enveloping the harmonics in the spectrogram. The first two formants lie approximately at 500 Hz and 900 Hz. The spectrogram also reveals the existence of additional formants at 2000 Hz and 3400 Hz.

1.1.2 Acoustic Properties of Phonemes

Each phoneme has a distinct acoustic character that can be tracked to a particular configuration of the vocal chords and the articulators. The air rushed through the vocal folds form the excitation signal. In voiced speech, the vocal folds are vibrating whereas unvoiced speech is characterized by an open configuration of the vocal fold and a sound is produced by an obstruction in the vocal tract. This constriction can be in the form of a complete stop such as in stop consonants or it can be partial giving rise to fricatives and approximants. The position of the obstruction also varies between different consonants. It can be formed at the lips, or by the tip, blade or back of the tongue.

The velum controls the opening to the nasal cavity. A velic closure, as its name indicate, closes the nasal cavity thus restricting the speech generation to the oral cavity. In nasal consonants, on the other hand, the velum is lowered to open the nasal cavity while the oral cavity is obstructed by the articulators.

In the case of vowels, the input excitation is generated by the oscillations of the vocal folds and the spectrum of this excitation signal is shaped by the articulators. However, the vocal tract is not obstructed like in the case of consonants but overtones of the pitch are produced by a narrowing of the vocal tract by the articulators. In front vowels, as the name suggests, the articulators that are responsible for shaping the spectrum of the speech signal, lie in the front of the vocal tract. The front vowels are produced by varying the position of the tip of the tongue and the opening of the mouth. Similarly, the back vowels are produced by raising the body of the tongue at the back of vocal track. These vowels can be further subdivided into high and low vowels based on the position of an articulators. The vowel /IX/ is a high-front vowel because the tip of the tongue is high and at the front of the mouth whereas it is back and low in the case of /AA/ which can be categorized as a low-back vowel. Finally, the shape of the lips determine the roundedness of a vowel. For example, /OW/ is a rounded vowel while /IY/ is not.

The particular configuration of the vocal apparatus is not known for the task of speech recognition, and an utterance must be identified based on the acoustic qualities of the speech signal. The spectrum of the speech signal stores important information about the identity of a speech segments.

In the case of vowels, the spectrum of the speech signal contains a number of distinct peaks or formants. These spectral peaks are the result of resonances produced by the articulators and their position in the frequency domain is closely correlated to the position of

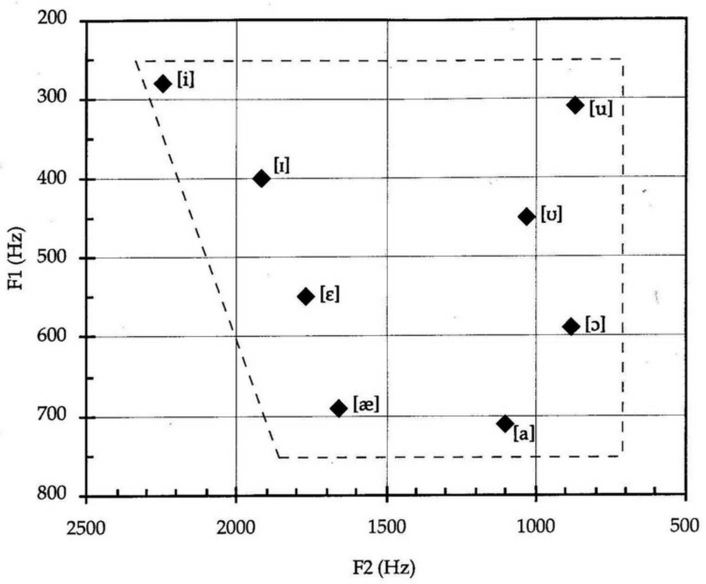


Figure 2: Vowel chart indicating the position of first two formants

the articulators in the vocal tract.

The higher formants are speaker-dependent and vary considerably from speaker to speaker. The first two formants, on the other hand, are a reliable indicator of the quality of a vowel. A vowel chart, illustrating the position of first two formants for eight vowels of North American English, is shown in Figure 2.

The position of a vowel along the vertical axis is determined by the frequency of its first formant while the frequency of the second formant define its position along the horizontal axis. Each of the vowel occupies a distinct region in the vowel chart. Naturally, there would be some variation in the formant frequency of different speakers and this variation would be different for speakers of different dialects, however, the position of the first two formants can be used to identify a given vowel.

1.1.3 Common Sources of Variation in Speech

There are certain variations in speech that do not change its perceptual qualities. Some of these variations, such as the rate of speech and average pitch values, are speaker specific; others are more contextual in nature. For instance, suprasegmental features only affect a few syllables at a time and may include a localized change in stress, length, or intonation.

Context-dependent Variation Speakers frequently raise or lower their voices to compensate for background noise. Sonority describes the relative loudness of a speech segment relative to others and can be used to emphasize a particular syllable.

Speech characteristics also change during propagation due to a loss of energy. This energy dissipation has a increasing frequency profile with higher frequencies components losing more energy than the lower frequencies.

These variations in speech also extend to the frequency domain. Pitch variations are an integral part of speech. A rising or falling pitch convey emotions, emphasize words, and enhance the meaning of an utterance. A subtle change in stress or pitch can dramatically change the meaning of an utterance. Without pitch variations, speech would sound monotonic and unnatural.

Speaker-dependent Variation The average values of pitch and formant frequencies exhibit certain trends across multiple speakers. These differences in the values of fundamental and formant frequencies have their origin in actual physiological differences among speakers.

Across a population, there is a general tendency of male speakers having a lower average fundamental frequency than female speakers. Both the area of the vocal folds and its elastic properties contribute to the natural fundamental frequency of speech. The larynx in an adult male speaker is much larger than a female speaker and contributes to a lower value of the average fundamental frequency.

Similarly, adult male speakers, in general, have longer vocal tracts than either female speakers or children. As a result, the corresponding formant frequencies for female speakers tend to be higher than that of male speakers.

The shift in the formant frequencies across speakers, however, is far from linear. A survey of both American English and Swedish [6] speakers revealed different trends in the ratio of average formant frequencies across speakers of different genders. It was observed that the formant frequencies between a female and a male speaker vary by a different scaling factor for different groups of vowels.

It has been observed that the vowels can be differentiated reliably by the position of the first three formants. Pitch, on the other hand, does not play as important a part in speech recognition. Speech, which has been synthesized from smoothed-out spectrum, has a whispering quality and still is perfectly intelligible. Consequently, the feature sets commonly used in speech recognition, such as Linear Predictive Coefficients (LPC) and Mel-frequency cepstrum coefficients (MFCC), remove pitch information from the speech spectrogram.

Additionally, speech recognition systems often employ speaker adaptation and normalization schemes to either adapt a model to a particular speaker or to normalize the speech features to a trained model. For example, in vocal tract length normalization (VTLN), the frequency axis is warped during feature extraction to account for the shift in the position of the formants.

1.1.4 Manifold Model of Observation Distribution

In a ideal feature set, observations from a single class occupy a distinct, non-overlapping region in the feature space. The samples from different classes in such a feature set can be classified using proximity-based clustering algorithms such as K-means or Gaussian mixture modeling (GMM).

Such well-behaved feature sets are not readily available for every application including speech recognition. Both time and frequency-based representation of speech exhibit huge variability across realizations of a single utterance. Frequency-domain representations of speech are more indicative of the perceptual qualities of speech than time samples, and experts can usually identify phonemes by visually analyzing a spectrogram of speech. However, even the spectrogram representation is not invariant to changes in energy, pitch or formants, and is not robust enough to be used for accurate automatic speech recognition.

Linear Predictive Coefficients (LPC) and Mel-Frequency Cepstral Coefficients (MFCC) are widely used in speech recognition. MFCC representation is robust to certain common sources of variations in speech. For example, a change in energy of the signal will only affect a single cepstral coefficient. Retaining the lowest cepstral coefficients results in a low-time liftering of the speech spectrum and hence cepstral representation is relatively unaffected by pitch variations. The discrete cosine transform (DCT) component of MFCC computation, however, is not shift-invariant and as a result any changes in the position of a formant can significantly alter the MFCC representation of speech. It is not difficult to see that the distribution of a single speech unit, say a vowel, would be in the form of a low-dimensional manifold in the high-dimensional MFCC domain.

A k-dimensional Manifold $\mathcal{M}$ is a topological space that is locally homeomorphic to R^k. Conceptually, a smooth k-dimensional manifold is nearly identical to a k-dimensional hyperplane on a local scale, but this relationship does not hold globally. A simple example would be of a circle, a one-dimensional manifold in R^2. Any small open segment of the circle would be similar to a line in R^1 but globally there does not exist a homeomorphic mapping that would map the entire circle to any open interval on the real line.

1.2 An Overview of Query-by-Example Spoken Term Detection Systems

The state-of-the-art in the field of speech recognition has come a long way over the last five decades. From the earlier systems that would only recognize isolated digits, speech recognition systems have evolved to be a part of our everyday life, becoming the human-machine interface of choice for the latest generation of mobile communication devices and gaining the ability to understand user's intent beyond those expressed by the utterances alone.

Even though, it is not possible to cover fifty years of progress and innovations in a single chapter, it is necessary to present a broad overview of earlier related work. The aim of this overview is to put the current work in perspective both by highlighting the motivations behind some of the ideas adopted in this work and at the same time, to distinguish the innovating features of the current research from those carried out earlier.

The task of a KWS system is to recognize all the occurrences of an utterance in the presence of speaker-dependent as well as contextual variations in speech. Some KWS systems are developed for a particular application domain, limited to a small, predefined vocabulary of utterances. Others may have the ability to process continuous speech and accept any sequence of phonemes as a query.

The form of the input query also varies across different KWS systems. Some systems are designed to accept text representations of keywords. A text query can either be the phonetic transcription of the keyword or it can be an orthographic transcription of a word. The orthographic representations are generally converted into a likely phonetic form with the help of a grapheme-to-phoneme dictionary.

For resource-limited languages with sparse text representations, it is often more convenient to perform keyword search with an actual waveform of an utterance.

Over the years, a number of different approaches have been proposed for keyword spotting. These systems have used several different strategies to handle the temporal and spectral variability in speech.

The variations along the time axis are handled either by the Viterbi algorithm or using algorithms based on dynamic programming. These algorithms find an optimal alignment between the observed frames and a reference model or template.

A number of different techniques have been adopted to handle the spectral variations in speech. The features used in speech recognition systems, such as LPC and MFCC, are nominally invariant to the changes in the fundamental frequency. Statistical modeling techniques use a mixture of Gaussians to represent the non-linear distribution of observations in the signal space. Furthermore, some systems employ speaker adaptation or normalization schemes to modify the model or the feature parameters to suit a particular speaker.

In the next section, a brief overview of template-based and statistical KWS systems is presented with a particular focus on their feasibility for languages with limited linguistic resources.

1.2.1 Unsupervised Approaches to Spoken Term Detection

In template-based approaches, prototypes or templates for a particular utterance are used as a reference signal and candidates are matched against these templates using dynamic programming algorithms.

Different occurrences of an utterance generally are not of the same length or composition. Consequently, the dynamic time warping (DTW) algorithm is used to optimally align a candidate utterance with the referenced one. The accumulated distance between the reference and the test utterances, after optimal alignment, is then used to rank putative hits.

The first generation of the template-based KWS systems were often limited to isolated word recognition. The system developed by Itakura [7] was an example of early single-speaker, isolated-word recognition systems. The developed system used LPC as a feature set and the minimum prediction residual as the distance measure between two samples. An optimal warping path between the test and the reference utterances was calculated using DTW.

A speaker-independent KWS system must also handle the variability introduced by multiple speakers. The speaker-independent speech recognition system proposed in [8] can be considered the precursor of modern statistical keyword modeling schemes. The system gathered statistics for a keyword by clustering 100 exemplars into groups of similar utterances. A matrix of distances between every pair of templates was tabulated and the templates were grouped into a fixed number of clusters such that the ratio of average inter-cluster distance to the average intra-cluster distances was minimized. A particular utterance was then represented by multiple templates, one from each cluster. Recognition involved the calculation of the accumulated distance between the test utterance and each of the reference templates. The average score of the top K comparisons was used for ranking the test utterances.

In an isolated-word recognition system, the boundaries for each word are assumed to be known in advance. Either the utterances are input to the system one at a time or the task of discovering word boundaries is delegated to a separate system. A number of approaches in [9], [10], [11] and [12] addressed the challenges faced in implementing an efficient KWS system for connected speech recognition.

For resource-limited languages, template-based systems offer a number of advantages over systems based on statistical models of speech. In a template-based KWS system, a single instance of a keyword is generally sufficient to initiate keyword search. Statistical techniques, on the other hand, are dependent on the availability of a large set of training data to model each keyword. Statistical techniques that model speech at subword or phoneme-level, require the full set of phonemes to be modeled before initiating the keyword search.

Template-based systems, however, also suffer from a number of disadvantages. The primary limitation of the template-based systems arises from their inability to leverage additional resources for a keyword. These systems rely heavily on particular instances of an utterance and lack an efficient mechanism to generalize patterns across multiple exemplars of a keyword. One approach to leverage the availability of multiple exemplars for a keyword would be to align them and take averages of the corresponding aligned segments. However, when the observation distribution is in the form of a low-dimensional manifold, averages often lie outside the non-linear distribution of the observations.

Some template-based systems utilize multiple prototypes of a keyword, as in [8]. However, the keyword search algorithm is applied repeatedly for each of the prototype, leading to a significant increase in the computational complexity of the systems. Furthermore, the use of multiple prototypes does not guarantee that the full range of variations in the pronunciation of an utterance will be captured. Even more difficult is the task of selecting a set of prototypes that is best representative of a class.

In absence of an explicit mechanism to model observation distributions, template-based systems have limited capabilities to generalize trends in phonetic units of speech and consequently, these systems have lower accuracies than probabilistic KWS techniques.

Template-based systems are also more computationally expensive than the probabilistic approaches to KWS. In DTW, all the frames from the test signal are compared against those from the reference signal. Speech is highly correlated over short intervals of time. A frame-level comparison makes template-based KWS less efficient than the probabilistic approaches where consecutive similar frames are often represented by a single state.

One of the earliest use of acoustic models, trained in an unsupervised manner, for keyword spotting can be traced to the system described in [13]. The process of training acoustic segment models (ASM), in this approach, can be divided into 3 distinct stages: segmentation, segment clustering and the generation of statistical models for each segment cluster. The short-term correlations in speech are captured by the segmentation process which divides the observation sequence into a number segments with similar characteristics. The space of acoustic segments are then reduced into a finite number of clusters. This clustering can be performed by k-means clustering. An HMM for the segments in a single clusters can then be trained using traditional HMM training algorithms. These acoustic segment models define the acoustic lexicon for the development of phone or word models that may also take into account the inter-segment transition characteristics.

Acoustic segment models dominated the field of unsupervised keyword spotting systems for the next decade. These models capture both the spectral and short-term temporal characteristics of speech and lead to much higher precisions than template-based approach. Various methods for the development of word and phoneme lexicon using these ASM have been discussed in [14],[15] and [16]. The statistical-type lexicon building methods are particularly suited for phoneme models and require availability of large amount of labeled data for extraction of the model statistics. Deterministic-type word models, on the other hand, can build a word model with the help of a number of samples of the query and can be used in scenarios where additional linguistic resources might not be available.

Another approach to unsupervised statistical modeling of speech that has gained popularity over the last few years, is using a single GMM universal background model (UBM) of speech [17][18] [19]. A GMM, trained in an unsupervised manner, reduces the feature space into a number of different classes. The posteriorgrams from the Gaussian mixtures can then be used for keyword search using a Segmental DTW algorithm.

In the next section, we present a brief overview of the probabilistic approach to speech recognition and highlight the issues preventing its application to resource-limited languages.

1.2.2 Supervised Statistical Modeling Techniques

In this approach to KWS, a probabilistic model of a unit of speech - either a word, syllable, or phoneme - is trained to capture the contextual and speaker-dependent variations in the acoustic entity using a large set of training data. Hidden Markov models (HMM) are the most popular statistical modeling technique for speech and have dominated the field of speech recognition for the last five decades. The probabilistic model of an utterance is generally represented with a number of states connected in a first-order Markov chain.

Each state in the HMM represents a collection of observations with similar acoustic character. The states in the model are hidden in the sense that the current state can only be estimated from the directly observable speech features. These speech features can either be continuous or may use a finite alphabet size. Most modern systems tend to use continuous value observations. The observation density for each state is modeled by a probability distribution, generally a mixture of Gaussian distributions.

HMMs can efficiently capture the non-linear distribution of observations in the signal space using the emission probability density of the states. A Gaussian distribution with a full covariance matrix is better suited to model complex observation distributions. However, the large number of free parameters in a full covariance matrix significantly increases the amount of training data required to estimate these parameters. Alternately, a mixture of Gaussian distributions, each with a diagonal covariance matrix, can also be used to model these distributions.

A diagonal covariance matrix greatly reduces the number of free parameters to be estimated and thus, reduces the amount of data required for training. The diagonal structure of the covariance matrix is also appropriate considering the coefficients of feature sets, such as MFCC, are purported to be uncorrelated. Furthermore, arbitrarily complex distribution of observation can be modeled with a small number of parameters by incorporating a number of these mixtures.

In addition to observation density modeling, HMMs also provide a statistical framework to model the temporal structure of speech. The temporal dynamics of speech are modeled by a matrix of state transition probabilities. The state transition is a random process in this approach, with the transition to the next state depending on the current state only. An initial state distribution defines the probability of occupancy of each state in the model.

Traditional speech recognition systems model units of speech using a left-right or Bakis HMM model. In this configuration, state transitions take place in a fixed order, starting from the left-most state and moving to the right as time progresses. At each time interval, the current state can either remain unchanged or move to one on the right of the current state. Figure 3 depicts a 3-state left-right HMM.

The sequence of states, however, may not always be strictly in the order shown in Figure 3. In some cases, pronunciations of an utterance may add or omit a phoneme. Such behavior can be modeled by adding additional edges between non-adjacent states and/or deleting an edge between neighboring states.

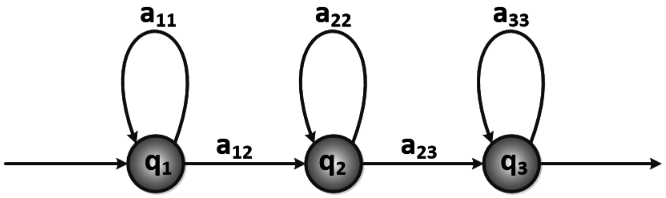


Figure 3: A 3-state left-right HMM.

Mathematically, an N-state hidden Markov model is defined by the parameters (A, B, π).

The initial probability of being in each state is defined in π, a vector of length N . The transition matrix A contains the transition probability between every two states in the model. These states are not directly observable, instead each state has a corresponding distribution of observations B.

The success of HMMs in speech recognition comes from a solid mathematical formulation and the existence of efficient algorithms for estimating the model parameters. These algorithms take advantage of the first-order Markovian assumption to estimate the parameters of the HMM in linear time. The Baum-Welch algorithm [20] estimates the parameters (A, B, π) using an iterative process.

A number of optimization criteria for estimating the parameters of the model have been proposed including maximum likelihood (ML) [21], maximum a posteriori probability (MAP) [22], maximum mutual information (MMI) [23] and minimum discrimination information (MDI) [24]. While the ML and MAP methods estimate the parameters of each model separately, the MMI and MDI methods are suitable for simultaneously training a set of HMMs. In the later two methods, the parameters for each model are estimated such that the optimality criterion is maximized over the complete set of models. The goal of the MMI approach is to find model parameters that maximize the mutual information between the observation sequence and a set of HMMs. The MDI approach estimates the parameters of a set of HMMs by minimizing the cross entropy between the signal probability densities and the set of HMM probability densities. However, these alternate optimization methods, in one way or another, uses the the Baum-Welch algorithm. An excellent resource on HMM in speech recognition can be found in [25].

Word vs. Subword Models of Speech The objects of interest in a KWS system are either words or strings of words in continuous speech. Although building statistical models of speech at word-level has a number of advantages, the focus has been shifted to modeling phonetic or acoustic units of speech in recent years.

Early speech recognition systems were designed for applications with a small vocabulary and generally built separate statistical models for each word in the dictionary. However, a KWS system would require tens of thousands of different models before it could be used for continuous speech. Training and maintaining such a large set of models quickly becomes infeasible and still would not handle any out-of-vocabulary words. Moreover, the addition of a word in the dictionary would require the new statistical model to be trained from scratch.

Modern speech recognition systems generally model speech at subword level units. The total number of phonemes vary from language to language. However, the number of models required for modeling phonemes is much smaller than that required to model whole words.

An advantage of using subword models of speech is that a single set of models can be used for all the utterances spoken in a particular language. Any keyword can immediately be converted into a sequence of phonemes with the help of a grapheme-to-phoneme dictionary. Early examples of such systems are those presented in [26] and [27] where left-right HMMs were trained for phonemes and phone-like units of speech.

A phoneme, on the other hand, exhibits much higher variability than a word. A phoneme may have a different form based on the phonemes immediately preceding or following it. These realizations or allophones are perceptually similar but they may have a significantly different representation in the signal space. Context-dependent models for phonemes are much better at capturing the characteristics of a phoneme in different settings [28].

Phonetic models of speech have been used extensively and successfully in KWS systems. In [29], the authors presented a system for searching keywords in audio archives using phone models of speech. The spoken term detection systems developed in [30] and [31] also circumvented the need for separate keyword models by using existing phone models.

Disadvantages of Statistical Keyword Modeling Schemes The first-order Markov assumption significantly reduces the computational complexity of the training process. However, this assumption is not reflective of the temporal structure that exists in speech.

Phonemes occur in a particular sequence in words and syllables. Human listeners can accurately predict whole words, and even in some cases complete sentences, by listening to a small segment of speech. Correlations in speech extend for much longer period of time than that represented by a single state of the HMM. A first order HMM has the shortcoming that all the prior states have no influence on the current state transitions.

According to the Markov assumption, the state transitions comprise a random process with a fixed likelihood assigned to each state transition. This assumption is generally not true over the life cycle of a phoneme. A phoneme has a rather short but well defined steady state and the likelihood of a state transition increasing as it reaches towards the end of this steady state. This behavior is not accurately modeled with a fixed state transition probability.

However, the most significant factor that limits the application of statistical modeling techniques to resource-limited languages is its dependence on the availability of extensive linguistic resource for training. These approaches require the knowledge of the linguistic composition of a language to identify the basic units of speech. Moreover, the transcription process requires trained linguists to identify and properly label each speech segment in tens or hundreds of hours of continuous speech.

The current paradigm of statistical modeling of speech based on human supervision has been extremely popular and is responsible for the major breakthroughs in the field of speech recognition. However, an excessive reliance on the availability of linguistic resources, makes this approach infeasible for resource-limited languages.

1.2.3 Unsupervised Density Modeling Schemes

Over the years, a number of attempts have been made to minimize or even eliminate the role of transcribed data from the training phase of the statistical modeling of speech. Some of the research. such as [32] and [33], have been focused on training initial acoustic models using a small amount of annotated data and refining these models with the help of unannotated or automatically annotated data.

Other research efforts have been truly unsupervised in nature where the acoustic models of speech are built using unsupervised machine learning algorithms. The objective of these systems is to decompose the signal space of speech into a number of clusters without any human supervision. These systems generally use a proximity-based clustering scheme to group similar observations into a single cluster.

The system developed in [34] is an example of unsupervised clustering of speech into similar acoustic units. Unlabeled speech is segmented using a DTW-based algorithm, polynomial models for the cepstral features in each segment are computed, and a clustering of states is performed using Gaussian mixture modeling (GMM). Using a very limited set of transcribed data, a joint multigram model is then built to map the clusters to the grapheme representation of speech.

In [18], the feature space was decomposed into 50 clusters using GMM. In [35], on the other hand, agglomerative hierarchical clustering of the observations has been used to discover the acoustic units of speech.

These unsupervised approaches to modeling speech use Euclidean distances as a similarity measure between observations. As discussed in Chapter 2, the Euclidean or Mahalanobis distances are not accurate measures of similarity when the observation distribution is in the form of low-dimensional non-linear manifold.

Additionally, a clustering scheme solely based on proximity measures, fails to take advantage of the temporal structure of speech. In the next section, we propose an unsupervised statistical modeling framework for speech that takes into consideration both the proximity and temporal behavior of the observation segments for signal classification.

1.3 Development and Test Databases

A number of different speech corpora have been used both as a training and test data sets for development and evaluations of the system. The development set must contain large number of utterances from a variety of different speakers in order to capture the structure of a particular language. The target languages for an EHMM-based VQbE system are those with scarce linguistic resources. However, labeled data is required to benchmark the system and to facilitate the evaluation of different configurations of the system.

During the course of this research, a number of different speech corpora have be used through the different stages of development. A brief description of each of these corpora will be presented in this section.

The first system was trained on a single-speaker subset of Wall Street Journal (WSJ) corpus [36]. Development was then moved to the TIMIT corpus [37] which offers a larger variety of words and speakers. Experiments on the TIDIGITS corpus[38] were crucial in understanding the effect of different speakers on the EHMM-based representation of speech.

A better estimate of precision of VQbE system can be provided by benchmarks on Fisher corpus [39] of speech contains telephonic conversations between two speakers. Compared to the high quality, read-from-transcript speech in TIMIT, the Fisher data set is sampled at lower frequency. Furthermore, conversations are in a more natural environment, dotted with laughter, pauses and other non-speech artifacts.

The speech corpora discussed so far, are publicly available and provide speech samples along with transcriptions to facilitate training and assessment of a VQbE system. These speech databases also provide a platform to compare and contrast different speech recognition systems. Although, these databases have widely been used for the evaluation of supervised speech recognition systems, they have not been embraced by the unsupervised speech recognition community. Systems, such as those developed in [4], [35] and [18], are based on proprietary databases which are not publicly available.

Recently, there has been new initiatives for introducing a common database for the training and comparison of different zero-resource systems. The Center for Language and Speech Processing at John Hopkins University conducted a workshop[40] where a number of state-of-the-art systems were implemented and tested on a large isolated-word recognition task.

1.3.1 TIDIGITS

TIDIGITS is a database of spoken digits and contains a large number of samples of both isolated and connected versions of digits zero through nine. The utterances are spoken in quiet environment and sampled at 20 KHz. While the database is not linguistically rich, it contains a small number of utterances spoken by a large number of speakers and it is particularly suitable for the study of speaker-dependent variations in speech. There are more than twenty five thousand digit sequences spoken by more than three hundred speakers of different ages, genders and belonging to 22 different dialect groups.

1.3.2 The Wall Street Journal Corpus

The WSJ corpus is a large collection of text and speech selected from 2,499 stories from the articles published in Wall Street Journal. The speech in the database is read from a script in a studio and recorded at a sampling rate of 16 KHz.

The subset of WSJ selected and used in this thesis, contained scripted, continuous speech from a single speaker. The selected subset of the corpus was appropriate for the investigative phase of the VQbE development as it has the necessary linguistic variety while avoiding the problems arising due to multiple speakers or from a noisy environment.

1.3.3 TIMIT Corpus

The TIMIT speech corpus contains speech from 630 different speakers. These speakers belong to eight major dialect groups of North American English. Each file in the database contains read speech from a single speaker and has been recorded at a sampling rate of 16 KHz. The scripts have been especially designed to maximize the linguistic variety in the data set. The TIMIT corpus includes time-aligned phonetic as well as word transcriptions.

Although, an EHMM can be trained using unlabeled speech data, the availability of time-aligned word and phoneme labels in the TIMIT corpus are invaluable for evaluation and refinement of the system.

1.3.4 The Fisher Corpus

The Fisher Corpus contains more than two thousand hours of speech, comprised of more than eleven thousand transcribed telephonic conversations. The conversations are up to 10 minutes long with a particular topic of discussion assigned each conversation to increase the variety of the speech data. For each conversation, the quality of the channel, time-aligned labels, and the age and gender of the speakers have also been recorded.

The Fisher Corpus has been developed in with particular focus on the development of automatic speech recognition systems. Unlike the scripted speech in other corpora, the conversations in this corpus consists of natural speech with cross-talk, non-speech artifacts and more visible dialect variations.

1.3.5 Hausa

In this research, a database of Hausa language has been collected to understand language-specific characteristics of an EHMM. Hausa is the second largest native language of Africa after Swahili and spoken across a large swath of Africa in countries such as Niger, Nigeria, Chad, Benin, Ghana and as far as Sudan. Hausa does not have a native written script and instead Latin or Arabic script is employed as a writing system. As a result, Hausa, though one of the widely spoken languages in Africa, has very limited textual resources for the development of a supervised speech recognition system.

The database collected in this research consists of more than 30 hours of speech and came from a number of radio broadcasts from Voice of America, British Broadcasting Corporation, and Deutsche Welle. The speakers were largely recorded in a studio environment, but a significant fraction of the conversations were held over a telephone line.

The recordings were manually edited to remove segments with background music, speech in other languages as well as poor quality speech. The original broadcasts were recorded at different sampling rates, so the edited speech was then re-sampled at 16 KHz to bring the different recording to a constant sampling rate.

1.3.6 Johns Hopkins University CLSB Collection

The JHU CLSB collection consists of 10825 utterances for which the benchmarks of a number of zero-resource systems are available. It consists of isolated utterances extracted from a multi-speaker speech corpus. The utterances in the collection are all greater than 0.35s in duration and at least six character in length. These utterances belong to 3745 word types. However, some of the word types have only a single example in the word collection.

The word collection has been inspired by the idea initially presented in [41] and aims to rapidly assess the ability of a keyword spotting system to identify words of the same type spoken by different speakers. The paper provided a recipe to select a collection of presegmented word examples drawn from a multi-speaker speech corpus. The JHU CLSP Workshop 2012 created two such word collections from TIMIT and Switchboard Corpora. However, only a fraction of the algorithms tested in the workshop were run on the word collection extracted from Switchboard Corpus. As a result, the word collection derived from TIMIT corpus has been used in this work.

1.3.7 MediaEval 2013 Spoken Web Search Database

The MediaEval Benchmark is an initiative to bring together researcher in the field of multimedia evaluation, to share ideas and compare their work. The MediaEval Spoken Web Search task is defined as “searching FOR audio content WITHIN audio content USING an audio content query.” Every year, a new speech database of resource-limited languages is released along with sets of development and evaluation queries.

The MediaEval 2011 database was collected by IBM Research India and consisted of approximately 3 hours of speech from Hindi, Telugu, and Gujarati languages. The

MediaEval 2012 database consisted of speech from four African languages and contained close to 4 hours of speech from isiNdebele, Siswati, Tshivenda, and Xitsonga languages.

The MediaEval 2013 database is the largest of these databases both in size and linguistic variety. The speech files in the database have been collected from a number of different sources and under different acoustic conditions. The database contains the following 9 languages: Albanian, Basque, Czech, non-native English, Isixhosa, Isizulu, Romanian, Sepedi and Setswana.

The test set consists of 10762 individual files containing approximately 20 hours of data. The development set contains 515 utterances while the number of queries in the evaluation set is 501. Ground truth files were provided for the development set. The queries and the test files have been scrambled, and no information have been provided about the speakers or the language spoken in each file.

CHAPTER 2

AN ERGODIC HIDDEN MARKOV MODEL OF SPEECH

Traditional statistical approaches model speech at word or subword levels using left-right HMMs. This requires knowledge of the linguistic structure of a language and availability of large amounts of labeled data for training. This approach to speech recognition cannot be adapted to a large number of resource-limited languages.

In this research, we propose a probabilistic approach to model speech signals using a single, ergodic hidden Markov model. While the phonemes always occur in a particular sequence in an utterance, the set of phonemes, in different combinations, form all the utterances in a language. Modeling the entire signal space of a language would require a model with a far more flexible structure than that of a left-right HMM.

EHMMs offer a good mix of straightforward training with the generalization power of statistical modeling techniques. An EHMM provides a statistical framework to efficiently capture the range of variations in speech which is absent in most template-based approaches. By building a single model of speech, this approach eliminates the reliance on time-aligned labeled data for training. Furthermore, the training process makes efficient use of the available data by using every frame for the parameter estimation of the EHMM.

The next section begins with an introduction to EHMMs, and continues with a discussion of different parameters governing the design of an EHMM of speech. The optimal number of states in an EHMM , the model of the observation distribution, and the training procedures will be discussed. The chapter will be concluded with a presentation of the spectral and temporal characteristics of EHMM of speech for a number of languages.

Definition A hidden Markov model λ is a collection of states connected in the form of a Markov chain. The model is completely defined by the parameters

$$\lambda = \{A, B, \pi\} \tag{1}$$

There are N states in the model but only a single state, denoted by q, can be occupied at any given time. The next state is chosen randomly at each time increment. The probabilities of transition between the states of the HMM are defined in the state transition matrix A. In a first-order Markov process, the probability of transition to state j depends only on the current state q. Consequently, the state transition matrix A is of size $N \times N$ with the element a_{ij} denoting the transition probability from state i to state j.

The states in the model are hidden in nature and the identity of the current state can only be inferred from a set of observations. Each state in the model has an associated probability distribution for observations defined in B. The observation density is often modeled as a mixture of Gaussian distributions. The initial state distribution is defined in π where $\pi_i = \Pr(q = s_i)$.

In an ergodic HMM, each state in the model can transition to any other state in a finite number of steps. Mathematically, a state in a Markov chain is recurrent if there is a non-zero probability that it will be visited. A state is aperiodic if the greatest common divisor of the number of time steps it takes to return to that state is 1. A Markov chain is irreducible if every state can be reached from any other state in a finite number of steps. A Markov chain is ergodic if it is irreducible and if every state in the model is aperiodic and recurrent.

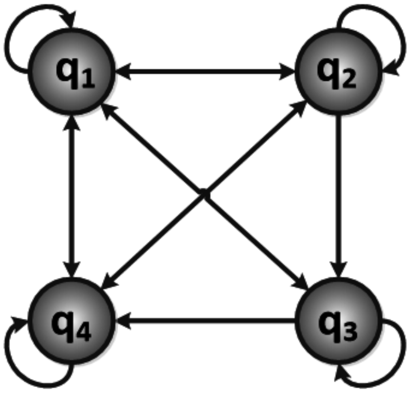


Figure 4: A 4-state ergodic HMM.

An example of an EHMM with 4 states is shown in Figure 4. The EHMM depicted in the figure is fully-connected, with edges connecting any two states in the model. An

EHMM of a language, on the other hand, would consist of a considerably larger number of states. Furthermore, due to the high short-term correlations in speech, the connectivity between the states of the EHMM is generally sparse. The design of an EHMM of speech is discussed in detail in the following section.

2.1 An EHMM of speech

The capability to capture non-linear observation distributions and the flexibility imparted by the ergodicity assumption allow an EHMM to model complex structures. In this research, the entire feature space of a language is modeled by single EHMM. A mixture of Gaussian distributions is generally used to approximate the observation distribution in a state and the short-term, temporal correlations in speech can be captured using the state transition matrix.

The design of an EHMM encompasses a number of factors including the size of the EHMM, the form of the observation distribution, initial values of the parameters, and the training methodology. During preliminary investigations, a number of EHMMs with different configurations were trained on a number of languages, with an aim to find the optimal configuration for an EHMM.

2.1.1 Number of States in an EHMM

An EHMM decomposes the feature space of speech into a number of different states, each with different observation density and state transition characteristics. The number of states in an EHMM determines the level of detail that can be captured by the model.

For the task of keyword spotting, the number of states in an ideal EHMM should be equal to the number of perceptual units of speech. In such a EHMM, a single state would represent the complete range of observations within a single phonemic class of speech. In such an ideal EHMM, different occurrence of a keyword would always correspond to a singe state sequence of the EHMM. However, such a mapping between a phoneme and state of the EHMM is not possible due to the unsupervised nature of the training.

Table 1: Average log-likelihood per frame for different EHMMs

Number of States	64	128	192	256
Average lg $P(O\|\lambda)$ per frame	-6.72	-6.61	-6.57	-6.54

Alternately, speech can be modeled with an EHMM with a very large number of states. In this approach, each state would consist of tightly clustered set of observations, with a high likelihood that the observations within a single state would also belong to the same perceptual class of speech. However, this approach does not guarantee that a single perceptual class of speech would be modeled by a single state of the EHMM, and would require a mechanism to identify and group the states with identical perceptual qualities. Such an EHMM would also be able to record a much finer level of detail about the temporal characteristics of observation sequences.

EHMMs have been previously used to model speech in a number of applications. In [3], it was reported that an EHMM with 64 states can store sufficient detail to synthesize intelligible speech from the sequence of states of an EHMM with very low entropy. Increasing the number of states from 64 to 128 resulted in a noticeable increase in the perceived quality of the synthesized speech. However, any further increase in the number of states resulted in marginal improvements in the quality and intelligibility of the synthesized speech.

In this research, a number of EHMMs with different number of states were trained for a number of languages. An EHMM models the speech generation mechanism. A measure of the success of a model in explaining an observation sequence is provided by $\Pr(O|\lambda)$, i.e., the probability of observation given the model.

The implementation of the Baum-Welch algorithm in HTK provides a scaled version of $\Pr(O|\lambda)$ in the form of $\log \Pr(O|\lambda)$ per utterance, at the completion of each iteration of the algorithm. Table 1 lists the average log-likelihood per frame for a number of EHMMs, after the same number of iterations of the training algorithm.

It can be seen in Table 1, the average log-likelihood of an utterances increases consistently with the number of states in the EHMM. The values in Table 1 indicate that larger

EHMMs are much better generative models of speech and the number of the states in an EHMM should be as large as possible within the allowed computation complexity of the system.

For a fixed number of phonemes in a language, increasing the number of states in the EHMM has an undesirable effect of fragmenting the observation distribution of a single perceptual class into a number of states of EHMM. This one-to-many mapping between a phoneme and the states of an EHMM can significantly complicate the task of keyword spotting and will be discussed in detail in Chapter 4. Hence, the number of states that leads to the highest $\Pr(O|\lambda)$ for a model may not be optimal for the task for keyword spotting.

Furthermore, the number of states in an EHMM cannot be increased indefinitely. In the above experiments, if the number of states were increased beyond a certain limit, then some of those states started to become redundant after a few iterations of the Baum-Welch algorithm. There were no transitions into or out of these states, excluding these states from modeling any observation in the training data.

For the EHMMs trained on the TIMIT corpus, it was observed that for EHMMs with total number of states greater than 200, some of the states in the model would become redundant in the later iterations of the training algorithm. This redundancy problem, however, was not observed for EHMMs with less than 200 states. For larger EHMMs, particularly with total number of states equal or greater than 512, a large number of states become redundant after just a few iterations of the Baum-Welch algorithm.

This behavior is not unexpected as the number of phonemes, and hence the acoustic variety, in a language is fixed and would require only a certain number of states to model them. The total number of states in such EHMMs is much larger than the number of phonemes and the training data can be modeled successfully with only a subset of the total states.

The maximum number of states that can be used without running into the redundancy problem, may also be dependent on the training procedure. In this research, a flat-start strategy has been used with the states initialized to values close to the global mean and variance. Alternate approaches may include training a number of EHMMs for different speakers or acoustic environments and merging them later into a single large EHMM of speech.

The optimal number of states in an EHMM is a function of the acoustic variety in the training data and may differ from one training set to another. Consequently, the number of states required to model the single-speaker speech in WSJ corpus may not be the same as those required for TIMIT corpus due to difference in speaker-dependent variations. Similarly, different languages generally have different set of phonemes and consequently, a multi-speaker training set in Hausa may require a different number of states to model than a comparable training set in English due to differences in acoustic variety between the languages.

2.1.2 Modeling the Observation Density

In continuous density HMM-based systems, the distribution of observations in a single state has historically been modeled using a mixture of Gaussian distributions. Any finite continuous-density observation distribution can be modeled with sufficiently large number of Gaussian mixtures. Each mixture is defined by a mean value and a covariance matrix. The contribution of each Gaussian mixture to the distribution is controlled using a weight factor.

The emission probability $b_j(o_t)$ can then be calculated as

$$b_j(o_t) = \sum_{i=1}^{M} c_{ji}\mathcal{N}(o_t; \mu_{ji}, U_{jm}), \tag{2}$$

where the number of mixtures for j^{th} state is M, c_{ji} is the weight of i^{th} mixture, while the mean vector and covariance matrix of i^{th} Gaussian mixture is given by μ_{ji} and U_{ji} respectively. The mixture weights c_{ji} for the j^{th} state must satisfy the conditions

$$c_{ji} \geq 0, \tag{3}$$

$$\sum_{i=1}^{M} c_{ji} = 1, \quad 1 \leq i \leq M. \tag{4}$$

Traditional HMM-based systems model speech either at word or sub-word level. Separate statistical models are trained for each of these units of speech using left-right HMMs.

The transcription process isolates all the examples of a particular class from the training data. There must be adequate variety in these examples to capture the full range of variations within a class of observations. Hence, the transcription process, taking advantage of human perception, classifies a wide range of potentially very different samples of a class and collects them under a single label. Due to the large variations in these samples, a mixture of Gaussians is required to approximate their observation density.

The left-right HMMs have a rigid structure with states occurring in a particular sequence. Although, left-right HMMs may allow certain states to be skipped to account for differences in the pronunciation of an utterance, however, an observation frame at the beginning of a training utterance will more likely influence a state at the beginning of the model and vice versa. Thus, the left-right structure of the HMM limits the range of influence an observation may have on the states during the estimation process.

The mixture parameters for each state in the HMM are estimated using multiple iterations of the Baum-Welch algorithm. During the parameter estimation, observations with a high probability of being emitted from a particular mixture would have a larger influence on the estimates of that Gaussian mixture. Hence, the weight an observation would have on the estimates of a mixture is inversely proportional to the Mahalanobis distance of an observation from the mean value of the mixture.

To summarize: the estimation of the HMM parameters is shaped and influenced by three separated mechanisms in supervised speech recognition systems.

1. Observations that are similar perceptually yet potentially very different from each other in the feature space, are grouped together under a single label by a human listener.

2. The influence these observations may have on the states of the HMM is controlled by the state transition matrix.

3. The estimates of mixture parameters are influenced by those observations that are in close proximity of those mixtures.

In the current approach, a single, large, ergodic HMM models the entire feature space of a language. In doing so, the system eliminates the dependence on the availability of time-aligned labeled data and the linguistic knowledge of these labels for training. However, this elimination comes at a cost since the transcription process is a very effective speaker-invariant classification scheme that brings together potentially different observation sequences under the same label.

Furthermore, the state connectivity is much more flexible in an EHMM with every state connected to another through a finite number of state transitions. This flexible structure of the EHMM, together with an absence of labeled data, leaves the proximity of observations from a Gaussian mixture as the primary force that shapes the parameter estimates.

The observation distribution within a single perceptual class of speech is highly nonlinear. As a result, some of these observations may occupy distinct regions in the feature space. The current approach, based on EHMM of speech, lacks a mechanism to bring together these acoustically distinct yet perceptually similar observations within a single state of the EHMM.

The focus of the training phase in the current research is to collect observations based on their proximity and represent them with a single Gaussian mixture. As a consequence, the observation distribution of a single perceptual class of speech could be represented by more than one state of the EHMM. Such a one-to-many mapping between a phoneme and the states of the EHMM can significantly degrade the precision of a system based on this approach. The identification of perceptually similar states of the EHMM will be the focus of Chapter 4.

2.1.3 Training

One of the attractive features of an EHMM is that it does not require time-aligned transcribed speech for training. Since a single EHMM models the entire speech, all the training data is considered to be of a single class and has a single label. Unannotated speech data is available for a large number of languages and can be readily gathered from a large variety of sources including television and radio broadcasts, podcasts, and streaming videos.

The training involves the established Baum-Welch algorithm and can be carried out using the Hidden Markov Model Toolkit (HTK) [42]. In this work, 39-dimensional Mel-frequency cepstral coefficients (MFCC) with delta and delta-delta coefficients were used as a feature set. The MFCC features were generated at a frame rate of 100 using 25 ms windows.

In the initial HMM definition, the states were fully-connected with equal transition probability between two different states. To account for the high level of temporal correlation present in speech, the initial self-transition probability was set to 0.7. The parameters of the model were estimated using multiple iterations of the Baum-Welch algorithm [20].

The distribution of observations for a single phoneme is highly non-linear. For this reason, the distribution of observation for a state is generally represented with a mixture of Gaussians in traditional HMM-based approaches. In the current approach, the primary motivation is to use a large number of states such that a single state consists of observations which are tightly clustered in the feature space and which can be modeled with a single Gaussian distribution. Hence, the observation density for each state was modeled with a single multivariate Gaussian distribution with a diagonal covariance matrix. The mean value and the covariance of every state was set to the global mean and covariance of the training data using the executable HCOMPV in the HTK. Random perturbations were added to the parameters of each state using a single iteration of the program, HINIT, which updates the mean and covariance of the states using uniform segmentation of the training data. The HEREST tool in the HTK was used to perform the iterations of the Baum-Welch algorithm.

The first few iterations of the Baum-Welch algorithm have the largest gains in the probability of the training data given the model $\Pr(O|\lambda)$. The incremental gain in $\Pr(O|\lambda)$ reduces significantly after 10 to 15 iterations of the algorithm at which point the training can be stopped.

In the next section, the acoustic properties of observations within a single state of an EHMM are presented.

2.2 Spectral Characteristics of States

The states of the EHMM were initialized randomly around the global mean value of the speech signal. After training, however, the states represented collections of observations having similar spectral characteristics. The states presented in this section, unless specified otherwise, are from a 256-state EHMM of North American English trained on the TIMIT corpus [37].

The observation distribution in each state can be visualized by transforming the mean value of a state from MFCC domain into linear frequency domain. The computation of MFCC is a lossy process due to the logarithmic quantization of the signal amplitude and non-linear sampling along the frequency axis. Consequently, the inverse transformation cannot exactly recover the mean value in the linear frequency domain and is an approximation.

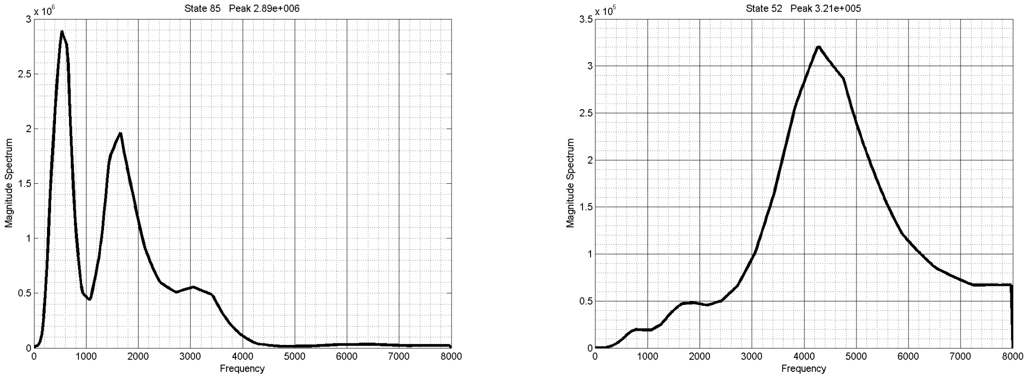
The Gaussian distributions, when transformed from MFCC to linear frequency domain, are revealed to have converged to values representing different phonetic classes of speech including vowels, consonants, and even pauses and silent periods in the training data.

Phonemes are the basic units of a spoken language just as the alphabets are the units in the writing system of a language,. Phonemes have unique acoustic characteristics owing to their particular speech generation mechanisms. As discussed in Chapter 1, vowels are characterized by the presence of formants and most of the energy of the signal is concentrated into these formants. Stop consonants, on the other hand, have noise-like characteristics and a wider bandwidth due to the perturbation produced by sudden release of air through the articulators.

In the case of the EHMM of North American English training using the TIMIT database, 110 of the 256 states have the characteristics of a vowel characterized by the presence of two prominent formants. Another 31 states can be categorized as semi-vowels or glides based on their spectral characteristics. The other 115 states collectively modeled the consonants including nasals, sibilants, fricatives, plosives as well as silent parts of speech. The silent states consist of both stop consonants, as well as pauses and silences between the utterances.

The observation distribution from a state can be represented by its mean value which is also the most likely observation to occur in a Gaussian-distributed observation. In order to visualize the characteristics of observations from a single state, it is necessary to transform the mean value from MFCC domain to linear frequency domain. The mean values, in the linear frequency domain, for two states of an EHMM are shown in Figure 5.

The vowel can be identified by calculating its formant frequencies and consulting the vowel chart. The first state in the Figure has its first formant close to 600 Hz while the frequency of the second formant is approximately 1600 Hz, making it the vowel /EH/. The second plot in Figure 5 does not have the characteristics formant structure of a vowel. The presence of high frequency components is characteristic of fricatives. Fricatives can either be voiced or unvoiced. Voiceless fricative such as /F/, /SH/ and /S/ can be distinguished from voiced fricative such as /V/, /ZH/, /Z/ based on the absence of harmonic structure in the spectrum. However, due to the liftering involved in the computation of MFCC, it is not possible to differentiate between the two categories using the mean values of a Gaussian



(a) *(a)* **(b)** *(b)*

Figure 5: Mean values of states of the EHMM corresponding to a vowel (a) and a sibilant (b).

mixture. However, the pitch of speech segments corresponding to a given state can be estimated separately. The estimates of the pitch values revealed that the corresponding speech segments were predominantly unvoiced, and hence the state is a voiceless fricative. The level of energy concentrated at high frequencies indicate a sibilant such as /S/, /SH/ which has higher energies compared to fricatives such as /F/ and /TH/. The energy of the sibilant /S/ is concentrated at frequencies higher than 5000 Hz while for /SH/ it is concentrated at lower frequencies.

2.2.1 An EHMM of Hausa

Hausa is a Chadic language spoken in West Africa. It is spoken by more than 52 million people and is one of the largest spoken African languages. Hausa does not have a native writing systems. Instead, it is written in a number of different writing systems in different regions of Africa. Hausa is written in Boko, a Latin script, in former British-occupied areas while it is written in Ajami, using the Arabic alphabets, in the rest of the continent.

Hausa is an ideal candidate for an unsupervised keyword spotting system since linguistic resources required for training supervised statistical models are scarce. In this research, an EHMM of Hausa was trained using 30 hours of radio broadcasts from sources including British Broadcasting Corporation, Deutsche Welle and Voice of America.

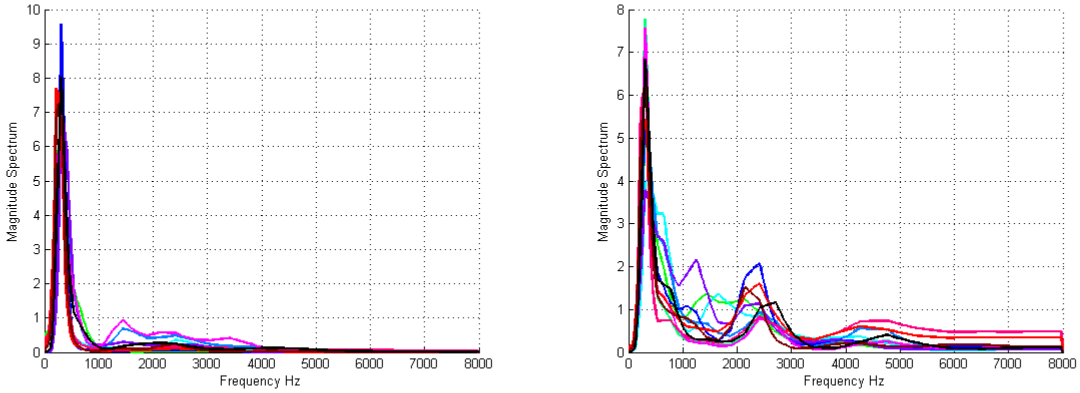
A detailed phonetic analysis of Hausa is beyond the scope of this thesis and very difficult in the absence of sufficient linguistic resources. However, a basic investigation of the states of the Hausa EHMM revealed some interesting information about the language. The mean values of observation distribution for each state were transformed to linear frequency domain and their spectral shape used for classification.

Hausa is a tonal language. The five basic vowels each have three variants with low tone, high tone and falling tone. This tone information is not encoded in standard writing systems. Moreover, each of the five Monophthongs has a long and a short version. Additionally, there are four Diphthongs, which are vowels which change quality during articulations.

A comparison of a 256-state EHMM of Hausa with an EHMM of North American English of similar size reveal some major differences. An analysis of the mean value of the states of the Hausa EHMM show that the number of states that can be categorized as vowels is 125, which is larger than the 110 for the TIMIT EHMM. Hence, a larger fraction of states in EHMM have been used to model the sounds of vowels in Hausa than for North American English even if the TIMIT corpus uses 20 different labels for vowels of North American English.

It is not just the number of vowels that is different between the two models but the spectral characteristics of some of the states also differ. A comparison of states of EHMM normally associated with the nasal /N/ in Hausa illustrated clear differences with those in North American English. For this study, 50 speech segments containing the phoneme /N/ were extracted and the ten most frequently occurring states were selected. The plots of spectrum of these states are shown in Figure 6a. For comparison, Figure 6b shows the plots of states for North American English.

It can be seen from Figure 6 that, for Hausa, most of the signal energy is concentrated in spectral peaks centered at a frequency range of 250-350 Hz. In contrast, the states for North America English have energy distributed over a wider frequency range and among a



(a) *Hausa* **(b)** *North American English*

Figure 6: Comparison of nasal consonant /N/ in Hausa and North American English

number of spectral peaks. This example demonstrates that an EHMM can effectively learn the structure of a language being modeled and groups together observation with similar spectral properties into a single state.

2.2.2 Classification of States based on Spectral Characteristics

Most of the states in an EHMM can easily be placed in a class of phonemes such as vowels or consonants based on the spectral characteristics of a state. The case of the vowels is easier of the two classes as the quality of a vowel can be determined reliably based on the position of the formants. However, classification is more complicated in the case of consonants.

Stop consonants, for example, are all characterized by a complete closure of the vocal tract, followed by a sudden release of air. During this closure, there are no distinguishing spectral features that can be used for classification. Once the stop is released, the presence or absence of a voice bar, the voice onset time, and the evolution of the formant frequencies can be used to distinguish between these consonants. For stop consonants at the beginning of an utterance, the effect of the position of an articulators, and hence the identity of the consonant, becomes apparent at the onset of the following vowel. Similarly, the movement of formants in the preceding vowels contains important clues to the identity of consonant terminating an utterance.

For most consonants, the spectral shape of a speech segment is often not enough to establish its identity. Any static classification scheme that ignores the time evolution of an observation and considers only the current state of observation is not expected to distinguish between such phonemes.

In the case of HMMs, however, the estimation process also takes into account the temporal characteristics of training observations in addition to spectral similarity, as discussed in detail in the next section.

2.3 Capturing the Temporal Characteristics of Speech

The idea of modeling the space of speech signal using a number of Gaussians is not unique to an EHMM. Gaussian Mixture Models (GMM), for example, are easy to implement and efficient to execute. GMMs have been successfully used to model speech and the Gaussian posteriorgrams were used for keyword spotting [18]. In contrast to GMMs, an EHMM is not a static clustering scheme and the estimation process takes into account both the proximity of the observation sequence in the feature space as well as their temporal characteristics.

The parameters of an EHMM are estimated using the Baum-Welch algorithm. This algorithm, which generally is considered a special case of the Expectation-Maximization (EM) algorithm, is iterative in nature. The observation likelihoods are calculated from an initial set of model parameters and these likelihoods are, in turn, used to improve the estimate of the model parameters. At each step of the algorithm, the mean value μ_i and variance U_i for a state i are estimated as given in the following equations.

$$\mu_i = \frac{\sum_{t=1}^{T} \gamma_t(i) o_t}{\sum_{t=1}^{T} \gamma_t(i)}, \tag{5}$$

$$U_i = \frac{\sum_{t=1}^{T} \gamma_t(i).(o_t - \mu_i)(o_t - \mu_i)'}{\sum_{t=1}^{T} \gamma_t(i)}. \tag{6}$$

The variable $\gamma_t(i)$ determines the influence of each observation on the estimates and is equal to the probability of being in state i at time t given the current model and the observation sequence $P(q_i|O, \lambda)$.

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{k=1}^{N} \alpha_t(k)\beta_t(k)}. \tag{7}$$

The terms $\alpha_t(i)$ and $\beta_i(i)$ are calculated using the forward-backward algorithm.

$$\alpha_t(i) = \left[\sum_{k=1}^{N} \alpha_{t-1}(k)a_{ki}\right] b_i(o_t), \tag{8}$$

$$\beta_t(i) = \sum_{k=1}^{N} a_{ik}b_k(o_{t+1})\beta_{t+1}(k). \tag{9}$$

The calculation of the α and β variables during the forward-backward algorithm depend both on the values of the emission probabilities and the state transition probabilities. The impact an observation has on the estimates of μ_i, U_i, and the transition matrix A is directly dependent on the value of transition probabilities a_{ij} and the emission probabilities $b_k(o_t)$. Consequently, a state in the EHMM represents a collection of observations of similar form and temporal characteristics.

An EHMM of North American English, trained on the training subset of TIMIT corpus is shown in Figure 7. The EHMM consists of 256 states, each with a single Gaussian mixture with diagonal covariance matrix. Although, the states were fully-connected before training, it can be seen from Figure 7, that the connection between the states of the trained EHMM are sparse. Given the current state, the next state is much easier to predict in the trained EHMM. This fact can be precisely measured with the entropy rate of the EHMM.

The entropy rate of the EHMM is defined as

$$H = -\sum_{i=1}^{N}\sum_{j=1}^{N} \pi_i a_{ij} \log_2 a_{ij}. \tag{10}$$

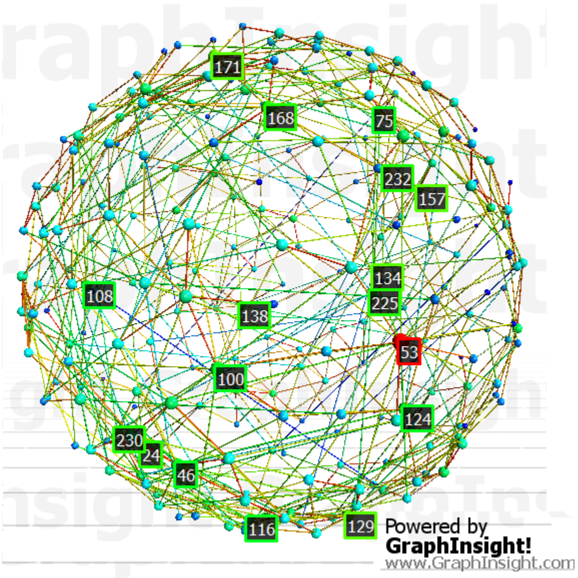


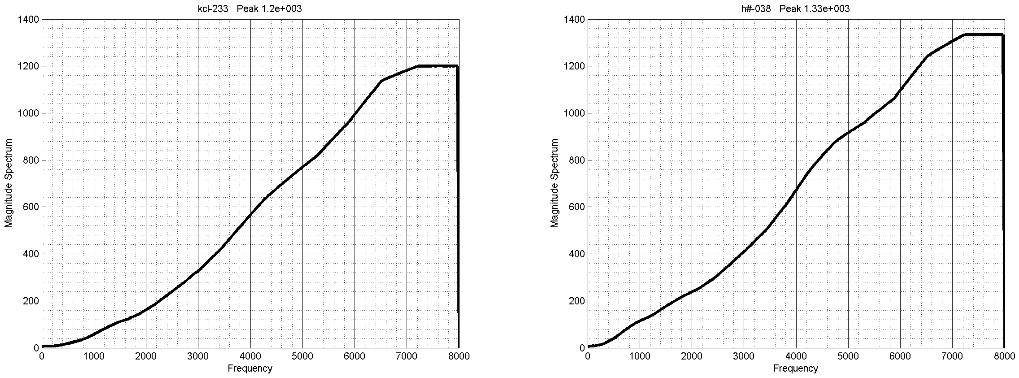
Figure 7: An ergodic HMM of English language trained on the TIMIT corpus

For a 256-state EHMM where all the states transitions are equiprobable, the entropy rate is equal to 8 bits. Another way to interpret this number is that given the current state, the next state can be any one of the 256 states of the EHMM and it would require 8 bits information to encode the next state. However, in this particular case, the next state is much easier to predict and the calculated entropy rate for the trained EHMM is 2.1383 bits. Thus, if the current state of the EHMM is known, then the next state can be optimally encoded with just 2.1283 bits of information. Another way to interpret this information is that a single state of the EHMM is connected, on average, to less than five other states indicating the sparse connectivity between the states. This measure indicates that the trained EHMM efficiently captured the temporal structure in speech.

The self-transition probability a_{ii} is related to the expected duration $\overline{d}$ of a state by the following equation [25].

$$\bar{d} = \frac{1}{1 - a_{ii}}. \tag{11}$$

The initial self-transition probability of 0.7 corresponds to a conservative estimate of 3 frames or 30 ms for the average duration of a state. In contrast to the initial EHMM, there



(a) *Spectrum of /KCL/*

(b) *Spectrum of State corresponding to a silence*

Figure 8: Spectrum of two states with different average durations

are large variations in the value of self-transition probabilities in the final EHMM.

Different phonemes have different average durations due to the physical mechanism underlying their generation. Vowels are easier to sustain due to their voiced nature. Consonants, on the other hand, are shorter in duration since the glottis is in an open configuration and the sound is produced by the rushing air passing through a constriction in the vocal cavity. As a consequence, vowels tend to be longer in duration and exhibit a larger variability in their length than stop consonants.

The average durations of the states in the EHMM follow similar trends. The self transition probabilities ranged from 0.1383 to 0.9227 with a mean value of 0.5955. The states with the longest durations, unsurprisingly, correspond to the silence that occurred at the beginning and the end of the sentences. Vowels, in general, have much larger self-transition probabilities, and hence, longer average durations than consonants.

As discussed in the last section, the spectrum of a speech segment is often not sufficient to distinguish between certain consonants. For example, the stop consonants are all characterized by a complete closure of an articulators. For these speech segments, classification must be based on a time evolution of the observations.

It can be seen from Eq. 8 and 9 that the parameter estimates are influenced both by the emission probabilities as well as the state transition probabilities. The states in Figure 8 demonstrate the fact that the temporal characteristics of observations indeed have a role in their state assignment. The states shown in Figure 8 are indistinguishable from each other in the frequency domain but have significantly different state transitions probabilities. The self-transition probability for state in Figure 8a is 0.362, almost half of that for state in Figure 8b which is 0.698. An analysis of state occupancy for the state in Figure 8b shows that the state almost always occurs (95.6%) when the speech segment is labeled as silence or /H#/. The state in 8a, on the other hand, has a probability of just 3.7% of having the same label /H#/ and the highest likelihood of being part of the stop consonant /KCL/. Hence, the observations have been differentiated successfully due to the difference exhibited in the time domain and have been represented by different states even when they have the same spectral characteristics.

CHAPTER 3

KEYWORD SPOTTING USING AN EHMM OF SPEECH

An EHMM models speech with a number of states, each containing observations having similar spectral and temporal characteristics. It has been successfully demonstrated previously that an EHMM stores enough information about speech that perfectly intelligible speech can be synthesized from a sequence of states of EHMM.

In this research, the role of an EHMM of speech for the task of keyword spotting has been investigated. This approach is particularly suited for languages with scarce linguistic resources. The Voice-Query-by-Example system, envisioned in this research, accepts a query in the form of speech waveform and retrieves all the occurrences of the query in a possible large set of speech data.

The EHMM of speech can be used to transform speech into a model-specific representation which can then be used by a VQbE system. In addition to a form of representation, the VQbE system would also require a framework for keyword modeling and efficient keyword search algorithms that can use these models to find candidate utterances in the test data.

In this research, we investigated two completely different approaches to keyword search using the trained EHMM. The first approach is template-based in nature, and converts a keyword into a posteriorgram of the states of the EHMM and uses this posteriogram as a template to find the matching instances of the keyword. The second approach generates a left-right HMM from the most likely sequence of states corresponding to a keyword and computes the likelihood of each candidate utterance using the Viterbi algorithm. The two approaches are presented in detail in the following sections.

3.1 Dynamic Time Warping

In the first approach, the VQbE system is template-based in nature and uses the DTW algorithm to find an optimal alignment between the test and the reference signals. In contrast to the traditional template-based KWS systems, an utterance is represented with the posterior probabilities of the EHMM states given the observation sequence. The states of the EHMM, trained on a large set of data, convey much more information about a speech segment than the traditional representations of speech obtained through a fixed transformation of the speech samples.

One way to represent speech in terms of the states of an EHMM is to use the most probable sequence of states of the EHMM corresponding to a speech segment. In this approach, two speech segments can be compared by calculating the edit distance between the corresponding sequences of states while accounting for addition, deletion, or substitution of states between the test and reference sequences. However, such a keyword search scheme would be a form of hard-decision decoding and ignores the information contained in N-1 states at each instant of time.

In this research, an alternate method has been adopted that takes into account the probabilities from all the states of the EHMM. A speech segment is transformed into State Posteriogram Representation (SPR) where an input speech sequence $\mathbf{O}$ of length T is converted into a $N \times T$ matrix of state posterior probabilities $P(q_i|\mathbf{O})$ for an EHMM with N states.

SPR is a EHMM-specific representation that is tied to a particular EHMM. In this representation, each column is a vector of posterior probabilities of the states at that instant of time and these probabilities are computed taking into account the complete observation sequence $\mathbf{O}$ using the forward-backward algorithm. It should be noted that the value of $P(q|\mathbf{O})$ depends both on the value of the observations and the corresponding state transitions.

Once the test and reference sequences have been converted into SPR, a similarity measure or distance measure must be defined between two posteriograms. Since the test and reference sequences are generally of different length, an alignment strategy must also be devised to align the similar segments of the two sequences in an optimal manner.

There are a number of different similarity and distance measures that have been defined for comparing probability density functions. The Minkowski family of distance measures includes the L_p norms in R^p. Other examples include the intersection family, the inner product family, the square-chord family and the Shannon entropy family. For a comprehensive survey of similarity and distance measures between probability density functions, see [43].

The inner product is one of the most computationally efficient similarity measure among those defined in [43] and has been used in this research. A lattice of size $T_r \times T_p$ of inner products is generated between the elements of the reference and the test signals with lengths T_r and T_t respectively. An optimal path, connecting the lower left corner to the upper right corner in this lattice, which maximizes the sum of the inner products is computed using the DTW algorithm. This calculated sum can then be used as a similarity measure between the test and reference signals.

Keyword spotting in this approach would involve taking a keyword-length sized window of observations and applying DTW to find the similarity score of the observation sequence. The window can then be moved forward by a fixed number of frames. Moving the window by a single frame yields the best alignment between the window and a matching utterance but it is the most expensive in terms of computations involved. On the other hand, if the window is advanced by n frames at a time, then the computation cost is reduced by a factor of n while the maximum possible misalignment between a window and a matching candidates would remain under $n/2$ frames. A local filtering of the similarity score is often required to remove the duplicate hits generated by a partial overlap of the window with a matching candidate. Finally, the candidates can be ranked by the descending order of their similarity score. A fixed number of the top candidates are then evaluated by comparing the associated transcript of each candidate with the label of the keyword.

3.2 The Viterbi Algorithm

Alternately, an utterance can be modeled as a left-right HMM with the model parameters derived from the states of the EHMM. The most likely state sequence for an utterance can be calculated by the standard Viterbi algorithm using the parameters of the EHMM of speech. This state sequence can then be converted into a left-right HMM by merging multiple consecutive frames which are part of the same state together to form a single state in the keyword model. A similar scheme has also been adopted by [4] where they cited the EHMM of speech developed in [3] as the motivation for their system.

A keyword is represented by a sequence of states $\mathbf{q} = (q_0, q_1, q_2, \ldots, q_M)$, together with the state transition probabilities and observations densities as defined in the EHMM. The states in $\mathbf{q}$ become part of the keyword model while the remaining states of the EHMM become part of the filler or background model.

For keyword search, an observation sequence $\mathbf{O} = (o_1, o_2, o_3, \ldots, o_T)$ is first converted into State Likelihood Representation (SLR). In this model-specific representation, speech is represented with a matrix $\mathbf{R}$ of size $M \times T$ represents where each column is a vector of observation likelihoods given each state i of the keyword model. Specifically, the element $R_{i,j}$ denotes the probability $\Pr(o_j \mid q = s_i)$.

The HMM model of the keyword has M states while there are T observations. Now there can be a large number of state sequences $Q = (q(1), q(2), \ldots, q(T))$ such that the state assignment follow the sequence of states in $\mathbf{q}$, i.e. $q(1) = s_0$, $q(T) = s_M$.

The likelihood of a given observation $P(O \mid \mathbf{q})$ can be computed as the sum of $P(O \mid Q)$ over all the possible ways a M-state sequence can be mapped to T-length observation sequence.

$$\Pr(O \mid \mathbf{q}) = \sum_{all\ Q} \Pr(O \mid Q) \Pr(Q \mid \mathbf{q}). \tag{12}$$

However, for continuous speech recognition, it is more practical to calculate the maximum over the posterior probability $\Pr(Q \mid O)$. Since $\Pr(O \mid \mathbf{q})$ is not involved in the optimization with respect to state sequence Q, we can instead maximize $\Pr(Q, O \mid \mathbf{q})$.

$$\max_{all\ Q} \Pr(O, Q \mid \mathbf{q}) = \max_{all\ Q} \Pr(O \mid Q, \mathbf{q}) \Pr(Q \mid \mathbf{q}). \tag{13}$$

which can be calculated recursively [25] as

$$\max(\Pr(O, Q \mid \mathbf{q})) = \delta_T(M) = [\max_i \delta_{T-1}(i) a_{iM}](o_T). \tag{14}$$

This is the standard formulation of the Viterbi algorithm which can then be used to compute the likelihood score of the observation sequences.

3.3 Comparison between Keyword Search Algorithms

The two approaches described above each have a number of advantages and disadvantages. The template-based approach, for instance, uses the more accurate MAP estimate of a state $P(q|\mathbf{O})$ as compared to the state likelihoods $P(o_j|q_i)$ used in the second approach. However, the MAP estimate is computationally more expensive to calculate than the state likelihoods.

The optimal path in the template-based approach has the highest sum of posterior probabilities of states satisfying a particular sequence. This strategy has the disadvantage that the optimal path may contain state transitions which may not be likely considering only the state transition matrix. The state sequence calculated by the Viterbi algorithm, on the other hand, is guaranteed to have the highest likelihood of any of the possible state sequences.

In the left-right HMM of a keyword, frames of observations are condensed into a set of states. This approximation while being computationally efficient, removes information about the length of the different segments of the keyword and replaces it with more general information contained in the state transition probabilities. In the template-based approach, the longer a particular segment of speech is, the higher its influence on the final similarity score.

From a computational point of view alone, Viterbi-based algorithms are far superior than DTW-based approaches. An immediate advantage of this approach comes from the form of representation used in the algorithm. The SPR representation is calculated using the forward-backward algorithm which is more computation intensive than the calculation required for computing the state likelihoods in the SLR. Furthermore, the number of unique states in an utterance is generally much smaller than the total number of frames. Consequently, the lattice structure for the Viterbi algorithm has smaller dimensions than that used in the DTW. In addition to the larger dimensions of the lattice, the template-based approach uses inner product as the local distances between posteriograms. The computation of the inner product is a relatively expensive operation compared to the addition of logarithmic values carried out in the Viterbi algorithm.

During the preliminary trials, it was observed that the template-based approach provided marginally improved accuracy at a much higher cost of computations. Consequently, the second approach based on the left-right HMM of the keywords and the Viterbi algorithm for keyword search, have been used in this research.

3.3.1 Experiments on WSJ Corpus

To determine the efficacy of the developed VQbE system, a number of experiments were carried out on a single-speaker subset of WSJ corpus. While the selected subset lacks the diversity of multi-speaker speech, there still are significant contextual variations in the pronunciation of each utterance.

A 256-state EHMM was trained on a single-speaker subset (sd_tr_l\400) of WSJ corpus. The speech data was over three hours in length and was converted to SLR for the KWS trials.

For each keyword, the frames corresponding to an utterance were isolated manually and the most likely state sequence was calculated using the Viterbi algorithm. The calculated state sequence was then used to construct a left-right HMM for each keyword with the state transition probabilities derived from the EHMM.

Keyword search was performed using the sliding-window method where the likelihood score for each window of observations was calculated using the Viterbi algorithm before sliding the window forward. Observation sequences with the highest score were then classified with the help of the transcripts.

The accuracy of a keyword search algorithm generally improves with an increase in the length of the target keyword since longer sequences are statistically less likely to be repeated in the data set. To take this effect into account, the keywords were divided into three groups of short, medium, and long utterances.

A total of 40 keywords were selected for the experiments with lengths ranging from 3 to 11 phoneme. The first group consisted of keywords with the number of phonemes ranging from 2 to 5. The second group contained keywords with lengths between 6-8 phonemes. The longest keywords with lengths greater than 8 phonemes were placed in the third group. The longest keyword in this group consisted of 11 phonemes.

The results of the KWS trials are presented in Figure 9. The true positive rate is plotted along the y-axis with the number of errors per hour along the x-axis. The plot conforms to the general trend with the VQbE having the highest accuracy for the longest keywords at any given error rate. For keyword group 3, the VQbE system correctly identified more than half of the candidate keywords after making only three false positives. The fraction of correctly identified keywords, or true positive rate, rose to around 80% at 10 errors per hour.

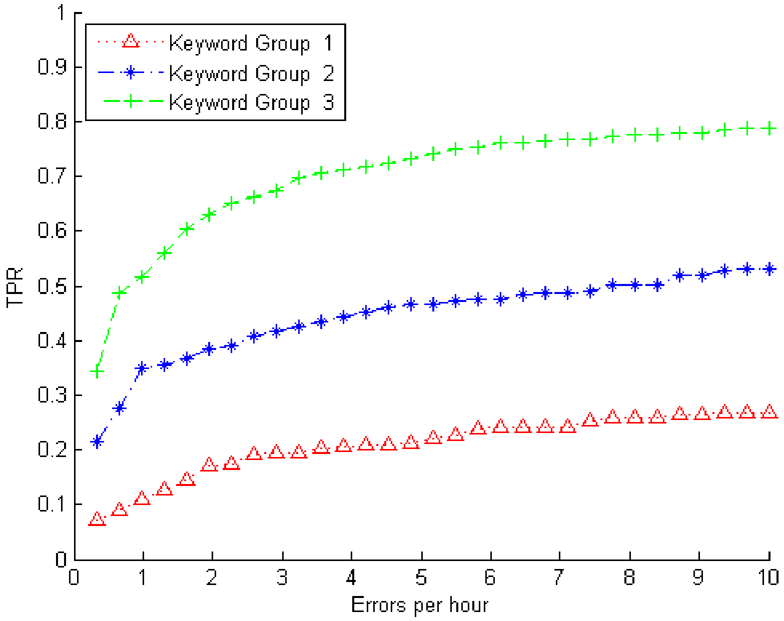


Figure 9: True positive rate vs. errors per hour for WSJ trials.

The accuracy of the VQbE was rather poor for the keywords in group 1 with only a quarter of candidate utterances found after making 10 errors per hour. However, shorter speech segments provide a smaller window of observations for the keyword search algorithm and are often part of other longer utterances.

The plot for keyword group 2 is more reflective of the average accuracy of the VQbE system. At an error rate of 10 errors per hour, the system successfully identified more than half of all the candidate utterances from more than three hours of continuous speech.

The precision of the implemented VQbE system is promising for a system that has been trained without any human supervision. However, the system failed to detect a significant fraction of the candidate utterances.

While there are no speaker-dependent variations in the test data, continuous speech still contains vast, contextual variations which can change the representation of an utterance far enough to miss detection. There were also a small number of candidate hits which were acoustically indistinguishable from the target keyword but had a different text representation and hence were rejected during the evaluation process.

Considering the fact that the training data in the trial is from a single-speaker, it is conceivable that the accuracy of the system on a multi-speaker every-day speech would be significantly lower than the one obtained in these trials.

3.3.2 Experiments on TIMIT Corpus

For the multi-speaker trials, a 256-state EHMM was trained using the TIMIT corpus. TIMIT contains speech from 630 different speakers of North American English from 8 major dialect regions within the United States. Compared to the WSJ corpus, speech from the TIMIT corpus contains a much wider variability of speech. It was not surprising that keyword search algorithm developed for a single-speaker application failed to detect many candidate utterances.

The variations inherent in the posteriogram of different instances of an utterance are illustrated in Figure 10. A posteriogram represents the probability$P(q_i|o_t)$ of each state i at time t. Time increases along the x-axis while the different states are plotted along the y-axis. For better visibility, the corresponding most probable state sequences for both the samples are denoted by (color-coded) points on the plot.

The plot depicts the state posterior probabilities for two instances of the utterance “California” spoken by two different speakers. The most probable state sequence for the first utterance, from a female speaker, is denoted by the red circles in the plot. Utterances 2, represented by the states denoted with “+”, was spoken by a male speaker.

There are natural variations between the two utterances along the time axis. The lengths of two utterances are different as well as the start and end frames of individual phonemes. More surprisingly, the sequences of states for each utterance are almost entirely different. A state occupancy histogram, indicating the number of times each state has been visited, reveals that a total of 47 different states were visited for the two utterances. Out of these 47 states, only 3 states were common between the two utterances.

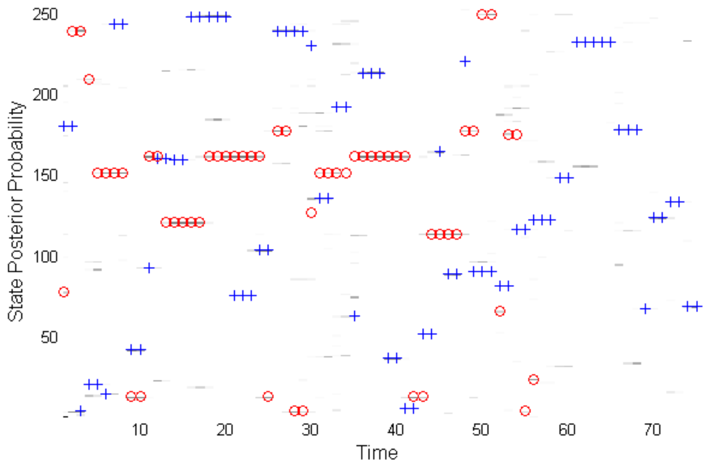


Figure 10: Posteriograms for two instances of the utterance “California”.

The example suggests that a single perceptual unit of speech may be mapped to more than one state of the EHMM. A single state in the EHMM represents observations that are similar both acoustically as well as perceptually. However, every perceptually similar observations may not have similar representations in the feature space and as a result, may be represented by different states of the EHMM.

From our preliminary investigation, it is clear that, in its current form, the VQbE system based on an EHMM of speech is not invariant to the contextual or speaker-dependent variations in speech. The proposed system must have a mechanism to identify and group together perceptually similar states of the EHMM. Any such mechanism would require significant changes in the way a keyword is modeled and would necessitate the development of novel search algorithms.

CHAPTER 4

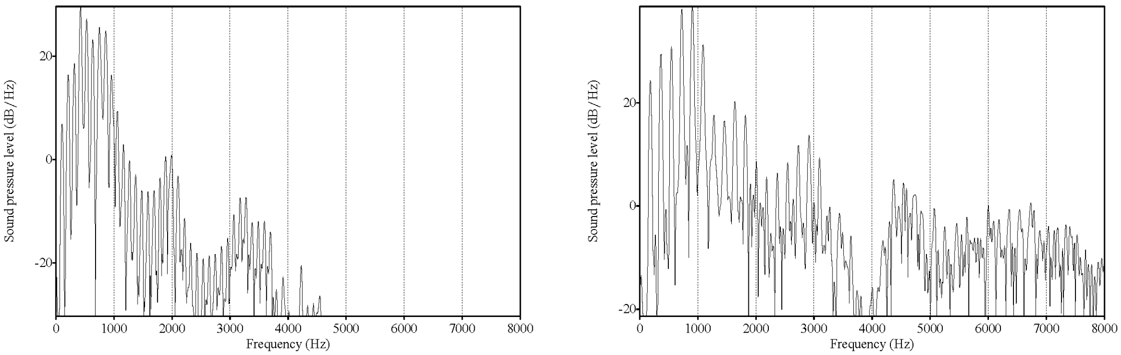
KEYWORD SPOTTING IN MULTI-SPEAKER ENVIRONMENT

The precision of the VQbE system based on an EHMM of speech is limited due to the wide variation displayed both by context dependent events as well as speaker variability. The range of variations extend to both time and frequency domains.

The rate of speech and as a result, the length of an utterance fluctuates with time. Furthermore, different people generally have different average rates of speech, leading to wide variations in the span of utterances. It is not just the total length of an utterance that can vary. Prosodic features, such as stress and rhythm, are important apparatus of communication and involve manipulation of lengths of individual phonemes to express emotions or convey additional information.

These differences are even more significant in the frequency domain. The frequency spectra of two perceptually similar segments of speech, extracted using Praat [44], are shown in Figure 11. The frequency spectrum of a segment of speech, a vowel, when spoken by a male speaker is shown in Figure 11a while Figure 11b contains the corresponding spectrum from a female speaker. The voiced nature of the speech segments is immediately evident from the presence of fine harmonic structure in the spectra. These spectral peaks occur regularly at integer multiples of the fundamental frequency. Female voices, in general, have higher pitch than voices from male speakers. This is also true for the example in Figure 11 where the harmonics are spaced wider in the spectrum corresponding to the female speaker, suggesting that fundamental frequency has a higher value for the female speaker than it has in the case of male speaker.

The position and shape of the formants also vary across different speakers. Formants are spectral peaks formed as a result of resonances produced by the articulators. These spectral peaks are wider and form the outer envelope of the fine harmonics generated by the oscillating glottal folds.



(a) *Male Speaker* **(b)** *Female Speaker*

Figure 11: Gender-specific variations in the spectrum of a vowel

There are significant differences in the number as well as position of the formants in the spectra in Figure 11. The spectrum in Figure 11a does not have any energy in frequencies higher than 4500 Hz. On the other hand, signal energy extends up to the maximum sampling frequency for the female speaker. This example suggests that the higher formants do not play a significant role in establishing the perceptual identity of a speech segment even though it may convey important speaker-specific information. Moreover, the first three formants, which characterize a vowel, are not constant for the two cases illustrated in Figure 11.

A speech recognition system must employ a number of different strategies, applied at different levels, to accommodate these variations while still rejecting false positives.

4.1 Speaker Invariance Schemes

The task of a speech recognition system is to recognize all the occurrences of a keyword in the presence of speaker, context, and channel induced variation. Channel induced artifacts are compensated by channel equalization and noise cancellation schemes and are not a focus of this research.

Speaker-dependent and context-induced variation have similar manifestations in time and frequency domain. Schemes aimed at compensating for these difference often conflict with the goal of minimizing erroneous detection of non-target utterances. The goal of this brief overview is to understand the various sources of speaker and contextual differences in speech segment and with an intention to highlight the deficiencies in an VQbE system based on an EHMM of speech.

4.1.1 Time Domain

In the time domain, algorithms based on dynamic programming paradigm optimally align an observation sequence to either a template or to a keyword model. Algorithms such as Dynamic Time Warping (DTW) and the Viterbi algorithm can identify a candidate utterance with different length composition than the original keyword.

4.1.2 Frequency Domain

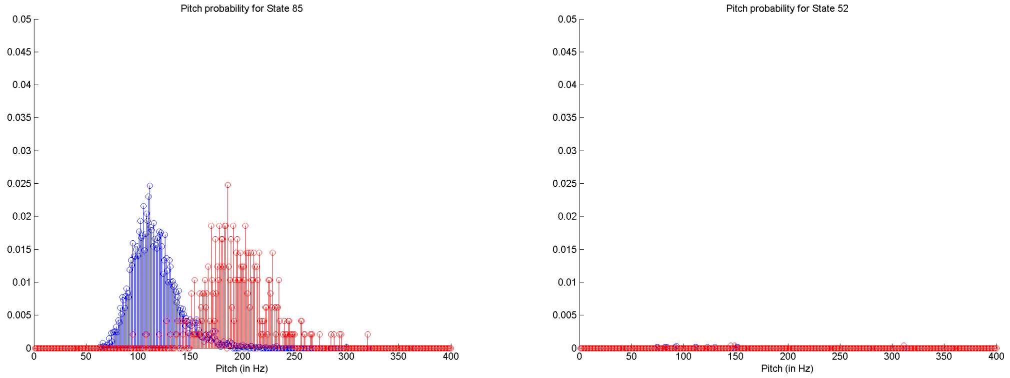
A number of different strategies are employed in VQbE systems to compensate for spectral variation that do not change the perception of a speech segment.

The form of speech representation plays an important part in the design of a speech recognition system. In a robust feature set, speaker or context-dependent variations must not change the representation of speech.

Features such as Linear Prediction Coefficients (LPC) and MFCC are robust to a number of sources of variation such as loudness and pitch differences in speech segments. For instance, loudness changes would significantly change the representation of a speech segment in time and frequency domains. The differences in energy levels of a speech signal, on the other hand, are limited to a single coefficient in the cepstral domain. Similarly, any channel artifacts affecting a signal during its propagation can simply be subtracted in MFCC domain compared to the complex deconvolution required in time domain.

4.1.2.1 Pitch Variation

In monotonic languages, changes in pitch levels do not affect the identity of a voiced segment of speech. The pitch values contain speaker-specific information and may convey some additional information like emotions. In tonal languages, on the other hand, a speech



(a) *State 85*

(b) *State 52*

Figure 12: Histogram of pitch values for different states of EHMM

segment with the same spectral shape but with a falling, rising, or constant pitch levels may each carry completely different meanings. Speech features, such as LPC and MFCC, discard the pitch information from the frequency spectrum and therefore, are more tailored to monotonic languages.

In MFCC, the smoothing of the speech spectrum is carried out by a process known as liftering. Liftering involves the low-pass filtering of the speech spectrum, which is achieved by discarding the higher-order cepstral coefficients.

In order to understand the effects of pitch levels on the states of an EHMM, the probability distribution of the fundamental frequency F0 was estimated for each state in the EHMM. The fundamental frequency is directly proportional to the perceived pitch of a speech segment, with higher F0 associated with higher perceived pitch levels. An estimation of F0 cannot be carried out in cepstral domain, therefore, estimates of F0 were made separately in Praat [44], a software package for acoustic analysis. The estimates, together with the corresponding EHMM state sequences, were used to generate a histogram of F0 for each state in the EHMM.

As an example, the histograms for the two states of the EHMM, one a vowel and the other a sibilant, are shown in Figure 12. The mean values of these states have been presented earlier in Figure 5. To reiterate the findings in Figure 5, the first state, labeled 85, has spectral characteristics that make it a vowel while the second one, state 52, has the characteristics of a sibilant consonant, most likely the palato-alveolar consonant /SH/.

F0 estimates for observations in state 52, an unvoiced consonant, have a flat histogram. The histogram for state 85 is much more interesting, and reveals a number of trends in F0 distribution. Firstly, most of the observations corresponding to this state were voiced, as can be expected from a vowel. More importantly, the pitch estimates have a bi-modal distribution with F0 estimates clustered around two distinct values. To get a clear picture, F0 histograms were then recalculated separately for male and female speakers. The histogram for male speakers is color-coded in blue while the red histogram corresponds to F0 estimates for female speakers. It can be seen that the distributions of F0 values for EHMM state are gender-specific with average F0 much higher for female speakers than it is for male speakers.

The characteristics of the probability distribution for F0 values have a number of important implications. The histograms demonstrate that MFCCs are invariant to differences in F0 values and therefore, pitch levels do not have an influence on the observation densities of states of an EHMM. Segments of speech would have similar representations in the MFCC domain and would be represented by a single state in the EHMM as long as they have similar spectral characteristics and exhibit similar state transitions. This would be considered an advantage for a monotonic language where the pitch levels have no contribution to the identity of a phoneme.

For tonal languages like Mandarin, the rising or falling pitch of a speech will be lost during MFCC calculations. A VQbE systems designed for these language must take into consideration the evolution of pitch levels for the task of recognition. Pitch information can be incorporated into EHMM models, trained using the HTK, by using multiple streams of input data during the training process.

The histograms for F0 for the voiced states of EHMM revealed another important characteristic. There is a definite pattern in values of F0 for a particular state of the EHMM.

Some of the states have a tendency of having high F0 values, while for others, F0 values are distributed around lower mean values. The statistics suggest that different states in the EHMM have a tendency to be spoken with certain pitch levels. This hypothesis is further strengthened by the observation that the trends in F0 distribution were consistent for both genders. The average value of F0 ranged from 111 Hz to 163 Hz for the male speakers in the data set, and for the female speakers, the fundamental frequency averaged from 177 Hz to 234 Hz.

This pattern of F0 values across the states of an EHMM can be used in a number of different applications. An EHMM of speech has been previously used in an low bitrate speech decoder [3]. F0 values and the state sequence corresponding to a speech segment was estimated and transmitted at a very low entropy. At the receiver, the speech spectrum was synthesized from the mean values of these states and the input excitation was generated from F0 estimates. An even more efficient speech decoder is conceivable, and currently being investigated, that would synthesize both the spectrum and the F0 values from the state sequence of EHMM of speech. With a conditional entropy close to 2 bits for an EHMM, such a speech decoder has the potential to achieve transmission rates of less than 200 bps.

4.1.2.2 Variation of Formants

Adult male speakers generally have longer vocal tracts and as a result, lower formant frequencies than children and females. However, formant variation are far more complex with formant frequencies exhibiting contextual as well as speaker-dependent variation that are highly non-linear. Furthermore, these differences may not be uniform for every acoustic unit of speech. In [6], it has been reported that the variation in first formant are more profound for open vowels than it is for half-open or rounded vowels. One the other hand, the second formant exhibits wider variability in the case of front vowels compared to back vowels, and this pattern is consistent for both genders.

Human perception is remarkably immune to variation in the formant frequencies. A human listener would have no problem in understanding speech with formant variation well beyond the range that would be considered normal. For instance, up-sampled or down-sampled speech may push the values of formant frequencies outside those encountered everyday yet a human listener would immediately adapt to those values.

Automatic speech recognition system, on the other hand, are not invariant to the changes in formant frequencies. Even a slight change in one of the formant frequencies may considerably change its acoustic manifestation in the feature space. In the case of MFCC, for example, the discrete cosine transform is not shift-invariant. Even a uniform shift in formant frequencies would change the corresponding cepstral coefficients. Further complicating the matter, formants frequencies may vary independently of each other and in a non-uniform fashion, making the observation distribution of a single acoustic class in the feature space highly non-linear .

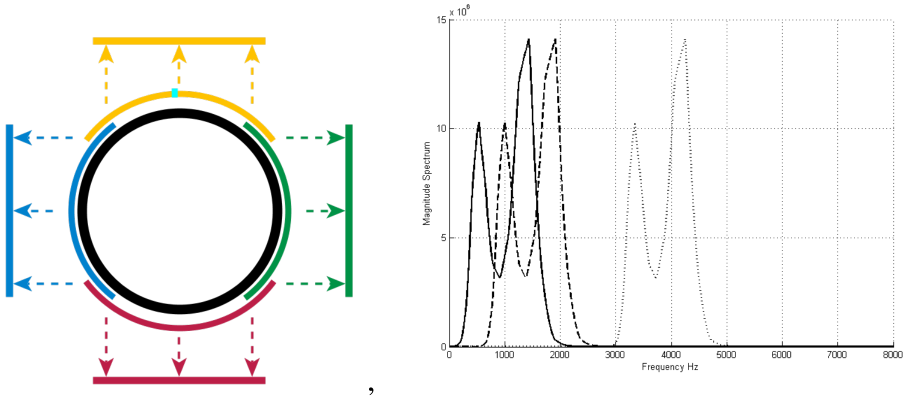
4.2 Manifold Model of Observation Density

The notion that the observation density for a single class could be in the form of a low-dimensional manifold is not new. In image processing, the projections of an object on a surface in different conditions of lighting, scale, or rotation, form a low-dimensional manifold in the high-dimensional feature space[45] and this realization has led to a large number of viewpoint-invariant image representation and recognition schemes.

In speech recognition, the low-dimensional manifold structure of the observation density has been the focus of a number of different approaches for phone classification [46], dimensionality reduction [47, 48], and design of speech features [49].

In order to understand the concept of a low-dimensional manifold and the possible implications that it may have on speech recognition, consider the example of two one-dimensional manifolds in R^2 (Figure 13).

The first example in Figure 13 is that of a circle in R^2. A circle is a one-dimensional



(a) *Circle, a 1-dimensional manifold* ,

(b) *Shifted spectrum*

Figure 13: Observation distribution in the form of manifold

manifold because it can be represented by two or more overlapping arcs, each homeomorphic to a (one-dimensional subspace) straight line.

In the case of circle, the distance from the origin ρ is the independent invariant which is unaffected by a change in angle θ. To rephrase, all the points on the circle have the property of being at a fixed distance from the origin and this identifying property does not change under the operation of rotation by an angle θ. This characterization of the problem is central to the justification of the clustering schemes presented in the later part of this chapter.

Consider the second example in Figure 13, and assume that the identity of the observation is preserved under a circular shift. Given the measurement space is of dimension N, this is an example of a one-dimensional manifold in N dimensional feature space. This artificial example has some similarities with the frequency spectrum of phonemes. The perception of a phoneme generally does not changes due to small shifts in formant frequencies. Similarly, voice quality changes from speaker to speaker with some speakers having well-defined, prominent formants. In other cases, the formants may have a wider spread or lower heights. Consequently, the dimensions of the observation manifold is larger in the case of speech since a larger number of parameters can be changed without changing the identity of the phoneme.

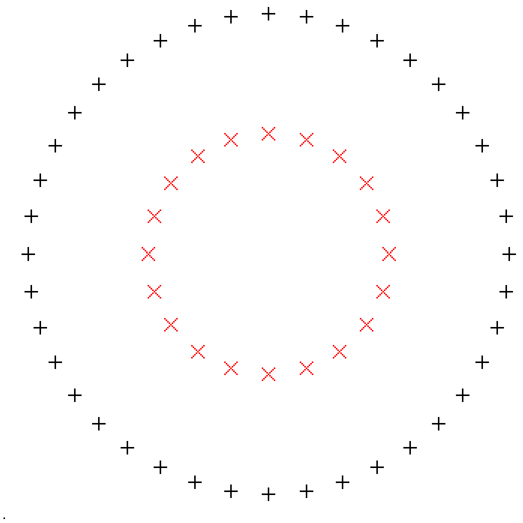


Figure 14: Classification of observations in the form of a manifold

The task of classification becomes non-trivial when the distribution of observations in a class is in the form of a manifold.

4.2.1 Clustering Observations on a Manifold

Most linear classifiers and proximity-based clustering algorithms fail completely when observation distribution within a class is in the form of a manifold. A major impediment to the task of classification arises from the fact that Euclidean distance are no longer a universal measure of dissimilarity or distance between two data points. This concept can be elaborated with the help of the example in Figure 14.

Consider the task of classification when the observation density of a class is in the form of a simple one-dimensional manifold. Consider the concentric circles of Figure 14. All the points on any given circle belong to a single class. While this classification task may seem trivial for a human being, automatic discovery of the two classes is extremely difficult for unsupervised machine learning algorithms.

For classification of observations in the form of a manifold, Euclidean distances are not a reliable indicator of class membership, though, Euclidean distances can be used in a small neighborhood or locality. For instance, two points with the distance between them

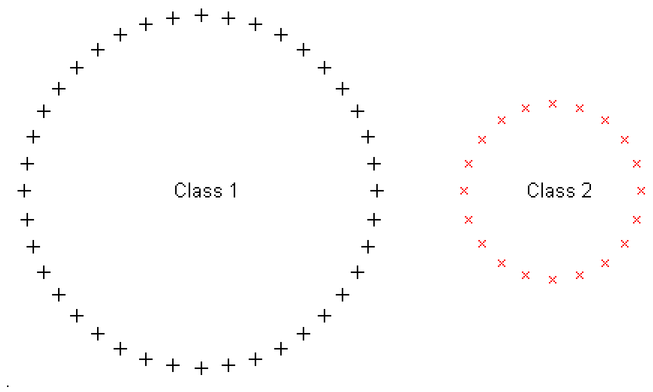


Figure 15: Isolating the classes using expert systems

smaller than the difference of the two radii would always belong to the same class. Distance measures, however, cannot be used for classification on a global scale. Any two points on the opposite sides of the outer circle are further apart from each other than they are from all the points on the inner circle. As such, neither a linear classifier nor a proximity-based clustering algorithms can be used for this task.

In this example, classification can be performed in at least three different ways. The first method is the simplest (and the most frequently used in speech recognition). Delegate the task to an expert system that actually knows how to solve this problem. Given the solution of a sufficiently large training data, new data points can be classified using their proximity from the pre-classified sample observations.

The second solution involves identification of a common property among the members of a class. In this example, members of the same class are at a constant distance from a single point in R^2, the center of the circle. Thus, any point can now be classified by measuring its distance from this reference point.

The third method involves the knowledge of an identity-preserving operation. The circle is a one-dimensional manifold in a two-dimensional feature space. It has exactly one degree of freedom, i.e., there is a dimension along which the independent invariant (distance from origin) remains constant. In the above example, rotation is one such operation that does not change the class membership of an observation. Consequently, every pair of points that can be mapped together through rotation, can be clustered together in a single class.

4.3 State Parameter Estimation for HMM

HMMs are capable of modeling complex non-linearly distributed, time-varying observations. Instead of the static clustering scheme described in the last section, consider the movement of minute and hour hands of a clock. The positions of the clock hands are again distributed in the form of a circle. Additionally, the position of the clock hands changes with time in a particular manner. HMMs can model the clock hand trajectories with a number of connected states. These states can model both the positions as well as the movements of the clock hands.

HMMs are particularly suited to model speech where both the values and time characteristics of observations must be taken into account. The Baum-Welch algorithm, described in Sections 2.1.2 and 2.3, relies on proximity measures between observations for estimating the parameters of a HMM. The emission probability $b_k(o_t)$ is inversely proportional to the square of the Mahalanobis distance between the observation o_t and the mean value μ_k of the k^{th} mixture. The parameter estimates are influenced by an observation in proportion of its emission probability, and consequently, in proportion of its proximity from the mean value of a mixture.

Given that the observation density in different classes of speech is non-linear, an algorithm that relies on proximity measures only is not expected to estimate the parameters of different classes of speech. However, the solutions to the current problem are still very similar to the ones discussed in the last section. These approaches are explained in the next section.

4.3.1 Learning by Example

Most contemporary speech recognition systems are based on the paradigm of learning-by-example. This approach is similar to the first solution for the static clustering task described in Figure 14. In supervised speech recognition systems, the task of the classification is actually performed by a human listener on a sufficiently large data set. A prerequisite to this approach is the knowledge of the phonetic structure of speech so that the training data can be isolated into these classes. As stated earlier, the observation distribution within a single class is non-linear and require a model that can represent non-linear distribution. A mixture of Gaussian if often used to model the observation density.

Human perception is remarkably immune to speaker-dependent or context-induced variation in speech. The transcriber listens to tens or hundreds of hours of speech and assigns time-aligned labels to each occurrences of a phonemic class in the training data. In doing so, a large body of samples of a particular class is isolated. The observation distribution for each class can now be modeled, in isolation from other classes, using a proximity-based parameter estimation algorithm.

The high precisions of modern speech recognition can be partially be attributed to the use of time-aligned annotated data for training. The transcription process offloads the difficult task of classification of the samples of a speech segment in a multitude of acoustic environment, and in the presence of speaker-dependent and context-induced variation, to a human being.

The transcription process is usually the most expensive phase in the development of a speech recognition system in terms of time and capital. The transcription process is generally carried out at utterance-level while the statistical models are generated at subword level. The orthographic transcription of speech is converted to a phonetic one with the help of a pronouncing dictionary. The use of pronunciation lexicon considerably accelerate the transcription process by automating the conversion of word labels to phoneme labels.

This approach of learning-by-example has also been carried out for the discovery of the phonetic structure of a large EHMM in [50] and [51]. An EHMM of North American English was trained using the TIMIT corpus [37]. The phonetic transcription of the training data and the associated state sequences of EHMM allowed the identification of the substructure of an EHMM associated with a particular phoneme. In this research, however, the target languages are resource-limited and as a consequence, acoustic models of speech must be trained without any linguistic resources.

In the next sections, we present two unsupervised clustering schemes based on a single large EHMM of speech. By definition, a manifold can be decomposed in a number of coordinate charts where a manifold is "similar" to a low-dimensional subspace. The open set on the manifold covered by a single coordinate chart is a neighborhood or locality where Euclidean distances are relevant (due to a homeomorphism mapping it to a subspace). In the proposed approach, an EHMM models the non-linearly distributed classes in speech with a large number of states. If the number of states are sufficiently large, then the collection of observations represented by a singe state would would belong to a single neighborhood.

Conceptually, a single state would consist of a set of observations that are both acoustically and perceptually similar. In this setting, observations are acoustically similar if they have similar representations in the feature space. Perceptual similarity, on the other hand, deals with how a speech segment is perceived by a listener. Hence, two instances of the same vowel uttered by an adult male speaker and a child are similar perceptually but may have completely different manifestations in the feature space.

The observation density for a single phoneme would be modeled by a number of such states. The goal of a clustering schemes, in this approach, is to identify the states in an EHMM which are similar perceptually but may have significantly different acoustic representations.

4.3.2 Nearest-neighbor Clustering

In Section 4.2.1, three different approaches were described for the classification of observations distributed in the form of a one-dimensional manifold. Given that the observation densities of the classes are continuous and non-overlapping, the classification task can also be performed using nearest-neighbor clustering.

This approach involves density modeling of the classes using a large number of Gaussians in such a way as to ensure that all the observations modeled by a single Gaussian belong to a single class. In the case of concentric circles, for example, this criteria can be satisfied by using a sufficiently large number of Gaussians such that the maximum distance between observations within the same Gaussian component is less than the difference in the radii of the circles.

Gaussian components can then be merged together using an nearest-neighbor clustering scheme. Gaussian components with the smallest distance between them are merged together to form a new cluster, consisting of a mixture of Gaussians. For clusters with more than one Gaussian components, the minimum distance between any of two Gaussian components from different classes can be used as the inter-cluster distance. It is obvious that a nearest-neighbor clustering will group together all the Gaussians on a single circle using a chain of increasingly longer Gaussian components.

This clustering scheme works in a similar manner on an EHMM of speech. The states of the EHMM can be grouped together based on the proximity of the corresponding observation distributions in the feature space. Clustering can be stopped when the desired number of clusters are obtained or if the minimum distance between two clusters reaches a particular threshold.

In the case of speech, however, the observation distributions are neither continuous nor the different densities are non-overlapping. While a sufficiently large set of training data can be used to estimate the observation density of each phonetic class, some of the phonetic classes in speech overlap in feature space. Diphthongs and the stop part of stop consonants are examples of acoustic units that overlap with other acoustic classes.

To avoid the merger of states from different acoustic classes, a K-nearest-neighbor approach can be adopted where the overall distance between two clusters is calculated based on the distances of at most k elements of each class.

Two different distance measures were explored in this research. The 39-dimensional MFCCs consist of 13 static cepstral features along with 13 delta, and 13 delta-delta coefficients. The delta and delta-delta features improve recognition accuracy by capturing the time variation of the static cepstral features. In the first approach, denoted as DMS, the proximity between two states is measured using the static cepstral coefficients. The second approach considers the full 39 cepstral coefficients to compute the dynamic distance between two states.

4.3.3 Clustering using an Identity-Preserving Operation

Representations of speech, including MFCC, are not completely invariant to speaker-dependent variation in speech. Speech recognition systems frequently employ speaker normalization and adaptation schemes to compensate for speaker induced differences. The task of speaker normalization or adaptation involves the application identity-preserving operations that transforms either the speech features or the model parameters to reduce the mismatch between model and observations.

Vocal tract length normalization (VTLN) has been widely used in speech recognition both in the context of normalizing data from different speakers as well as in adapting statistical models to new speakers. These speaker-adapted speech recognition systems have higher accuracy than their non-adapted versions.

Single-parameter vocal tract length normalization (VTLN) schemes assume that all the formant frequencies undergo similar changes. These VTLN techniques compensate for formant variation by a linear scaling of the frequency axis.

However, the variation in formant frequencies among different speakers is far from linear. Alternate approaches to VTLN, such as those proposed in [52] and [53], use multiple parameters for speaker adaptation and normalization, and are more effective than their single parameter counterparts.

In traditional speaker adaptation and normalization schemes, the parameters for VTLN are estimated from a sufficiently large set of training data using maximum likelihood approaches. Formant variation tend to average out across large number of samples and as a result, there is an upper limit to the number of parameters that can be used for VTLN.

In the case of an EHMM, each state of the model can be represented by a single point in feature space, the mean value of its observation distribution. Consequently, the optimal frequency warping path can be estimated in much finer detail in the case of EHMM.

In the next section, a non-linear, dynamic programming approach to frequency warping is presented for the calculation of optimal alignment between spectrally similar states of an EHMM.

4.3.3.1 Dynamic Frequency Warping

Dynamic frequency warping (DFW) is an algorithm that can be used to optimally match the spectral features of two states in an EHMM using frequency warping. This technique is based on the premise that two states of an EHMM are similar if their corresponding spectra can be aligned by frequency warping within a small user-configurable range.

Variation in second and third formants, on average, are larger than those for the first formant [6]. DFW takes this into account and allow a linearly increasing margin of frequencies for higher formants as shown in Figure 16. The optimal warping path can lie anywhere within this allowable frequency range and is calculated using dynamic programming.

To compare states of the EHMM using DFW, the mean value of each state in cepstral domain must be converted back to linear frequency domain [54]. The MFCC coefficients are generated by sampling the speech spectrogram along Mel-spaced frequency intervals using the triangular function T, then taking the logarithm of the signal amplitude and finally calculating the cepstral coefficient by taking its discrete cosine transform denoted by the

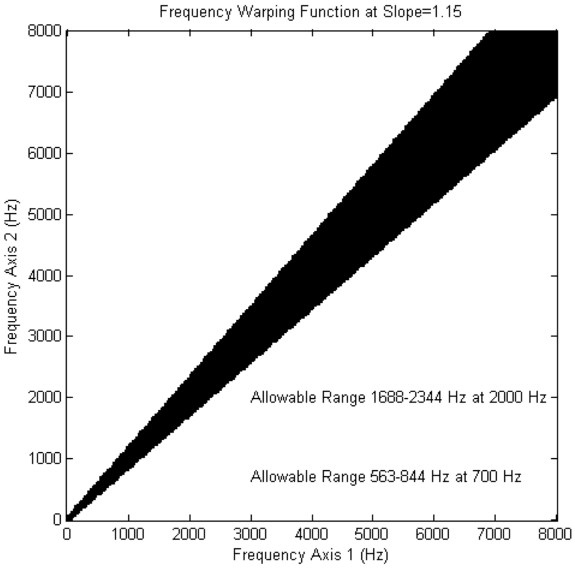


Figure 16: Dynamic Frequency Warping

operator C.

Let m_i be the MFCC coefficients representing the mean value of the state i. We can calculate the mean value s_i of the state in linear frequency domain using the approximate inverse operations to T and C.

$$s_i = T^{-1}\exp(C^{-1}m_i) \tag{15}$$

The frequency-domain representation of a model state, unlike its MFCC representation, is sensitive to differences in amplitude. Before calculating the optimal frequency warping between two states, the mean values must be normalized with respect to energy, using scaling coefficients c_i and c_j. The optimal frequency warping $\mathcal{F}$ between the two energy-normalized states can then be calculated using dynamic programming. The l_2-norm of the difference between the optimally warped mean values is taken as the dissimilarity measure between the two states.

$$d(s_i, s_j) = \min_{\mathcal{F}} \left| c_i s_i - \mathcal{F}(c_j s_j) \right| \tag{16}$$

Formants not only differ in position but also in height and width. Wider formants tend

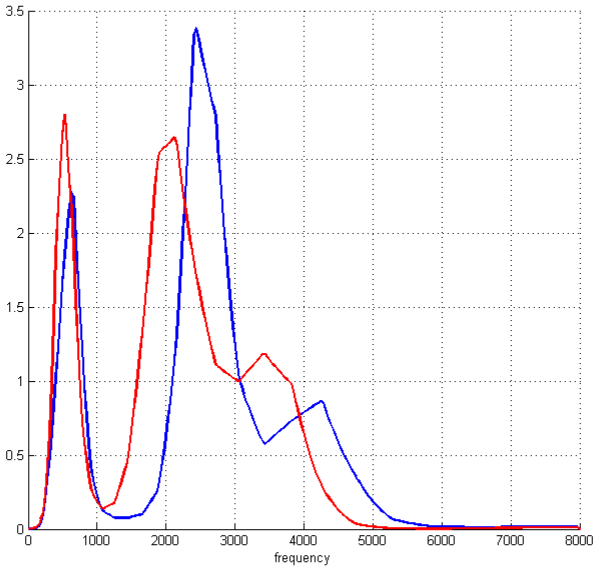


Figure 17: Similar states discovered by the DFW-based clustering algorithm

to have lower peak values than narrower ones. DFW between such states at equal energy can lead to larger dissimilarity measures due to a mismatch of formant heights. Instead of separately normalizing the energy of each state, a more appropriate constraint would be to limit the combined energy of s_i and s_j, and to scale them to minimize the dissimilarity measure. The new scaling factor k can be calculated using gradient-descent search. The new value of the dissimilarity measure is estimated by iteratively optimizing the values of k and $\mathcal{F}$.

$$d(s_1, s_2) = \min_{k,\mathcal{F}} \left| \sqrt{k} c_i s_i - \sqrt{2-k}\mathcal{F}(c_j s_j) \right| \tag{17}$$

4.3.4 Hierarchical Clustering

We can obtain a dissimilarity matrix for the EHMM by applying DFW between every pair of states in the EHMM. The distance measure defined in Eq. 17 satisfies the commutative property and a total of $\frac{N(N-1)}{2}$ pair-wise calculations are required to populate the dissimilarity matrix for an N-state EHMM.

Using an agglomerative hierarchical clustering algorithm, states with the lowest dissimilarity measure are progressively merged together into a single cluster. Clustering can be stopped when the dissimilarity measure reaches a threshold or when the desired number of clusters is achieved.

At the end of the clustering phase, states with similar spectral shapes are grouped together into a single cluster. Figure 17 shows an example of similar states discovered by the clustering algorithm.

CHAPTER 5

DESIGN OF VOICE-QUERY-BY-EXAMPLE SYSTEM USING AN EHMM OF SPEECH

In absence of any time-aligned labeled data for training, observations are grouped into states of an EHMM based on their proximity in the feature space and exhibition of similar temporal characteristics. Since most feature sets, including MFCCs, are not invariant to all speaker-dependent variation, speech segments with the same perceptual characteristics may occupy distinct regions in the feature space, and consequently, mapped to different states of the EHMM.

In the last chapter, a number of different schemes have been presented for the discovery of states with similar spectral or cepstral characteristics. A knowledge of these clusters of similar states can be used by a VQbE system to construct a improved keyword models with an ability to search for a wider variety of candidate utterances matching a query.

In this approach, a EHMM-based VQbE system must take into account the one-to-many relationship between phonemes and the states of the EHMM. A clustering of similar states is not the only way that a query could be represented with more than sequences of EHMM states. In some cases, more than one instance of a query may be available, each corresponding to a different sequence of states of an EHMM. A query must be modeled in such a way as to allow for its alternate pronunciations while preserving the unique time characteristics of each pronunciation.

In this Chapter, two different approaches have been proposed to integrate the knowledge of similarity of EHMM states to leverage the performance of a VQbE system. In the first approach, similar states of an EHMM are merged together to form a new superstate and in the process, this merger creates a new EHMM. The second method uses a different approach and considers the state clustering as a logical construct and preserve the original structure of the EHMM. The detailed explanation of these methods and their relative merits

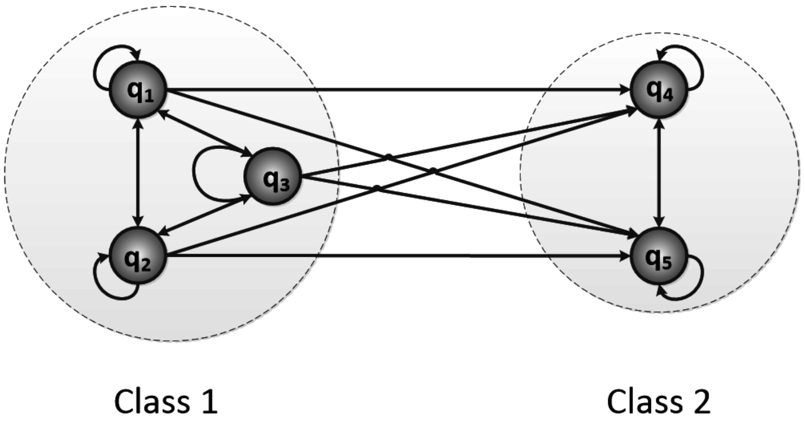


Figure 18: The composition of states of a CISI-EHMM of speech.

are discussed in the next sections.

5.1 A CISI-EHMM of Speech

A single state of an EHMM, when trained in an unsupervised manner, consists of observations that are perceptually as well as acoustically similar. Using a set of mappings between phonemes and the EHMM states, an EHMM can be reduced to a context-independent, speaker-independent EHMM (CISI-EHMM) by merging similar states into a new super-state.

In a CISI-EHMM, a single state is a perceptual unit of speech and composed of a Gaussian mixtures, each a collection of observations occupying distinct regions in the feature space.

A CISI-EHMM can convert speech into a form of representation that is invariant to common contextual and speaker-dependent variations.

In the next section, a procedure is presented to estimate the parameters of an CISI-EHMM given the mappings between phoneme and states of an EHMM.

5.1.1 Estimating the Parameters of the CISI-EHMM

The new EHMM is defined by the parameters (π', A', B'). These parameters can be estimated from the parameters of the initial EHMM (π, A, B) and the result of the state classification.

The classification of EHMM states can be carried out in a supervised manner or any of the unsupervised clustering schemes presented in the last chapter. The classification of the N states of an EHMM is represented with M classes $C_1, C_2, \cdots, C_M$, where each class is a set of indices of the states of the EHMM.

An EHMM has a stationary distribution π such that

$$\pi A^n = \pi. \tag{18}$$

The elements of π are equal to the probability of occupancy of a particular state, i.e., $\pi_i = \Pr(q = s_i)$. Since an EHMM is aperiodic and irreducible, from the fundamental theorem of the Markov chain, the stationary distribution exists and is unique. It can be calculated by taking the left eigenvector of the transition matrix A corresponding to an eigenvalue of 1.

The stationary distributions π' for the new CISI-EHMM can be directly calculated from the stationary distribution of the original EHMM. For a superstate C_i in the new EHMM, the stationary distribution π'_i is given by Equation 19.

$$\pi'_i = \sum_{j \in C_i} \pi_j. \tag{19}$$

The distribution of observations corresponding to a state of the CISI-EHMM can be represented with a mixture of Gaussian distributions. The individual Gaussian distributions have been estimated during the training of the original EHMM. However, these distributions are not all equally probable and each can be assigned a weight to reflect this fact. In the CISI-EHMM, the weight w_i of each state can be assigned according to the likelihood of each state.

$$w_k = \Pr(s_k \mid C_i) = \frac{\Pr(s_k)}{\Pr(C_i)} = \frac{\pi_k}{\sum_{l \in C_i} \pi_l}. \tag{20}$$

The emission probability $b_j(o_t)$ can then be calculated as

$$b_i(o_t) = \sum_{j \in C_i} w_j \mathcal{N}(o_t; \mu_j, U_j), \tag{21}$$

where μ_j, and U_j are the mean and covariance of the j^{th} state in the original EHMM.

The state transition matrix determines the probability of transition between any two states. The original state transition matrix is of size $N \times N$ with the (i, j) element equal to the probability of transition from state i to state j. The sum of probabilities in each row of the transition matrix must be equal to 1. In the modified EHMM, the N states of the original EHMM have been grouped into M classes, leading to a new transition matrix of dimensions $M \times M$ for the CISI-EHMM.

A state transition from one state of the same class to another is considered as a self-transition. Similarly, all the state transitions from any of the states in one class to any of the states in another must be considered to calculate the transition probability from class i to class j.

$$a'_{ij} = \sum_{k \in C_i} \Pr(s_k \mid C_i) \sum_{l \in C_j} a_{kl} = \frac{\displaystyle\sum_{k \in C_i} \sum_{l \in C_j} \pi_k a_{kl}}{\displaystyle\sum_{k \in C_i} \pi_k} \tag{22}$$

5.1.2 Keyword Model and Keyword Search Algorithm

The VQbE systems proposed in Chapter 3 can be used, without any major modifications, for keyword search using a CISI-EHMM of speech. The only difference between these EHMMs is that states with single Gaussian distributions have been replaced with super-states having a distribution modeled by a mixture of Gaussians. A query can be represented with a left-right HMM and keyword search can be carried out using the standard Viterbi algorithm.

The HMM model of a query consists of a number of superstates connected in a left-to-right fashion. The observation distribution of the new superstates accurately reflect the non-linear distribution of acoustic units of speech. However, the merger of states of an EHMM does not preserve the individual state transition probabilities. As discussed in Chapter 2, the conditional entropy of an EHMM of speech is very low. For most of the EHMMs trained in this research, the conditional entropy is found to be close to 2 bits. The low value of entropy is an indication of the high short-term correlations in speech. These short-term correlations lead to sparse connectivity among the EHMM states.

The elements of a superstate have in common certain similarities in cepstral or spectral domains. However, these states may have significantly different state connectivity. For example, states within a single cluster that correspond to different speaker groups could very well be isolated from each other.

The distinct transition probabilities of these states retain the time-evolution of the utterances that they model. These transition probabilities give an EHMM-based VQbE system the ability to distinguish among observation sequences based on their time characteristics. The merger of these states lead to a loss of information, replacing the individual state transition probabilities of the original EHMM with ones averaged from a number of states.

In the next section, an alternate VQbE system is presented that has been particularly tailored to preserve both the spectral as well as temporal characteristics of individual states in an EHMM while retaining the ability to model individual segments in utterances with multiple EHMM states.

5.2 Graphical Keyword Model and 3D Viterbi Algorithm

In conversational speech, there are large variations in the pronunciation of an utterance. Speakers from different dialect groups pronounce certain phonemes differently. In some cases, phonemes may be added or omitted altogether from the pronunciation of an utterance.

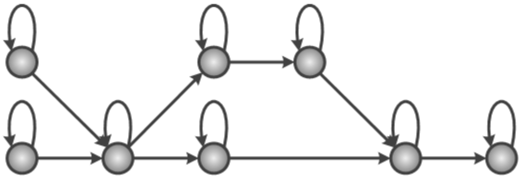


Figure 19: Statistical model of keyword based on an EHMM of Speech

It is not difficult to see that different instances of the same utterance may correspond to different state sequences. Left-right HMMs are restrictive in the sense that they always assume that the same sequence of states constitutes an utterance.

We propose a keyword model that reflects the fact that utterances and phonemes generally form sub-structures consisting of a number of states of the EHMM. The graphical keyword model consists of a sequence of sets of states of the EHMM. Each set in the sequence may contain a number of states corresponding to different realizations of a speech segment. While the keyword model has a sequential structure, a state in a particular tier can be connected to any state in the same set or adjacent sets in the model, each with a different transition probability.

Figure 19 shows an example of such a keyword model for an utterance. In this example, the first phoneme of the utterance has two different forms modeled by the two states in the first set in the keyword model. This example also illustrate how an added or omitted phoneme can be modeled in this scheme.

In contrast to LRHMM, the graphical model based on an EHMM is richer in detail and captures a wider range of information about the utterance. Since the elements of the graphical keyword model are states of the EHMM, the statistics of each state are estimated using all the training data unlike LRHMMs where models are trained individually using separate training data.

The precision of a keyword search algorithm can be improved significantly by using a background model. An added advantage of proposed approach is that both the keyword model and the background/filler model can be generated from the states of EHMM. The states that are not part of the keyword model become the filler or background model. In contrast, background models must be trained separately for LRHMMs.

5.2.1 A 3D Viterbi Algorithm

The proposed keyword model may have multiple entry and exit points, with multiple paths joining each pair of entry and exit points. Likelihood score calculations cannot be done using the traditional Viterbi algorithm due to the presence of additional paths in the proposed keyword model. However, recast in three dimensions, the likelihood score of an observation sequence can be calculated recursively using the proposed keyword model.

Let an utterance be represented by a sequence of set of states $\mathbf{q} = (\bar{q}_0, \bar{q}_1, \bar{q}_2, \ldots, \bar{q}_M)$ where $\bar{q}_i$ is the i^{th} set in the keyword model. The states in the model are connected with each other according to the state transition matrix A. Instead of a two-dimensional lattice structure, a three-dimensional lattice is used with the alternate states in a set being represented along the third dimension. Figure 20 shows the modified lattice structure for the modified Viterbi algorithm.

The Viterbi algorithm is relatively unchanged with the main difference being that now a node in the lattice can be reached from all the nodes corresponding from the same set or from the preceding set. Let y_j be a state in set Y and x_i be a state in the preceding set X, then the log-likelihood score at Y is given by

$$\delta_t(Y) = [\max_{i,j}(\delta_{t-1}(x_i)a_{x_i y_j}, \delta_{t-1}(y_j)a_{y_j y_j})]b_{y_j}(o_t). \tag{23}$$

In the original Viterbi algorithm, there were only two transitions at each node - a self transition into the same node and a transition from the previous node in the LRHMM to the current node. Hence, the keyword search could be performed in order of $\mathbf{\Theta}(MTP)$ operations when the test set consists of P frames of speech data, T being the size of each window and M the number of states in the LRHMM of keyword.

If implemented directly, the added dimension of the lattice considerably increases the

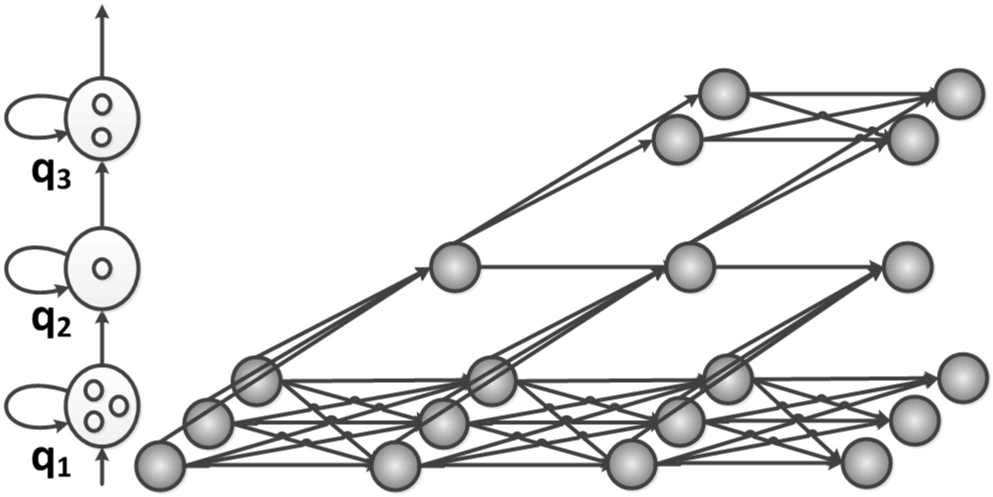


Figure 20: A three-dimensional lattice for graphical keyword model

algorithm's computational complexity. The number of nodes in each window of the modified lattice increases to $N \times M \times T$ nodes where N is the average number of states in each set. Furthermore, at each node, there are $2N$ possible state transitions, increasing the total computational cost of the keyword search algorithm to $\Theta(N^2MTP)$.

5.2.2 Length-normalized Optimality Criteria

We have assumed so far that the observation sequence contains a perfectly aligned occurrence of the keyword. However, when processing continuous speech, the start and end positions of a matching utterance are not known in advance. One way to circumvent this problem is to assume that all the occurrences of the keyword have the same length as the original template.

In this sliding-window approach, a keyword-sized window of observations is processed by the Viterbi algorithm to calculate the likelihood that it is emitted from the keyword model. The starting position is then advanced by one frame. It is obvious that this process is computationally very expensive. Another downside to this approach is the presence of multiple hits near a matching candidate. As a matching candidate starts to enter the window used by the Viterbi algorithm, the log-likelihood score gradually increases until it reaches the peak when the candidate is optimally aligned to the search window. As a

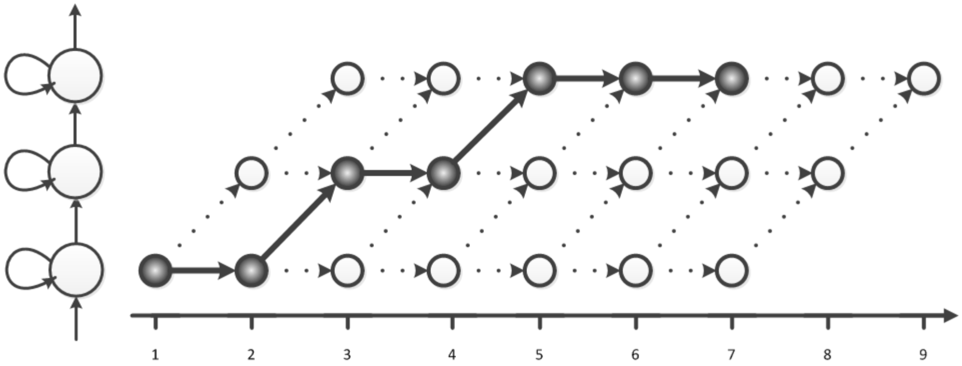


Figure 21: The Viterbi window for a hypothetical match

consequence, the likelihood scores must be filtered to remove duplicate high scores from a single candidate.

Considering the large temporal variations in conversational speech, it would be impractical to assume that all the realizations of a given utterance would have the same length. As the likelihood $\Pr(o_t \mid q_i)$ will always be less than or equal to 1, uncertainty can only increase as more and more observations become part of the optimal path. The optimality criteria used by the Viterbi algorithm, i.e., the sum of log-likelihood of an observations sequence, always favor the shortest possible length of the optimal path. Hence, the observation sequence that maximizes the log-likelihood may not always be the same sequence that contains the actual candidate.

5.2.3 Length Estimation

The lattice structure and the mapping of the observation sequence to a sample 3-state keyword model can be seen in Figure 21. The optimal path in the lattice corresponding to the true position of the utterance is represented by the dark nodes connected by solid lines. In this example, the length of the sliding window is 9 while the true candidate length is 7. It is obvious that the likelihood score at $t = 9$ is not truly representative of the candidate and would be smaller than the true score of the candidate. Similarly, the likelihood score will be maximum at $t = 5$, again not indicative of the true score of the keyword.

However, if the likelihood score is normalized with respect to the observation length and given that the observations samples from time $t = 6$ to $t = 9$ have a log-likelihood score that is greater or equal to the running average up to that point, then the length-normalized likelihood score would be maximum at $t = 7$, the true length of the candidate utterance.

This suggests a mechanism for estimating the true length of a candidate utterance. Instead of maximizing the sum of log-likelihood of a fixed length observation sequence, the observation sequence that has the highest log-likelihood, considering its length, is selected. In its simplest form, this measure is $\frac{\delta_T(M)}{T}$. A more general form of this measure would be $\frac{\delta_T(M)}{T^L}$ where the parameter L can be used to favor shorter or longer candidates.

5.2.4 Fast Viterbi algorithm using Token-passing

In addition to estimating the length of a candidate keyword, the use of length-normalized likelihood can significantly reduce the computational complexity of the modified Viterbi algorithm.

In the sliding window approach, the lattice structure is populated for a window of observations and the Viterbi algorithm is calculated in isolation for this set of observations. All the paths that reach a given node have originated from a single starting index and hence have the same length. The path having the highest likelihood is retained while all the other paths are discarded.

If the log-likelihood score is normalized with respect to length, then it can also be used to select the best path among those originating from different starting indices. We can achieve significant computational improvement by using the length-normalized optimality criteria at all the nodes of the lattice and not just at the one corresponding to the last state of the keyword model. Instead of applying the Viterbi algorithm on a window of observations and advancing this window one frame at a time, we can populate a single lattice for the entire observation sequence. The Viterbi algorithm can be applied only once on this lattice and can be used to select and propagate the paths that are optimal considering their length.

To do this, however, the algorithm must keep track of the length of each path. This can

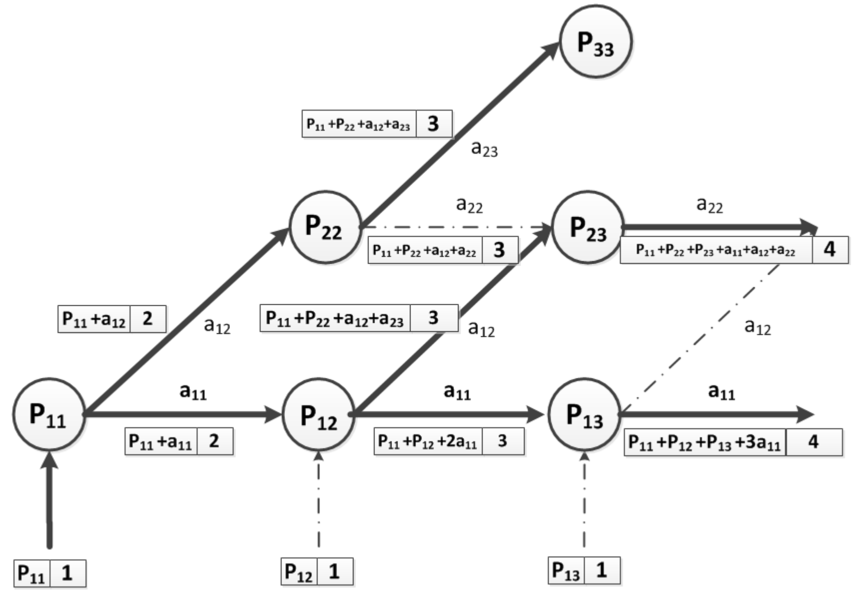


Figure 22: The token-passing algorithm

be achieved by a token passing approach. Figure 22 illustrates the token-passing mechanism in the modified Viterbi algorithm.

It can be seen in Figure 22 that a new path is initiated at each frame by creating a new token. The token contains the log-likelihood of the path, which at this point is equal to $\Pr(o_i|q_1)$, and its initial length. At each node in the lattice, the token having the higher length-normalized log-likelihood would be chosen; the log-likelihood is updated in the same manner as in the standard Viterbi algorithm and the length field in the token would be incremented. The token is then forwarded to the adjacent nodes in the lattice until it reaches the last row, corresponding to the last state of the keyword model. At the end of this phase, each of the nodes in the last row of the lattice contains the token with the log-likelihood and length of each optimal path terminating at that node.

The reduction in the complexity of the algorithm due to token passing is significant. While the sliding window approach has $M \times T$ number of operation for each window leading to a computational complexity of $\boldsymbol{\Theta}(MTP)$ for P frames of data, the token-passing approach only require computations to the order $\boldsymbol{\Theta}(MP)$.

The token passing algorithm also offers a simple mechanism for filtering the duplicate results due to a single matching candidate. The optimal path that reaches the last row annihilates not only the sub-optimal paths from the same starting frame but also those emanating from neighboring frames. Also, in the last row of the lattice, more than one node in a neighborhood may still share a segment of the optimal path and have the same starting index. One such example is illustrated in Figure 21 where the nodes in the last row at indices 5, 6, and 7 are part of the same optimal path and have the same starting index. Using the length field of the token, the start index of each optimal path can be traced back to the starting frame, and all the tokens except the one having the largest length-normalized log-likelihood can be discarded.

The final form of the proposed keyword search algorithm incorporate the three-dimensional lattice structure along with the token-passing mechanism to efficiently scores observation sequences based on a length-normalized optimality criterion. The proposed algorithm not only returns a likelihood score for each observation sequence but also provides an estimate of the length of each candidate sequence.

CHAPTER 6

EXPERIMENTS AND RESULTS

A framework for developing a Query-by-Example system based on an EHMM model of speech has been presented in this research. EHMMs are easy to train and significantly reduces the dimensionality of the problem by modeling speech with a number of hidden states connected in a Markovian fashion.

The states in an EHMM consists of observations with similar acoustic and perceptual qualities. Nevertheless, a single phoneme could be mapped to more than one state of an EHMM due to the nonlinear distribution of observations and the unsupervised nature of the training. In Chapter 4, two different clustering schemes for the identification of perceptually similar states of the EHMM were introduced. In Chapter 5, a framework for a VQbE system based on an EHMM of speech has been presented. The framework consists of a graphical keyword model for queries and a modified 3D Viterbi search algorithm to take advantage of the proposed keyword model. The graphical keyword model can be constructed from multiple examples of a query or it can incorporate the phoneme-to-states mappings to generate a robust model of the query that is invariant to most speaker-dependent variations.

No matter how good a VQbE may look on paper, the true value of an approach can only be measured by testing it on natural continuous speech. The experiments that were carried out to evaluate the different clustering schemes, keyword models, and search algorithms are the focus of the current chapter.

The experiments can be divided into two categories. A VQbE system is a complex system that employs a large number of parameters to control different aspect of speech modeling and keyword search. These parameters may range from the obvious model parameters, such as the number of states in an EHMM and the number of mixtures within a state, to the not so obvious ones, such as the slope and offset variables controlling the optimal path during the Viterbi algorithm. These parameters must be optimized in order to achieve the true potential of an VQbE system.

The initial set of experiments were performed on the TIMIT corpus to take advantage of the availability of time-aligned word and phoneme labels. The primary goal of these experiments were to find the optimal configurations for a particular approach. These configurations were used for the comparison of the systems based on different clustering schemes. The VQbE system with the highest precision in these experiments was then used in the second group of experiments.

The second part of the chapter focuses on the comparison of the proposed VQbE system with the current state-of-the-art in the field of zero-resource VQbE. Over the last few years, a number of databases have been introduced to serve as a unified platform for development and evaluation of zero-resource VQbE systems. The EHMM-based system, developed in this research, has been evaluated on two such databases, i.e., the JHU CLSP isolated word-pair recognition task and the MediaEval 2013 Spoken Web Search database. The task descriptions, observations and results for these experiments are described in the second part of this chapter.

6.1 Comparison of Different Clustering Schemes

The primary goal of this set of experiments was to investigate the effects of different clustering schemes on the precision of VQbE systems. These clustering schemes have been proposed to address the speaker-dependent variations in state sequence representation of speech segments.

The TIMIT Corpus contains phonetically rich speech from a large number of speakers. The variety of speech and the size of the database is adequate for the exploratory nature of these experiments. Furthermore, time-aligned word and phoneme labels for the utterances simplify and expedite the evaluation process.

In the first phase of these experiments, an initial benchmark was established using the traditional Viterbi algorithm and left-right HMMs of the queries. The second phase of these experiments were carried out with the graphical keyword model of queries constructed with the help of different clustering schemes.

Two different schemes have been proposed in this research to identify and group states of an EHMM with similar spectral characteristics. The first method is a form of nearest-neighbor clustering and uses the distance between the mean values of states in the feature space (cepstral domain) for state mergers. In the second approach, two states are deemed similar if their spectral representations can be mapped through an identity-preserving frequency warping.

For each query in the evaluation set, graphical keyword models were constructed using each of the clustering schemes. These keyword models were then used for keyword search using the 3D Viterbi algorithm and the precision measures observed for each system were recorded and compared with the initial benchmarks. The details of these experiments including the test setup, evaluation metric, graphical keyword model derivation and the results of the experiments are presented in the following sections.

6.1.1 Test Setup

The TIMIT corpus is composed of two parts; approximately 3 hours of training data and 1.5 hours of speech for test purposes. The separation of the corpus into training and test sets is aimed at supervised speech recognitions systems. In these systems, the training data is used to train keyword or phoneme models which can then be evaluated on the test set. An EHMM, on the other hand, has a completely different training methodology which does not rely on any linguistic resources. Taking this fact into consideration, the training and test sets were combined for these experiments.

An EHMM with 256 states was trained using the combined TIMIT data. This EHMM was the largest EHMM that could be trained without the loss of states during training. The redundant states in an EHMM is an indication that the number of states in the model is larger than that required to model the data. These redundant states will exponentially increase the computational complexity of the search algorithm without making any improvements in the speech model.

The number of queries chosen for the trials were comparable to that reported in similar studies [30][18]. The system developed in [30] was based on posteriorgrams of phoneme models trained in supervised manner and the test setup consisted of 40 queries and 6 hours of test data from the Fisher Corpus. The system described in [18] was unsupervised in nature, with the evaluation set composed of 30 queries and 7,375 utterances from the MIT lecture corpus [55].

In the following experiments, the evaluation set of queries is composed of 60 utterances, ranging in length from 3 to 11 phonemes. Keywords were divided into three groups according to the number of phoneme in each utterance. The first group consisted of keywords with 3-5 phonemes while the second and third group had 6-8 and 9-11 phonemes respectively. The average number of occurrences of each keyword in the test set was 19.85.

6.1.1.1 Evaluation Metric

The metric used to evaluate the precision of the systems in [30][18] have been P@N, and P@10. For a systems which returns a ranked list of candidates, P@N stands for the average precision of the top N candidates and would represent the point on the precision-recall curve where precision and recall are equal to each other. Similarly, P@10 is equal to the average precision of the top ten results returned by the system.

The calculation of these measure require inspection of a small number of top hits from a system and would be convenient in a scenario where an exhaustive evaluation of the result cannot be carried out automatically due to lack of labeled data. However, these precision measures provide a snapshot of the system at a single point on the precision-recall curve.

A finer level of detail about the accuracy of the system can be observed by plotting the probability of detection, i.e., the fraction of the total candidates that have been found against the number of false positives that have been encountered. Although, this plot does not paint as complete a picture of the system precision as Receiver Operating Characteristics (ROC) or Detection Error Tradeoff (DET) [56], this plot covers in sufficient detail the area of interest for most practical applications.

6.1.1.2 Derivation of Graphical Keyword Models

For the initial benchmark, left-right HMMs of the queries were constructed using the most likely EHMM state sequence corresponding to each query. Multiple consecutive occurrences of a state in the sequence were replaced by a single state to condense the sequence into a left-right HMM of the query. The state transition probabilities of the states in the left-right HMM were derived from those in the EHMM. Specifically, the self-transition probability a_{ii} of i^{th} state in the left-right HMM, which is related to the average duration of a state, is set equal to the value in the EHMM and the probability of transition to the next state is set to $1 - a_{ii}$.

The graphical keyword models can be constructed with the help of the left-right HMMs developed in the last step and the hierarchical clustering of the states described in Section 4.3.4. The clustering schemes described in Chapter 4 groups together states that are perceived to be similar based on some similarity measure.

The left-right HMM of a query can be converted into a graphical keyword model by replacing each state in the left-right model with all the states from the same cluster. Each of added state retain their statistics, observation densities and transition probabilities, from the original EHMM. The resulting graphical keyword model may have more than one terminal states and potentially multiple paths connecting each pair of terminal states. This graphical model can then used for keyword search using the three-dimensional Viterbi algorithm.

6.1.2 Nearest-neighbor Clustering Scheme

In these experiments, the clusters of states used for construction of the graphical keyword models were obtained by nearest-neighbor clustering of states in the cepstral domain [57]. There are a number of different configurations in which such a clustering could be used for keyword model generation.

Cluster size	Group 1		Group 2		Group 3	
	DMS	DMD	DMS	DMD	DMS	DMD
256	29.58		40.05		56.90	
224	16.40	17.96	38.54	39.04	62.50	62.54
192	15.33	16.90	36.97	36.62	60.88	60.30
160	13.50	15.22	37.98	37.98	59.18	57.46
128	13.28	13.36	35.22	34.18	52.35	53.42
64	11.52	11.87	33.66	32.67	47.80	49.83

Table 2: Probability of detection for different clustering schemes.

The states of the original 256-states EHMM were merged in a agglomerative hierarchical clustering scheme with the initial number of clusters equal to the number of states in the EHMM. Each subsequent state merger resulted in a modified model with the total number of clusters one less than before. A number of these models with different total number of clusters, each at an intermediate step during the clustering, were used to generate graphical keyword models and the recall rates of the corresponding systems were tabulated as a function of the number of clusters in the model.

The core 13-coefficient MFCC correspond to the spectral shape of a speech segment while the velocity and acceleration coefficients represent the time evolution of these coefficient. The addition of time derivatives of the cepstral coefficients have been reported to significantly increase the precision of speech recognition systems. As discussed in Section 4.3.2, both 13-dimensional static (DMS) coefficients as well as full 39-dimensional cepstral (DMD) coefficients were used for clustering and the recall rates of the corresponding systems were computed for speech models with different number of clusters.

The fraction of true positives discovered by the system at a rate of 10 false positives per hour, for different distance measures and cluster sizes is listed in Table 2. As discussed in Section 6.1.1, system recalls at different rates of false positives were calculated separately for different keyword groups.

The statistics in Table 2 reveal the detection probability of the system to be slightly

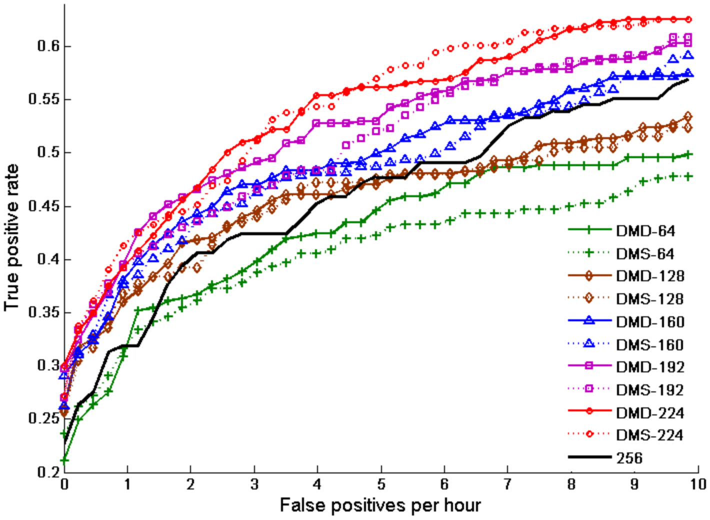


Figure 23: Detection probability for queries in Group 3.

higher when the full 39-dimensional MFCC (DMD) were used for clustering. The difference is recall, though small, is consistent across the different configurations of the systems. This observation reinforces the notion that the addition of time-derivatives of the cepstral coefficients increases the precision of speech recognition systems.

As expected, detection probability for a particular system was different for different keyword groups. Relatively higher recall rates were observed for longer utterances at the same error threshold, with the system recall dropping significantly for keywords in Group 1.

Another trend that can be observed in Table 2 is that the probability of detecting a true positive for a system declined with a decrease in the total number of clusters. System recall for speech models with number of clusters equal to 160, 192, and 244 were higher than that of the left-right models for the queries in Group 3 (Figure 23). For every other group, the overall detection probability actually decreased relative to the system using left-right HMMs.

When averaged across the keyword groups, the fraction of queries found at a false positive rate of 10 per hour was equal to 0.4112 for speech model with 224 clusters, and 0.4013 and 0.4046 for EHMMs with total clusters equal to 208 and 192. This figure was

Clusters	Group 1	Group 2	Group 3	Overall
256	29.58	40.05	56.90	42.18
224	17.58	40.23	70.48	42.76
192	18.00	41.49	72.22	43.90
160	16.25	42.09	72.14	43.49
128	15.69	42.20	71.4	43.10
64	16.09	42.40	72.86	43.78

Table 3: Probability of detection for different cluster sizes and clustering schemes.

0.4218 for the system using left-right HMMs of the queries.

The result of the experiments conducted on the system based on DFW clustering is presented in the next section, and a detailed analysis of the results is carried out in the subsequent section.

6.1.3 Clustering using Dynamic Frequency Warping

In contrast to the nearest-neighbor clustering scheme that was based on proximity of states in the cepstral domain, the states of the EHMM in this approach are compared in linear frequency domain. The spectra of two states are mapped to each other through dynamic frequency warping. Dynamic frequency warping is an identity-preserving operation that attempts to warp the two spectra within a limited frequency interval in order to minimize the mean square error between them.

The probability of detection, at an error rate of 10 false positves per hour, for systems based on EHMMs with different number of clusters is given in Table 3.

System recalls for these systems are much higher than that observed for EHMMs based on nearest-neighbor clustering in cepstral domain. The overall detection probabilities for the systems would have been even higher if not for the queries in Group 1.

The gain in the detection probability by the addition of new states in the keyword model is clearly correlated with the length of a query. The graphical keyword models led to the highest improvements in system recall for longer queries, a slight increase of keywords of medium length and the detection probability actually dropped for the queries in Group 1,

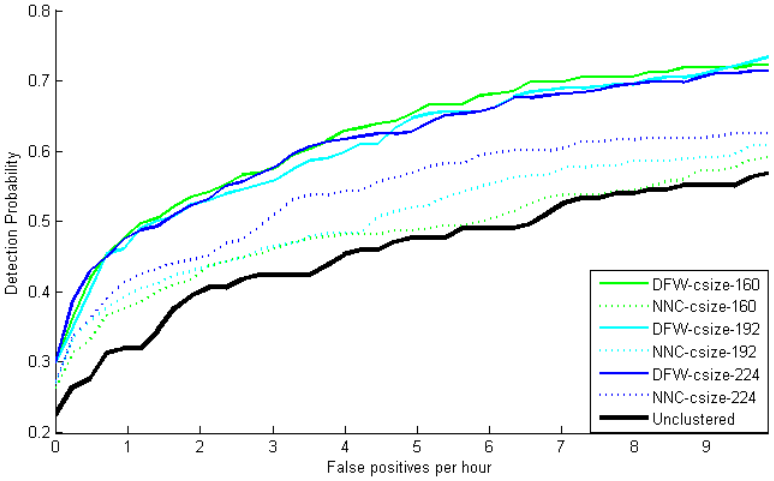


Figure 24: Plot of true positive rate for keyword group 3.

which are 3-5 phonemes long.

The overall recall of the system peaked for the speech model with total number of clusters equal to 192. The precision measures, otherwise, exhibit similar characteristics to those for nearest-neighbor clustering. The detection probabilities of the systems, for queries in Group 3, at a false detection threshold of 10 errors per hour is shown in Figure 24. For comparison, the values observed for nearest-neighborhood clustering scheme are also shown in dotted lines.

6.1.4 Comparison of Clustering Schemes

The results of the experiments in the last sections reveal a number of similar trends for both the systems. System recall at any given false positive rate, was directly proportional to the length of a query. For the shortest queries in the test set, detection probability actually declined for both the clustering schemes.

This pattern can be attributed to a number of different factors. Shorter queries are generally composed of a smaller number of states and there is a larger impact on the overall log-likelihood score of a candidate by a match or mismatch of a single state in the model. For example, a change of a few frames may convert the utterance “bat” to “pat.” On the contrary, longer utterances can still be identified if a small number of frames are masked or replaced.

Moreover, shorter speech sequences are encountered more frequently in speech than longer utterances. Consequently, shorter queries tend to have a larger number of similar-sounding false positives. Moreover, shorter utterances are often part of other longer utterances. For example, one of the keywords in the test set "age" is also part of other longer utterances such as "voltage," "heritage," and "percentage" but these speech segments would be marked as false by an evaluation scheme dependent on word transcriptions of the utterances. These speech segments are acoustically identical and can only be differentiated by a language model. Hence, the precision of the current systems can be improved by utilizing a language model at a higher layer to differentiate between such word pairs.

In context of keyword search, the states that are part of a keyword model define the observation space of the query while the remaining states constitute the background model. The expansion of the keyword model with the states from the same cluster, increases the allowable range of observations for a given segment of speech. While this addition accommodates larger variations in a speech segment, it also increases the likelihood of false positives. For longer keyword lengths, the advantages of expanding the keyword model outweighs its disadvantages, resulting in higher system precision. For shorter keywords, on the other hand, expanding the keyword model significantly reduces the system precision because of the large number of false positives that enter the keyword model space, some of them potentially acoustically indistinguishable from the query.

These experiments revealed a clear winner in terms of precision. The VQbE systems based on DFW-clustering had a higher recall at the same values of false positive rate than those based on nearest-neighbor clustering in every configuration. The DFW-based systems consistently outperformed their counterparts at every cluster size and for every keyword group.

The outcome of the experiments is not unexpected and has been partly explained in

Section 4.1. A nearest-neighbor clustering scheme is based on the assumption that observations in close proximity of each other are similar and different classes of these observation occupy non-overlapping regions in feature space. Both of these assumption may not be true in the case of MFCC features. The discrete-cosine transform used in the calculation of cepstral coefficients is not shift-invariant. The formants, on the other hand, vary in frequencies for different speakers. Furthermore, the different formants may undergo different frequency shifts for different speakers [6]. Hence, a non-linear shift in formant frequencies may result in similar observations occupying distinct regions in space. While a nearest-neighbor clustering scheme, in theory. may eventually unite these distinct observations into a single class through a chain of intermediate observations, the different classes must have non-overlapping observation densities in order to do so. In speech, however, different phonemes are often overlapped as in the case of diphthongs and stop consonants. A nearest-neighbor clustering scheme, while effective for a more speaker tolerant feature sets, did not perform as well for MFCCs.

DFW-clustering, on the other hand, is a feature-independent scheme. The mean values of EHMM states are transformed back to frequency domain where their corresponding spectra are compared. The dynamic frequency warping algorithm performs a formant matching within a user-configurable frequency range. The results in 3 confirms the success of DFW-clustering in identifying perceptually similar states of the EHMM.

There is a strong correlation between the number of clusters in an EHMM and the precision observed for these systems. The hierarchical clustering scheme merges clusters with the shortest distance between them. States involved in earlier mergers are more likely to be similar to each to each other than those in subsequent mergers. As a result, the gain in precision of the system would be higher from these relevant states than the later ones.

It can be seen from Tables 2 and 3 that the number of clusters that led to the highest precision is different for both schemes. For the systems based on nearest-neighbor clustering, system precision declined with a reduction in the total number of clusters in the

EHMM. This indicate that only the first few state mergers may potentially have contributed positively to the detection probability of the system and most of the later addition only degraded the precision of the system.

In the case of DFW-clustering, on the other hand, the EHMM with 192 clusters led to the highest observed precision. This particular EHMM has been formed after 64 state mergers of the original 256-state EHMM. This fact, and the overall precision of the systems based on DFW suggest that DFW-clustering leads to a better clustering of states of the EHMM.

6.2 Experiments on JHU CLSP Word Collection

In summer of 2012, a multidisciplinary workshop was held at Center of Language and Speech Processing at Johns Hopkins University on the topic of "Zero-Resource Technologies and Models of Early Language Development." One of the goals of the workshop was to identify a unified evaluation framework for speech recognition systems that should be relevant across multiple disciplines. Towards this goal, an isolated word collection, with around eleven thousand words, was created.

A number of approaches consisting of different speech features as well as keyword models were implemented and tested on the word collection. Features such as MFCC, PLP and Intrinsic Spectral Analysis (ISA) [49] were used to represent utterances which were compared using Dynamic Time Warping algorithm. In these template-based approaches, the frame-to-frame comparisons were carried out using cosine product and symmetric KL-divergence. In addition to the template-based approaches, systems using GMM [18] and non-parametric Bayesian (NP Bayes) approach [58] to model speech were also investigated.

6.2.1 Test Setup

In addition to the creation of a word-collection, the workshop proposed a recognition task for the evaluation of different speech recognition systems.

Let M be the utterances $\mathcal{W} = w_{i_{i=1}}^{M}$ in the word collection where each word $w_i = x_1x_2...x_{T_i}$, $x_t \in R^d$ is a T_i-length sequence of d-dimensional observation vectors.

The word collection has been designed with DTW-based systems in mind using a symmetric similarity/distance measure. As a result, it is assumed that there are $\frac{M\times(M-1)}{2}$ (around 60 million) unique word example pairs in the collection. Any word pair, $(w_i, w_j) \in \mathcal{W} \times \mathcal{W}$, $i \neq j$, can only have the following four combination of words.

1. C_1: Same word, same speaker (SWSP).

2. C_2: Same word, different speakers (SWDP).

3. C_3: Different word, same speaker (DWSP).

4. C_4: Different word, different speakers (DWDP).

By design, $|C_1| \ll |C_2| \ll |C_3| \ll |C_4|$. Hence, the number of instances of the same word spoken by the same speaker is much smaller than instances of same words spoken by different speakers. The majority of the word-pairs would consists of different utterances spoken by different speakers.

The similarity or dissimilarity between a word pair is calculated using dynamic time warping. A word-level dissimilarity or similarity matrix is calculated using length-normalized dynamic time warping distances. For a particular threshold, the detection is deemed successful if both the utterances in a word-pair returned by the system belong to the same word class. These utterances may or may not be spoken by the same speaker. Similarly, if the word-pair returned by the system consists of different word types, then it will be treated as a false positive.

6.2.2 Evaluation Metric

This dissimilarity or similarity matrix obtained in the last section is used to calculate the average precision (AP) of a system.

Average precision is defined as:

$$AP = \int_0^1 p(r)\mathrm{d}r. \tag{24}$$

The word-pairs are sorted into a single list of 60 million word-pairs using their similarity/distance scores. Since the precision-recall curve is discrete in this case, the integral in Equation 24 is replaced with a summation. The final form of the equation for the discrete precision-recall curve is given below:

$$AP = \frac{1}{M}\sum_{k=0}^{N} p(k)rel(k). \tag{25}$$

In the above equation, N denotes the number of elements in the sorted list of candidates and M is the number of true positives in this list. The precision at the k^{th} element is denoted by$p(k)$ and $rel(k)$ is an indicator function and is equal to 1 if the k^{th} element is a true positive and 0 otherwise.

6.2.3 Issues Specific to HMM-based Systems

The systems implemented and evaluated at JHU CLSP workshop 2012 used cosine product and symmetric Kullback-Liebler divergence to compare corresponding frames of utterances in a word example pair.

Cosine product is a similarity metric whereas the symmetric KL-divergence is a distance metric but both these measures satisfy the symmetry property. A similarity or distance measure will satisfy the symmetry property if $d(x, y) = d(y, x)$. As a result, the order in which the two utterances appear in a word example pair will not affect the outcome of the measurement.

In a VQbE system based on an EHMM of speech, on the other hand, a query is used to construct a (graphical) keyword model which is then used to calculate the likelihood that an observation sequence is generated by the keyword model. For a word example pair A-B, the likelihood of utterance B given a keyword model of A will not be the same as the likelihood of observation sequence A given a keyword model of B. In fact, the observation

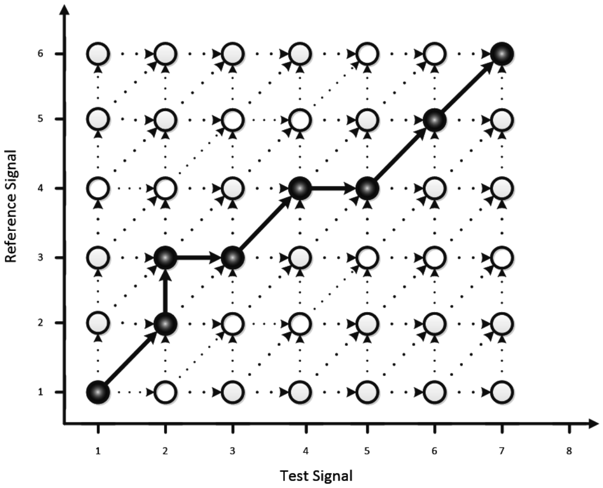


Figure 25: Lattice Structure for DTW

sequence used to generate the keyword model is not guaranteed to be the one with the highest likelihood for the given model. Consequently, the number of unique word example pairs in the current setup will be twice that for systems in [40].

The similarity (cosine product) and dissimilarity (Symmetric KL-divergence) measures used in the JHU workshop have another desirable property that is advantageous for the calculation of a single average precision for a system. The range of values for both cosine product and KL-divergence is much smaller than log-likelihoods. Subsequently, the range of variations in the scores of word example pairs is much smaller when either cosine product or KL-divergence is used compared to log-likelihoods.

The overall similarity (dissimilarity) measure between two utterances is calculated using the DTW algorithm. Let l_1 and l_2 be the lengths of two utterances in a word example pair. The lattice generated for the algorithm would consist of $l_1 \times l_2$ nodes, and the node at i^{th} row and j^{th} column in the lattice would contain the local similarity/distance measure between frame i of first utterance and frame j of the second utterance. The optimal path connects the nodes at (1,1) and (l_1, l_2) and has the largest (smallest) possible sum of local frame-to-frame similarity (distance) scores among all the possible paths connecting the terminal nodes in the lattice.

Let N be the number of lattice nodes in the optimal path. Given that there are no restriction on horizontal or vertical steps, the number of nodes in this path would be at least equal to largest of l_1 and l_2, and cannot be greater than $l_1 + l_2$. Hence, the total similar/distance measure between two utterances cannot be less than 0 or greater than N, assuming the local metric has a value between 0 and 1.

The similarity/distance measure between two utterances in this scheme would be a function of the lengths of the two utterances. The similarity score between two short utterances would be smaller than that for two utterances with the same level of similarity but longer lengths. In order to get a true estimate of the average precision, the scores must be ranked according to their similarity and this score must not be influenced by the lengths of utterances in the pair.

The systems in the JHU Workshop normalized the scores by dividing the similarity/distance score for each word example pair by the sum of their lengths $l_1 + l_2$. The selection of $l_1 + l_2$ as a normalizing factor is interesting since it is an upper bound on the number of nodes N in the optimal path. The optimal path is more likely to be closer to the diagonal connecting the terminal nodes of the trellis for similar word pairs. However, the normalizing factor $l_1 + l_2$ led to a higher value of average precision than the true value of N, during the reproduction of results in this research.

6.2.4 Results

The average precisions for different systems studied at JHU Workshop are presented in Table 4. The score for the EHMM-based system, which was not part of the workshop, has also been added to this list.

The average precision observed for the EHMM-based system is 35.1%. As discussed in the last section, the number of word-example pairs processed by the EHMM-based system is two times the number of word-example pairs in the original DTW-based systems in the workshop.

The precision of the EHMM-based system is higher than any of the template-based systems except for the ISA features. Furthermore, the EHMM-based VQbE system also

Table 4: Average precision of an EHMM-based system vs. systems evaluated at JHU Workshop.

Representation	Metric	Average Precision
EHMM	log-likelihood	0.351
ISA	cosine	0.496
PLP	cosine	0.348
MFCC	cosine	0.338
MFCC+NB Bayes (507 units)	symm KL	0.445
MFCC+GMM UBM (507 units)	symm KL	0.271
MFCC+GMM UBM (50 units)	symm KL	0.236
ISA+NB Bayes (507 units)	symm KL	0.464
ISA+GMM UBM (507 units)	symm KL	0.447
ISA+GMM UBM (50 units)	symm KL	0.332
English NN Posteriorgrams	symm KL	0.846

scored a higher average precision than both the 50-component and 507-component GMM-based systems using the same (MFCC) features.

The precision of every system based on ISA features, on the other hand, is higher than any of the systems based on MFCC or PLP features. Nevertheless, the EHMM-based system, which uses MFCC features, scored a higher average precision than a 50-unit GMM UBM using ISA.

An EHMM is a feature-independent statistical modeling scheme and can be trained on any speech features including ISA. It would be futile to compare an EHMM-based system using one feature set to an alternate speech modeling schemes that uses any other feature set, such as ISA.

An EHMM-based VQbE is a clear winner when compared to any of the GMM-based system. However, it scored lower than the non-parametric Bayesian model of speech. To reiterate, an EHMM-based system processed a collection of word example pairs that is double in size to that processed by other systems and it also suffers from score normalization issues arising from the use of log-likelihoods.

In order to investigate the extent of degradation in precision due to issues rooted in score normalization, the mean average precision for the set of trials was calculated in the case of EHMM-based system. Instead of compiling the score for every word pair into a single large list, the average precision was calculated for each trial and then averaged over the entire set. This approach has the advantage that the candidates for each trial are evaluated on a separate scale and hence, any variations in candidate scores across different trials do not affect the calculated precision for the system.

For each trial, a graphical keyword model of an utterance was constructed and was used to evaluate every word examples in the set. The candidates were then sorted according to their log-likelihoods and the average precision was calculated for the trial. Finally, the precision measure for the entire set was calculated by averaging the precision of individual trials.

The mean average precision calculated for the EHMM-based system was 68.43%. Thus, the average precision of the EHMM-based system on a trial-to-trial basis is more than double the figure that has been calculated when the trials are merged into a single list. The differences in these values indicate the score normalization to be a substantial factor contributing to the lower calculated average precision of the system. However, a mean AP of 68% is also not a conclusive proof that an EHMM-based system is better than the NP-Bayes-based systems since these precision measures have been calculated in a entirely different settings. A true test of these two systems can only be carried out on a single database and using a single precision measure that does not favor a particular system.

6.3 Experiments on MediaEval 2013 Corpus

MediaEval is a community-driven benchmarking initiative offering tasks and evaluation databases in a number of different areas including but not limited to speech, audio, images, tags, users and contexts.

In 2013, a number of databases and associated tasks were offered. These include social event detection for social multimedia, search and hyperlinking of television content, geo-coordinate prediction for social multimedia, violent scenes detection in film, preserving privacy in surveillance videos, question answering for the spoken web, soundtrack selection for commercials, similar segments of social speech, retrieving diverse social images, emotion in music, crowd-sourcing for social multimedia and spoken web search.

The Spoken Web Search task has been defined as "searching FOR audio content WITHIN audio content USING an audio content query." The primary emphasis of the task is the development of speech recognition systems for resource-limited languages, with the accompanying evaluation database provided without any additional linguistic resources.

6.3.1 Test Setup

The Spoken Web Search 2013 database contains 20 hours of unlabeled speech with two sets of queries for development and evaluation purposes. The SWS2013 database is particularly challenging as it contains speech from nine different languages and has been recorded in a number of different acoustic environment. Both the development and evaluation sets contains approximately 1500 word examples from more than 500 word types. The ground truth files for the development set have been provided to allow developers to optimize their systems. The evaluation set, on the other hand, was tested by the organizers and the results shared with the participants.

6.3.2 Evaluation Metrics

The systems participating in the challenge were evaluated using Term-Weighted Value (TWV). TWV is a weighted combination of the miss and false alarm error rates over the set of queries Q. It is defined as

$$TWV(\theta) = 1 - \frac{1}{\mid Q \mid} \sum_{\forall q \in Q} (P_{miss}(q, \theta) + \beta P_{fa}(q, \theta)), \tag{26}$$

$$TWV(\theta) = 1 - (P_{miss}(\theta) + \beta P_{fa}(\theta)).$$

In the above equations, P_{miss} and P_{fa} denote the probabilities of missed detection and false alarm respectively, for a query q and threshold θ. The weight factor β is defined as

$$\beta = \frac{C_{fa}(1 - P_{target})}{C_{miss} P_{target}} \tag{27}$$

$C_{fa}$and C_{miss} are the costs of false alarm and missed detection errors whereas P_{target} is the prior probability of a target trial. For SWS2013, the values of these parameters were set to $P_{target} = 0.00015$, $C_{fa} = 1$, and $C_{miss} = 100$, leading to a value of β equal to 66.66.

The miss error rate at a given threshold θ for a query q is given by

$$P_{miss}(q, \theta) = \frac{N_{miss}(q, \theta)}{N_{act}(q)}, \tag{28}$$

where $N_{miss}(q, \theta)$ is the number of miss errors for query q at a threshold θ, and $N_{act}(q)$ is the total number of occurrences of query q in the data.

While the number of occurrences $N_{act}(q)$ for a query in the test data is known, the calculation of false alarm rate is not straightforward since the total number of occurrences for every utterance is not known for the database. Instead, it is estimated that there are n_{tps} utterances for every second of speech in the test data. The value of n_{tps} for the SWS 2013 database is set to 1. The false alarm error rate $N_{nt}(q)$ for a query is thus estimated using this following equation:

$$N_{nt}(q) = N - N_{act}(q) = n_{tps} \cdot T_{audio} - N_{act}(q). \tag{29}$$

The false alarm error rate is then calculated as

$$P_{fa}(q, \theta) = N_{fa}(q, \theta) N_{nt}(q). \tag{30}$$

The number of false alarm errors corresponding to query q and threshold θ is denoted by $N_{fa}(q, \theta)$.

The average error rates over the entire set of queries Q can then be calculated as

$$P_{miss}(\theta) = \frac{1}{\mid Q \mid} \sum_{\forall q \in Q} P_{miss}(q, \theta), \tag{31}$$

$$P_{fa}(\theta) = \frac{1}{\mid Q \mid} \sum_{\forall q \in Q} P_{fa}(q, \theta). \tag{32}$$

It can be seen that the calculation of TWV, similar to the calculation of average precision for JHU word collection, uses a single threshold θ across the trials.

In addition to the above precision measure, the participants of SWS 2013 were also required to submit performance measures for their systems. Average memory consumption and real-time factor (RTF) were calculated to measure the performance of the submitted systems..

Real-time factor is a measure of speed of an automatic speech recognition systems and is equal to the total time taken by a speech recognition system to process a body of speech divided by the total duration of the speech contained in the database. Let A denote the total processing time of a system for the entire set of queries, and B be equal to the total duration of the reference database with the same unit of time used to measure both quantities. Then for each query, the entire reference database B is processed by the system. Given that the total number of queries is C, the total duration of speech processed by the speech recognition system is $B \times C$, leading to the following equation for the calculation of RTF:

$$RTF = \frac{A}{B \times C}. \tag{33}$$

For the SWS 2013 database, $B = 71,839$ seconds, C is equal to 684 seconds for the development set and 696 for the queries in the evaluation set.

6.3.3 Results

The SWS 2013 database is different from all the other speech corpora that have been used so far in this research. Speech in this database belongs to a large number of different languages, and recorded in a variety of acoustic environments, ranging from telephone conversations to studio recordings.

Due to the heterogeneous nature of the database, a large number of possible configurations of the EHMM were tested on the database to identify the optimal parameters for the task of keyword spotting.

EHMMs with different number of states were trained and tested for the task of keyword spotting. Additionally, the relationship between the level of training, i.e., the number of iterations of Baum-Welch algorithm and the precision of the VQbE system was studied. Finally, the impact of the clustering on VQbE system was also investigated, with the system precisions calculated for a number of different clustered EHMMs.

6.3.3.1 Precision and Level of Training

One way to assess the ability of an EHMM to model an observation sequence is to calculate $\Pr(O|\lambda)$, the likelihood of observations to be emitted from the model. It has been reported in Section 2.1.1 that the average probability of observation frames given the model increased consistently with an increase in the number of iterations of Baum-Welch algorithm (Table 1).

In these experiments, the effects of number of iterations of the training algorithm on the speech modeling abilities of an EHMM were investigated with a focus on the precision of the corresponding VQbE system. A 256-state EHMM, at different levels of training, was used for data representation and keyword model generation. Only left-right HMMs of queries were used for these experiments. The maximum term-weighted value (MTWV) observed for queries in the development set at different iterations of the EHMM are given in Table 5.

Table 5: Precision of EHMM after different level of training

Number of Iterations	10	13	14	15	16	17	20
MTWV	0.0955	0.0908	0.0991	0.0988	0.0913	0.088	0.0785

Unlike Pr($O|\lambda$) which consistently increases with the number of iterations of the training algorithm, TWV for the VQbE system increases for a initial period, and after reaching a peak value, the precision of the system actually declines if the training algorithm is not halted. For SWS 2013, this point was reached at 14 iterations of the Baum-Welch algorithm. This pattern in TWV values would suggest that there is an initial phase during which the model is improved with training, gaining more information about the observation distributions and their temporal dynamics. Continuing the training beyond a certain point, however, results in overfitting and over-generalization. The model parameters, transition probabilities more likely than observation densities, are fitted to the majority of the observations, leading to higher Pr($O|\lambda$) but at the cost of loss of information about the less frequently encountered speech segments.

6.3.3.2 Optimal Number of States for Keyword Spotting

The optimal number of states in an EHMM is a function of acoustic variety in the training data and may differ from one language to another. A small number of states may not fully capture the characteristics of the modeled data. Conversely, if the number of states in the EHMM is larger than that required then only a subset of the states will be used for modeling.

For the SWS2013 data, a number of different EHMMs were trained and tested for keyword spotting. The larger EHMMS, with number of states greater than 192, lost some of the states during training. For these EHMMs, there would be no state transitions into or out of these states during training, an indication that training data is being modeled with a smaller subset of states of the EHMM. For this reason, the maximum size of the EHMMs was limited to 256 states. The MTWVs of VQbE systems based on these EHMMs are given in Table 6.

Table 6: MTWV for the development set as a function of number of states in the model

States	64	128	192	256
MTWV	0.0565	0.0830	0.0868	0.0933

The maximum value of TWV was observed for the EHMM with 256 states (Table 6). This result was particularly unexpected given the elimination of states during training for the larger EHMMs. These experiments should have been repeated on a wider range of EHMM sizes since the MTWV maxima has been observed for the largest EHMM in the test set. However, the new EHMMs could not be trained and tested due to the time constraints imposed by the SWS 2013 submission dateline.

6.3.3.3 Clustering the States of EHMM

It has been observed in previous experiments that the precision of an EHMM-based keyword spotting system is improved when perceptually similar states are grouped together in the form of a superstate or as a cluster of states.

A graphical keyword model can be constructed from a single utterance by incorporating knowledge of similar states of the EHMM, which can either be obtained through supervised or unsupervised clustering schemes.

The SWS 2013 database is provided without any additional linguistics resources, and hence, the states in the model can only be grouped in an unsupervised manner. Starting from the pair of states with the smallest distance between them, states in the 256-state EHMM were merged in succession to form a new model with smaller number of clusters. The MTWVs for VQbE systems based on different EHMMs are listed in Table 7.

Table 7: MTWV for the development set as a function of number of clusters in the model

Clusters	128	160	192	224	256
MTWV	0.0849	0.1026	0.1033	0.102	0.0993

The EHMM with 192 superstates yielded the highest value of MTWV for the systems trained in this work. Additional state mergers led to a decrease in the MTWV of the systems. The relationship of precision and number of clusters in the EHMM is surprisingly similar to those observed for the isolated word-pair recognition task discussed in Section 6.2 and the experiments in Section 6.1.3.

In these experiments, the new EHMMs, obtained by clustering similar states of 256-state EHMM, consistently yielded a higher precision than EHMMs of similar sizes trained directly using the Baum-Welch algorithm. For comparison, the MTWV for EHMM with 192 clusters is 19% higher than that of the 192-state EHMM trained directly using the Baum-Welch algorithm.

6.3.3.4 Score Normalization

The histograms of log-likelihood scores of candidate utterances from two different trials, are shown in Figure 26. The two queries selected for this example have significantly different lengths. Even though the log-likelihood scores for each trial were length-normalized (LNorm) by dividing the total log-likelihood score of an observation sequence by its length, the histograms in Figure 26 show significant differences in candidate scores between the trials. In fact, the score distributions are non-overlapping with the best candidate for one trial scoring less than the lowest scoring candidate for the second trial.

This distribution of scores is particularly problematic for the calculation of TWV. The evaluation metric is calculated on a common threshold θ, as discussed in Section 6.3.2. In the case of trials illustrated in Figure 26, even if the individual TWVs have the maximum possible value of 1, the overall TWV would be much lower since the scales have not been scaled properly.

Another indication that length-normalization is not adequately scaling the scores, comes from the shape of the DET curve [56] in Figure 27. This DET curve has been calculated for the entire set of queries in the development set. In a binary classification task, the distribution of scores of each of the classes tend to be Gaussian distributed with different mean

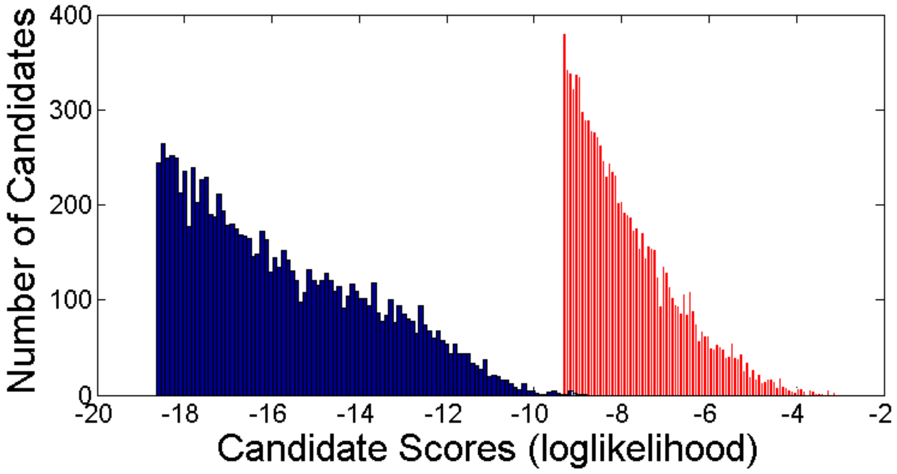


Figure 26: Histogram of length-normalized scores for two different trials

values and variances. DET curves takes advantage of this distribution of the elements of the classes and using the probit function converts produces linear tradeoff curves.

It is immediately obvious from Figure 27 that the DET curve is not linear, a clear indication that the candidate scores of true positives and false positives over the entire set of queries do not have a Gaussian distribution. This fact can be attributed to the differences in mean value and variance of individual trials that would lead to an overall distribution that is not Gaussian. Another consequence of this issue is a calculated TWV, 0.02 in this case, which is much lower than the TWV of a system with proper score normalization.

Score normalization is a prerequisite for any VQbE system when the precision measurements involve an absolute threshold θ. However, it is particularly challenging for a system that uses log-likelihoods due the large range of variations involved. In this work, two alternate schemes for score normalization were investigated. Zero Normalization (ZNorm) [59] rescales the score distribution to a zero mean and unit variance distribution. In order to do so, the rescaling parameters are estimated using the entire set of candidates, true positives as well as false alarms.

In Test Normalization (TNorm) [60], on the other hand, only the “imposter” scores are used to estimate the scaling parameters. The exact number of false positives in a trial

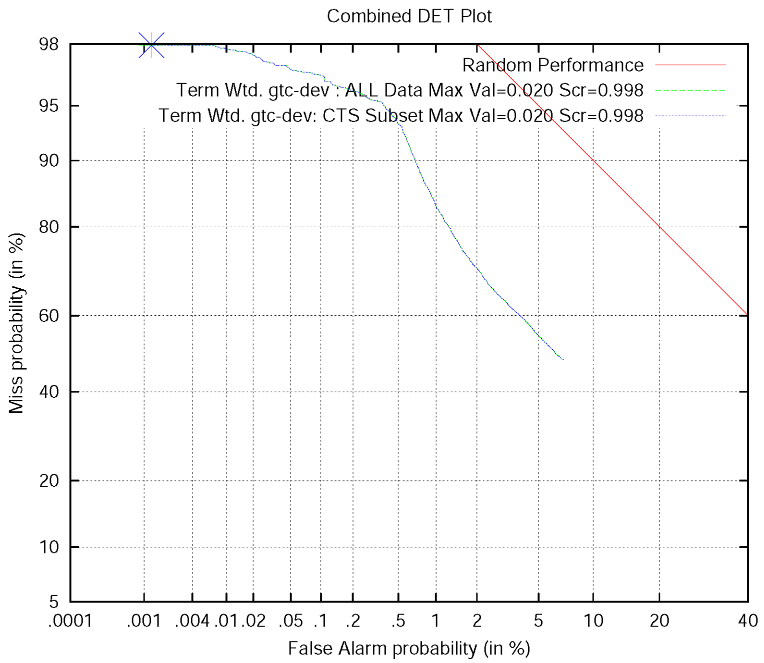


Figure 27: TWV for length-normalized scores

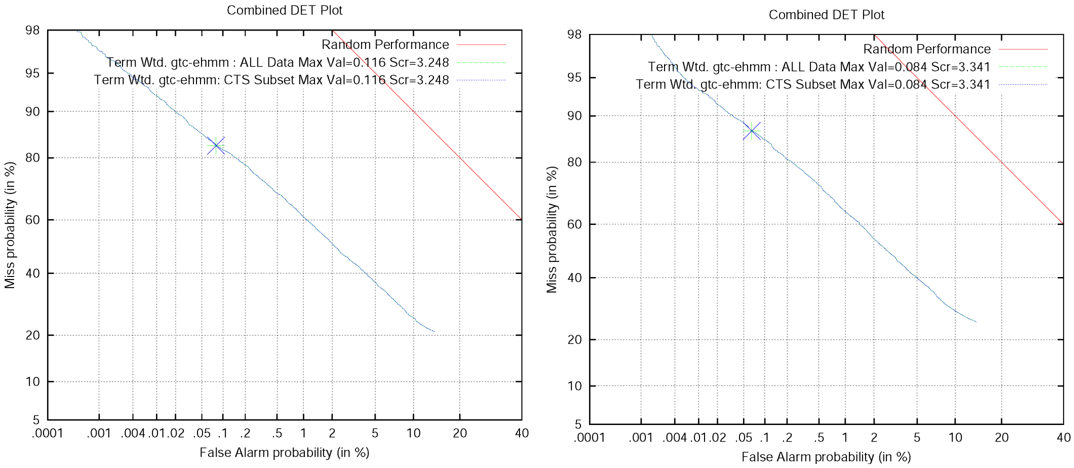
Table 8: Calculated TWV for a single trial using different score normalization schemes.

Scheme	LNorm	ZNorm	TNorm
TWV	0.02	0.1033	0.116

is not known but false positives greatly outnumber the true positives in a typical trial. By excluding a small fraction of highest scoring candidates, it is essentially guaranteed that the remaining candidates will be dominated by the imposters or false candidates. The fraction of candidates that has to be excluded can be chosen by trial and error. In these experiments, it was observed that the calculated TWV was maximized when the scaling parameters were estimated after excluding the top 90 candidates.

The value of TWV for the same set of results but different length normalization schemes is given in Figure 8.

The estimation of scaling parameters is straightforward in the case of Zero normalization. The TWV calculated in this manner, however, is noticeably lower than TWV calculated after TNorm score normalization. Consequently, most of the trials, including the scores given in Tables 6 and 7 were normalized using ZNorm and the scores for the final



(a) *Development Set* **(b)** *Evaluation Set*

Figure 28: Maximum Term Weighted Value for Development and Evaluation Sets.

submissions (Table 11) were normalized using TNorm.

The TNorm scheme requires a number of trials to estimate the optimal fraction of candidates for mean and variance estimates. Furthermore, these scaling parameters can only be estimated for the development set due to the availability of ground truth files. The same scaling parameters were then used for score normalization for the trials carried out on the evaluation set, with the assumption that score distributions are similar for both the query sets.

6.3.4 Final Term Weighted Value for MediaEval SWS 2013

The DET curves and maximum term-weighted values of an EHMM-based VQbE system (participated in SWS 2013 as Georgia Tech, Team Clements (GTC)) for the development and evaluation sets are shown in Figure 28. The system scored an MTWV of 0.116 for the development queries and 0.0838 for the set of evaluation queries [61].

The systems participating in SWS 2013 can be broadly classified into two categories. Zero-resource systems, as their name indicate, have been developed using only the resources provided in the SWS 2013 database. Open systems, on the other hand, are systems that have taken advantage of external linguistic resources, in addition to the SWS 2013

Table 9: MTWVs for zero-resource systems submitted to MediaEval 2013.

System	Dev.		Eval.		Real-time Factor
	MTWV	ATWV	MTWV	ATWV	
CUHK	0.367	0.367	0.307	0.304	0.018
ELiRF	0.1699	0.1697	0.1593	0.1591	0.003
GTC	0.116	0.1156	0.0838	0.082	0.003
Telefonica	0.1158	0.0961	0.0925	0.0793	0.003
LIA	-	0.0045	-	-0.0013	0.00013
UNIZA	0	-0.091	0.001	-0.027	0.017
TUKE	-	-0.1371	-	-0.1372	0.0048

database. These additional resources may include time-aligned labeled training data from other speech corpora. These semi-supervised systems generally outperform their zero-resource counterparts.

The precisions of zero-resource system along with their real-time factors, are listed in Table 9.

It is immediately clear from the values in Table 9 that the systems based on static clustering schemes, such as k-means and GMM, are not very effective for the task of keyword spotting on a complex database such as SWS 2013.

The TUKE system [62] compressed the feature space of speech signal using PCA segmentation and k-mean clustering, followed by GMM training on these clusters. Keyword search was performed using Segmental DTW. The UNIZA system [63] is designed on a similar principle and models the speech signal with 257 clusters obtained through k-mean clustering. A segmental 1-state HMM is then used to model 0.1 sec long segments of speech. Finally, these segment models are again grouped into 207 clusters using k-means clustering.

The LIA system, described in [64], uses I-Vectors for the task of keyword spotting. I-Vectors have been successfully used for language recognition [65] and speaker driarization [66] previously. The LIA system, however, suffered from a unusually large number of false positives that consisted of silence. The authors postulated the lack of a voice activity detector to be responsible for the contamination of the candidate list by these false positive and hence, the low value of MTWV.

Another GMM-based system that achieved relatively high precision was the Telefonica [67] spoken web search engine. The Telefonica research group has been participating in this challenge since its inception. Their system is a well-designed with a number of unique features. An initial GMM clustering of speech is followed by a VTLN-based speaker normalization scheme. The VTLN parameters are then used to train a new background model that is more resistant to speaker-dependent variations. The system also implements a voice activity detector to minimize the loss of precision due to non-speech segments. Keyword search is carried out using a proprietary two-pass DTW-based algorithm known as Information Retrieval DTW [68, 69].

The ELiRF system [70] is a straightforward template-matching system that uses MFCC features with Subsequence DTW [71], to achieve surprisingly high values of MTWV considering the low complexity of the system. The system, however, implemented a voice activity detector and performed post-processing steps to remove false positives from the list of candidates. The use of cosine distance and effective score normalization may have also contributed to the precision of the system.

The highest precision in the zero-resource category was achieved by the CUHK System [72] again this year. The CUHK 2012 [73] spoken web search system was a massive system with five subsystems. Three of these subsystems were tokenizers in the split temporal context network structure and based on Czech, Hungarian, Russian, English and Mandarin phoneme recognizers. The two unsupervised tokenizers were based on GMM and Acoustic Segment Modeling (ASM). The tokenizers were used to generate five streams of posteriorgrams, and a DTW matrix combination approach was used to aggregate the scores from individual subsystems.

Th CUHK SWS 2013 system uses Gaussian component clustering [74], an extension of the zero-resource GMM subsystem presented in [73]. MFCC features with delta and

Table 10: MTWV of open systems submitted to MediaEval 2013

System	Dev.		Eval.		Real-time Factor
	MTWV	ATWV	MTWV	ATWV	
BUT	0.4373	-	0.3776	-	0.18
GTTS	0.4174	0.4078	0.3992	0.3806	0.24
L2F	0.3905	0.3883	0.342	0.3376	77.33
CMTECH	0.2685	0.2683	0.2623	0.2619	0.0056
IIIT	0.2765	-	0.2413	-	0.0017
SpeeD	0.068	-	0.058	-	0.00006

delta-delta coefficients were preprocessed with a voice activity detector and VTLN to minimize speaker variations. A GMM with 4096 components was trained, and unsupervised phoneme segmentation was performed on the GMM-based representation of speech. For each of these segments, a Gaussian posterior vector is then calculated by averaging the posteriorgram of individual frames in the segment. At this point, speech is represented by a Gaussian-by-segment data matrix X. The similarity matrix XX^T is decomposed into 150 clusters using spectral clustering. Keyword search was performed using DTW on the posteriorgrams of the final 150 clusters.

The open systems have a completely different philosophy from the systems presented so far. These systems try to leverage existing linguistic resources by using phoneme models or acoustic priors trained in supervised fashion and adapt them to the SWS2013 database.

The SpeeD system [75] is an automatic speech recognition system trained on 170 million utterances in Romanian language. The precision of the system indicates that the 26 phoneme models in Romanian language are inadequate for modeling the speech in the SWS 2013 database.

The IIIT system [76] is again based on a single language, Telugu in this case. This system used a combination of phone models and Multilayer perceptrons (MLP) trained on 23 articulatory features. Bottle-neck features were extracted from speech parameters and used to train a GMM with 128 components. Keyword search was performed using DTW on these posteriorgrams.

The CMTECH system described in [77] is based on the fusion of two different models. The spectral model, composed of a GMM, is trained on the TIMIT corpus and then adapted to the target database. The temporal models were again trained using the GMM features, initially on SWS 2012 data and migrated to the current data, using a novel approach [78].

The open systems discussed so far, are based on acoustic models trained on a single high-resource language. These models are either applied directly or adapted to the resource-limited set of languages in SWS 2013. The precision observed for these systems is also dependent on the level of similarity among the languages in the training and test databases. Compared to the 26 phonemes in Romanian language, Telugu has 49 different phonemes which could have contributed to the higher precision of the IIIT system. Hence, the precision of a system can be increased further by extending the repository of acoustic models borrowed from resource-rich languages. This is exactly the approach adopted by the systems presented next in this section.

The remaining systems presented in this section [79][80][81], are all trained on a large number of languages and the trained acoustic models were then migrated to the test database. As expected, these systems have the highest precision of any systems presented so far.

The L2F system in [81], for example, consists of six individual sub-systems based on phonetic classifiers of European Portuguese, Brazilian Portuguese, and European Spanish. The GTTS [80] and BUT [79] systems are both based on the Brno University of Technology phoneme models. The GTTS system is based on phone models derived from 3 different languages, i.e. , Czech, Hungarian and Russian. The BUT "Massive Parallel Approach," true to its name, consists of 26 sub-systems, which are trained on a number of different languages including Czech, Hungarian, Russian, Cantonese, Pashto, Tagalog, Turkish, English, German, Portuguese, Spanish, and Vietnamese. The BUT system, unsurprisingly, had the highest value of MTWV of any system submitted to SWS 2013.

6.3.4.1 Keyword Model using Multiple Instances of a Query

The development and evaluation query sets in SWS 2013, each contain more than 500 unique utterances. The evaluation of a system on this basic set of utterances constituted the mandatory part of the SWS 2013 task.

The SWS 2013 task also defined a set of optional or extended trials to study the precision of VQbE systems that can utilize multiple examples of an utterance. For this purpose, the development and evaluation sets also contained additional samples for some of the queries. Specifically, 194 out of a total of 505 queries in the development set and 188 of the 501 queries in the evaluation set have more than one examples available.

In these experiments, a graphical keyword model for a query was constructed from a number of examples of a query. In order to study the relationship between precision and the number of available examples of a query, a number of different GKM were developed with different number of sample utterances.

The scoring script provided by the organizers, however, expected the complete list of queries in order to calculate the MTWV of the system. Consequently, the scores given in Table 11 were calculated over the complete set of utterances, with only the queries with multiple examples altered for different trials.

The precisions for the systems investigated in these experiments, grouped by the number of examples used in multi-sample keyword model, is given in Table 11. Since the precisions vary for a subset of the total queries while the MTWV is calculated over the entire set of queries, the actual gain in precision due to the use of multiple examples would be higher than that given in Table 11.

Table 11: MTWV for the development set in extended trials.

Instances	2	3	5	7	9
MTWV	0.1259	0.1219	0.1151	0.1096	0.1081

Surprisingly, the highest gain in MTWV was observed after the addition of the second example and the precision decreased consistently (Table 11) after the inclusion of each additional example.

This pattern is similar to that observed for EHMMs constructed by unsupervised clustering schemes. For those systems, precision initially increased and after reaching an epoch, further state mergers led to a decline in system precision. It was postulated that the states merger later in the clustering were not as good a match as the earlier state merger. These "bad" state mergers were considered to be the primary factor responsible for the drop in precision.

However, all the states added to graphical keyword model in the current scheme, are from the true instances of the query. This degradation of the system precision by the addition of these "good" states indicate that factors other than "bad" mergers are contributing to a decline in precision.

In order to understand this behavior, we must consider the composition of the test data. In 20 hours of speech, even with a conservative estimate of $n_{tps} = 1$ (1 term per second), there are more than 72 thousand utterances. On average, there are only 11 instances of a particular query from the development set in 20 hours of test data. Only a fraction of these true positives would probably benefit from the addition of a state to the graphical keyword model. On the opposite end, even if a very small fraction of the false positives enter the model space (the set of states that constitute the keyword model) of a query then the number of new false positives would greatly outnumber the newly discovered true positives.

CHAPTER 7

CONCLUSIONS

Automatic speech recognition systems have become an essential part of our everyday life. These systems are mature, have very high precisions and have gained wide acceptance as a human-machine interface on mobile platforms.

The development of an automatic speech recognition system is an expensive endeavor, reliant on the availability of tens or even hundreds of hours of time-aligned labeled speech. Speech recognitions systems have been developed for the major languages of the world, catering to the needs of the majority of world's population. There are more than seven thousand different languages spoken around the world. For majority of these languages, the development of a speech recognition system is either commercially inviable or simply impossible due to a lack of trained linguistics.

An EHMM can be useful in extracting underlying structure embedded in connected speech without the need for a time-aligned transcribed corpus. This technique would be useful for languages with moderate amounts of waveform data, but no text renderings. This approach to statistically model speech offers a good mix of straightforward training with the generalization powers of traditional statistical modeling techniques. It is based on the same mathematical formulation that is the foundation of most state-of-the-art speech recognition systems.

An EHMM can model non-linear observation distributions using a number of states with Gaussian observation distributions. The training process takes into account the temporal structure of speech to group observations with similar form and dynamics within a single model state, an ability absent in most static, proximity-based clustering schemes such as k-means and GMM.

In contrast to supervised statistical modeling methods, an EHMM can be trained from unlabeled speech. As a result, the vast majority of the recorded speech available on the web, radio, and television archives, can be used to train an EHMM. Due to the unsupervised nature of the training, an EHMM can be trained, with minimum effort, for any language that has a small body of recorded speech available. Another desirable feature of an EHMM is its adaptability. An existing EHMM can be easily adapted to another data set using the Baum-Welch algorithm. States in the original EHMM are influenced by and adapted to the closest matching observations in the training data, and both the spectral features and temporal behavior of the states are altered in the process.

An EHMM is an efficient model of speech. It greatly reduces the dimensionality of the problem by modeling speech with a finite number of states while preserving substantial amount of information contained within the original speech. It has been demonstrated that perfectly intelligible speech can be synthesized from a sequence of states of an EHMM. The distributions of fundamental frequencies for the different states of EHMM exhibit a consistent pattern that can be used to construct a pitch profile of different segments of speech. Consequently, an EHMM can be used to synthesize not only the speech spectrum but also its excitation signal.

In absence of any human supervision, an EHMM is reliant on the robustness of speech features to handle speaker-dependent variations. During preliminary investigations, it was observed that the most likely state sequences corresponding to multiple realizations of an utterance may vary significantly. This lack of invariance to contextual or speaker-dependent variations in speech degrades the precision of an EHMM-based VQbE system.

An EHMM-based VQbE system can greatly benefit from a more robust speech representation. In the case of speech features that are sensitive to contextual or speaker-dependent variations, perceptually similar speech segments may not be localized in a particular region, making the overall distribution of observations from a single perceptual class highly non-linear. Due to a dependence on Mahalanobis distances to group observations, a single state of an EHMM would represent observations that are in close proximity to each other in the feature space. Consequently, the observation distribution for a single phoneme may span more than one states of the EHMM.

The one-to-many nature of the mappings between phonemes and states of an EHMM greatly complicates the task of keyword modeling. Different realizations of an utterance may correspond to different sequences of EHMM states. One way to integrate the different acoustic realizations of a keyword in a single model is to merge all the states corresponding to a single phoneme into a new superstate. This approach can accurately model the observation density of a phoneme with a mixture of Gaussians. However, valuable information about the individual state transitions will be lost in this process.

An alternate scheme to incorporate the one-to-many phoneme-to-state mappings in a keyword model is by treating these clusters as a logical construct. A single unit of speech, in this approach, would be modeled simultaneously by a number of states, each state corresponding to one of its possible acoustic manifestations. The set of EHMM states corresponding to an utterance will constitute a substructure on the EHMM. This collection of states can be represented with a graph, with states constituting the nodes and state transition probabilities defining the connectivity between the nodes. This graphical keyword model has the ability to accurately model the spectral and temporal characteristics of the model states.

In this research, a novel variant of Viterbi algorithm has been developed for keyword search using the graphical keyword model of a query. In this algorithm, the trellis has a 3-dimensional structure, with a slice of trellis, at any given instant of time, reflecting the structure of the corresponding graphical keyword model. The computational complexity of the 3D Viterbi algorithm is greatly increased by the large number of possible states transitions. There can be a number of initiating and terminating nodes in the trellis with multiple paths connecting each pair of terminal nodes. Tremendous speed improvements can be achieved by a modification of the optimality criteria. Integrating the length of a candidate path into the optimality criteria permits comparison of candidate paths initiating at different instants of time. This modification in the search algorithm leads to an implementation of the VQbE system with a real-time factor comparable to and in some cases, exceeding the state-of-the-art DTW-based algorithms.

The mappings between phonemes and states of an EHMM can be discovered in a number of ways. The correlations between EHMM states and phonemes can be mapped by leveraging existing linguistic resources in a supervised manner. Conversely, these mappings can be discovered automatically by taking advantage of the acoustic properties of speech.

In this research, two different schemes for the identification of similar states have been investigated. The first method examined the role of proximity in cepstral domain as a similarity measure and grouped together states in close proximity to each other using a K-nearest neighbor clustering approach. Due to the non-linear density and overlapping distributions of observations for different phonemes, this scheme had limited success in improving the precision of a EHMM-based VQbE system.

The second approach compared the spectral features of observations from different states for the discovery of similar states. A feature-independent scheme, state mean values were transformed back to linear frequency domain, where the corresponding spectral features were optimally aligned through a warping of the frequency axis. The dynamic frequency warping reduces the mean square error between the two spectra by aligning the corresponding formants within a user-configurable range of frequencies.

The DFW-based clustering of states proved to be a promising approach, leading to a significant increase in precision in experiments conducted on a number of speech corpora. In these experiments, graphical keyword models, constructed from states clusters obtained through DFW-clustering, led to noticeably higher system precisions compared to left-right HMMs and unsupervised GMM-based systems.

There a number of unresolved issues in the current approach to EHMM-based VQbE systems. One of these issues is the variation in scale of candidate scores from one trial to another. The log-likelihood scores of candidates vary non-linearly with the length of a observation sequence. While these characteristics of score distribution do not affect the precision of individual trials, it can have a negative impact on the overall calculated precision of the system when an absolute value of threshold is required.

It has been observed in the experiments that increasing the initial size of the EHMMs led to higher system precisions. Yet, in the flat-start approach to training, some states in the larger EHMMs become redundant during training. Although, this trend of increasing precisions with larger EHMMs cannot be expected to sustain due to the fragmentation of phonemes into larger number of states, an open area of research is the development of alternate training initialization schemes that may lead to the training of large EHMMs without encountering the state redundancy problem.

The optimal number of clusters that lead to the highest precision must be calculated by trial and error. Ideally, a perfect clustering schemes would discover the perceptual classes in speech, with the total number of clusters equal to the number of phonemes in a language. However, the optimal number of clusters, with regards to precision, was found to be much higher than this value. System precision peaked for EHMM with 192 clusters but gradually declined after this peak value. While this behavior can be partly attributed to bad mergers of phonetically different states, surprisingly, a similar behavior was observed for keyword models constructed with actual instances of a keyword, an indication that this loss of precision may have more than one source.

The length composition of states in a model are enforced implicitly by the state transition probabilities. These transition probabilities have a smaller contribution to the overall likelihood of an observation sequence, with the final sum dominated by the emission probabilities. The loss of precision may have likely been caused by the weak structure imposed on the durations of states by a graphical keyword model, allowing a large number of partially matching observations to be classified as positives. Hence, a solution to these issues would require fundamental changes to the keyword modeling scheme and subsequently, to the Viterbi algorithm.

The higher precisions observed for the EHMM-based system over static GMM-based system is a testament to the superiority of clustering based on a combination of similarity in form and behavior over static clustering schemes. However, both the unsupervised clustering schemes presented in this work are static in nature, with states compared and clustered using spectral or cepstral features only. However, phonemes are complex acoustic structures with spans much longer than a single state of EHMM. In order to discover the phonemic or morphemic structure of speech, correlations must be considered over a longer duration of time than that allowed by the first order Markovian assumption underlying an EHMM. An EHMM-based VQbE system would greatly benefit from a clustering scheme that would group states with similar spectral and temporal characteristics.

In this thesis, the use of an ergodic hidden Markov model of speech for the task of query-by-example spoken term detection has been investigated. The characteristics of EHMM states have been studied both in time and frequency domains. The effects of speaker-dependent and context-induced variation on the EHMM-based representation of speech have been studied and used to devise schemes to minimize these variations. Two different schemes for the identification of perceptually similar states of the EHMM have been presented. The clustering of states of the EHMM using dynamic frequency warping led to significant gains in system precision. A search framework, consisting of a keyword modeling scheme and a modified Viterbi algorithm, has also been implemented. The EHMM-based VQbE system has been demonstrated to attain higher precision than systems of similar complexity based on static clustering schemes.

REFERENCES

[1] S. Santosh Kumar and V. Ramasubramanian, "Automatic language identification using ergodic HMM," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, vol. 1, 18-23, 2005, pp. 609 – 612.

[2] R. R. Rao and A. Prasad, "Glottal excitation feature based gender identification system using ergodic HMM," *International Journal of Computer Applications*, vol. 17, no. 3, pp. 31–36, March 2011, published by Foundation of Computer Science.

[3] M. Lee, A. Durey, E. Moore, and M. Clements, "Ultra low bit rate speech coding using an ergodic hidden Markov model," in *ICASSP 2005*, vol. 1, 18-23, 2005, pp. 765 – 768.

[4] P. Li, J. Liang, and B. Xu, "A novel instance matching based unsupervised keyword spotting system," in *Innovative Computing, Information and Control, 2007. ICICIC '07. Second International Conference on*, September 2007, p. 550.

[5] M. Mokhtar and A. El-Abddin, "A model for the acoustic phonetic structure of arabic language using a single ergodic hidden Markov model," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 1, October 1996, pp. 330 –333 vol.1.

[6] G. Fant, "A note on vocal tract size factors and non-uniform F-pattern scalings," *STL-QPSR*, vol. 7, pp. 22–30, 1966.

[7] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 23, no. 1, pp. 67–72, February 1975.

[8] L. Rabiner, S. Levinson, A. Rosenberg, and J. Wilpon, “Speaker-independent recognition of isolated words using clustering techniques,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 27, no. 4, pp. 336–349, August 1979.

[9] H. Sakoe, “Two-level DP-matching–a dynamic programming-based pattern matching algorithm for connected word recognition,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 27, no. 6, pp. 588–595, December 1979.

[10] T. Vintsyuk, “Element-wise recognition of continuous speech composed of words from a specified dictionary,” *Cybernetics*, vol. 7, pp. 361–372, 1971.

[11] J. Bridle, M. Brown, and R. Chamberlain, “An algorithm for connected word recognition,” in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP ’82.*, vol. 7, May 1982, pp. 899 – 902.

[12] H. Ney, “The use of a one-stage dynamic programming algorithm for connected word recognition,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 32, no. 2, pp. 263–271, April 1984.

[13] C.-H. Lee, F. K. Soong, and B.-H. Juang, “A segment model based approach to speech recognition,” in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*. IEEE, 1988, pp. 501–541.

[14] K. Paliwal, “Lexicon-building methods for an acoustic sub-word based speech recognizer,” in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, 1990, pp. 729–732 vol.2.

[15] T. Fukada, M. Bacchiani, K. K. Paliwal, and Y. Sagisaka, “Speech recognition based on acoustically derived segment units,” in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 2. IEEE, 1996, pp. 1077–1080.

[16] M. Bacchiani and M. Ostendorf, "Using automatically-derived acoustic sub-word units in large vocabulary speech recognition." in *ICSLP*, 1998.

[17] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 1, pp. 186–197, 2008.

[18] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," *Proceedings - Automatic Speech Recognition and Understanding*, 2009.

[19] ——, "Towards multi-speaker unsupervised speech pattern discovery," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4366–4369.

[20] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The annals of mathematical statistics*, vol. 41, pp. 164–171, 1970.

[21] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 5, no. 2, pp. 179–190, 1983.

[22] J. Gauvain and C. Lee, "MAP estimation of continuous density HMM: Theory and applications," in *In: Proceedings of DARPA Speech and Natural Language Workshop*. Morgan Kaufmann, 1992, pp. 185–190.

[23] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *ICASSP 1986*, vol. 11, April 1986, pp. 49 – 52.

[24] Y. Ephraim, A. Dembo, and L. R. Rabiner, “A minimum discrimination information approach for hidden Markov modeling,” *IEEE Transactions on Information Theory*, pp. 1001–1013, 1989.

[25] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[26] R. Schwartz, Y. Chow, S. Roucos, M. Krasner, and J. Makhoul, “Improved hidden Markov modeling of phonemes for continuous speech recognition,” in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP ’84.*, vol. 9, March 1984, pp. 21 – 24.

[27] C.-H. Lee, L. R. Rabiner, R. Pieraccini, and J. G. Wilpon, “Acoustic modeling of subword units for speech recognition,” in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. IEEE, 1990, pp. 721–724.

[28] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner, and J. Makhoul, “Context-dependent modeling for acoustic-phonetic recognition of continuous speech,” in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP 1985*, vol. 10, April 1985, pp. 1205 – 1208.

[29] P. Cardillo, M. Clements, and M. Miller, “Phonetic searching vs. LVCSR: How to find what you really want in audio archives,” *International Journal of Speech Technology*, vol. 5, no. 1, pp. 9–22, 2002.

[30] T. Hazen, W. Shen, and C. White, “Query-by-example spoken term detection using phonetic posteriorgram templates,” *Proceedings - Automatic Speech Recognition and Understanding*, 2009.

[31] W. Shen, C. M. White, and T. J. Hazen, “A comparison of query-by-example methods for spoken term detection,” in *INTERSPEECH*, 2009, pp. 2143–2146.

[32] L. Lamel, J. Gauvain, and G. Adda, “Lightly supervised and unsupervised acoustic model training,” *Computer Speech and Language*, vol. 16, pp. 115–129, 2002.

[33] S. Novotney, R. Schwartz, and J. Ma, “Unsupervised acoustic and language model training with small amounts of labelled data,” in *ICASSP 2009*, April 2009, pp. 4297 –4300.

[34] A. Garcia and H. Gish, “Keyword spotting of arbitrary words using minimal speech resources,” in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, May 2006, p. I.

[35] M. Huijbregts, M. McLaren, and D. A. van Leeuwen, “Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection,” in *ICASSP*, 2011, pp. 4436–4439.

[36] E. Charniak, D. Blaheta, N. Ge, K. Hall, J. Hale, and M. Johnson, “BLLIP 1987-89 WSJ Corpus Release 1, Linguistic Data Consortium, Philadelphia,” 2000. [Online]. Available: http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2000T43

[37] J. S. Garofalo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, “TIMIT acoustic-phonetic continuous speech corpus,” 1993.

[38] R. G. Leonard and G. Doddington, “TIDIGITS,Linguistic Data Consortium, Philadelphia,” 1993. [Online]. Available: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S10

[39] C. C. David, D. Miller, and K. Walker, “The Fisher corpus: a resource for the next generations of speech-to-text,” in *in Proceedings 4th International Conference on Language Resources and Evaluation*, 2004, pp. 69–71.

[40] A. Jansen, E. Dupoux, S. Goldwater, M. Johnson, S. Khudanpur, K. Church, N. Feldman, H. Hermansky, F. Metze, and R. Rose, “A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition,” in *Proceedings of ICASSP*, vol. 2013, 2013.

[41] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, “Rapid evaluation of speech representations for spoken term discovery.” in *INTERSPEECH*, 2011, pp. 821–824.

[42] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, “The HTK Book,” *Cambridge University Engineering Department*, 2002.

[43] S. Cha, “Comprehensive survey on distance/similarity measures between probability density functions,” *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 2, pp. 300–307, 2007.

[44] P. Boersma, “Praat, a system for doing phonetics by computer.” *Glot International*, vol. 5:9/10, pp. 341–345, 2001.

[45] L. V. Gool, T. Moons, E. Pauwels, and A. Oosterlinck, “Vision and Lie’s approach to invariance,” *Image and Vision Computing*, vol. 13, no. 4, pp. 259–277, 1995.

[46] A. Errity and J. McKenna, “An investigation of manifold learning for speech analysis.” in *INTERSPEECH*. Citeseer, 2006.

[47] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[48] L. K. Saul and S. T. Roweis, “Think globally, fit locally: unsupervised learning of low dimensional manifolds,” *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, Dec. 2003. [Online]. Available: http://dx.doi.org/10.1162/153244304322972667

[49] A. Jansen and P. Niyogi, “Intrinsic Fourier analysis on the manifold of speech sounds,” in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I.

[50] D. Pepper and M. Clements, “On the phonetic structure of a large hidden Markov model,” in *ICASSP 1991*, April 1991, pp. 465–468 vol.1.

[51] ——, “Phonemic recognition using a large hidden Markov model,” *Signal Processing, IEEE Transactions on*, vol. 40, no. 6, pp. 1590–1595, 1992.

[52] J. McDonough, T. Schaaf, and A. Waibel, “Speaker adaptation with all-pass transforms,” *Speech Communication*, vol. 42, no. 1, pp. 75–91, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639303001183

[53] S. Panchapagesan and A. Alwan, “Multi-parameter frequency warping for VTLN by gradient search,” in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, May 2006, pp. 1181–1184.

[54] M. Pitz and H. Ney, “Vocal tract normalization as linear transformation of MFCC,” in *INTERSPEECH’03*, 2003, pp. 1445–1448.

[55] J. Glass, T. J. Hazen, L. Hetherington, and C. Wang, “Analysis and processing of lecture audio data: Preliminary investigations,” in *Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004*. Association for Computational Linguistics, 2004, pp. 9–12.

[56] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, “The DET curve in assessment of detection task performance,” DTIC Document, Tech. Rep., 1997.

[57] A. Ali, S. Rustamov, and M. A. Clements, “Speech retrieval using a single ergodic hidden Markov model,” *AICT 2013*, forthcoming.

[58] C.-y. Lee and J. Glass, “A nonparametric bayesian approach to acoustic model discovery,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 40–49.

[59] D. A. Reynolds, “Comparison of background normalization methods for text-independent speaker verification.” in *Eurospeech*, 1997.

[60] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, “Score normalization for text-independent speaker verification systems,” *Digital Signal Processing*, vol. 10, no. 1, pp. 42–54, 2000.

[61] A. Ali and M. A. Clements, “Spoken web search using an ergodic hidden Markov model of speech,” in *MediaEval*, 2013.

[62] J. Vavrek, M. Pleva, M. Lojka, P. Viszlay, E. Kiktová, D. Hládek, and J. Juhár, “TUKE at MediaEval 2013 spoken web search task,” in *MediaEval*, 2013.

[63] R. Jarina, M. Kuba, R. Gubka, M. Chmulik, and M. Paralic, “UNIZA system for the spoken web search task at MediaEval2013,” in *MediaEval*, 2013.

[64] M. Bouallegue, G. Senay, M. Morchid, D. Matrouf, G. Linarès, and R. Dufour, “LIA @ MediaEval 2013 spoken web search task: An I-Vector based approach,” in *MediaEval*, 2013.

[65] D. Martınez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, “Language recognition in IVectors space,” *Proceedings of Interspeech, Florence, Italy*, pp. 861–864, 2011.

[66] J. Franco-Pedroso, I. Lopez-Moreno, D. T. Toledano, and J. Gonzalez-Rodriguez, "ATVS-UAM system description for the audio segmentation and speaker diarization albayzin 2010 evaluation," in *FALA VI Jornadas en Tecnologa del Habla and II Iberian SLTech Workshop*, 2010, pp. 415–418.

[67] X. Anguera, M. Skácel, V. Vorwerk, and J. Luque, "The Telefonica Research spoken web search system for MediaEval 2013," in *MediaEval*, 2013.

[68] X. Angeura, "Information retrieval-based dynamic time warping," in *Interspeech*, 2013.

[69] G. Mantena and X. Anguera, "Speed improvements to information retrieval-based dynamic time warping using hierarchical k-means clustering," in *ICASSP 2013*, 2013.

[70] J. A. Gómez, L.-F. Hurtado, M. Calvo, and E. Sanchis, "ELiRF at MediaEval 2013: Spoken web search task," in *MediaEval*, 2013.

[71] M. Müller, *Information retrieval for Music and Motion*. Springer, 2007.

[72] H. Wang and T. Lee, "The CUHK spoken web search system for MediaEval 2013," in *MediaEval*, 2013.

[73] ——, "The CUHK spoken web search system for MediaEval 2012," in *MediaEval*, 2012.

[74] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Unsupervised mining of acoustic subword units with segment-levelGaussian posteriorgrams," in *Interspeech*, 2013.

[75] A. Buzo, H. Cucu, and I. Molnar, "SpeeD @ MediaEval 2013: A phone recognition approach to spoken term detection," in *MediaEval*, 2013.

[76] G. Mantena and K. Prahallad, "IIIT-H SWS 2013: Gaussian posteriorgrams of bottleneck features for query-by-example spoken term detection," in *MediaEval*, 2013.

[77] C. Gracia, X. Angeura, and X. Binefa, “The CMTECH spoken web search system for MediaEval 2013,” in *MediaEval*, 2013.

[78] P. SCHWARZ, “Phoneme recognition based on long temporal context,” Ph.D. dissertation, BRNO UNIVERSITY OF TECHNOLOGY, 2008.

[79] I. Szöke, L. Burget, F. Grézl, and L. Onde, “BUT SWS 2013 - massive parallel approach,” in *MediaEval*, 2013.

[80] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, “GTTS systems for the SWS Task at MediaEval 2013,” in *MediaEval*, 2013.

[81] A. Abad, R. F. Astudillo, and I. Trancoso, “The L2F spoken web search system for MediaEval 2013,” in *MediaEval*, 2013.