Unsupervised Modeling of Latent Topics and Lexical Units in Speech Audio

by

David F. Harwath

B.S., University of Illinois at Urbana-Champaign (2010)

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

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author .....
Department of Electrical Engineering and Computer Science
May 22, 2013

Certified by .....
James R. Glass
Senior Research Scientist
Thesis Supervisor

Certified by .....
Timothy J. Hazen
Principal Scientist, Microsoft
Thesis Supervisor

Accepted by .....
Leslie Kolodziejski
Chairman, Department Committee on Graduate Students

Unsupervised Modeling of Latent Topics and Lexical Units in Speech Audio

by

David F. Harwath

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Zero-resource speech processing involves the automatic analysis of a collection of speech data in a completely unsupervised fashion without the benefit of any transcriptions or annotations of the data. In this thesis, we describe a zero-resource framework that automatically discovers important words, phrases and topical themes present in an audio corpus. This system employs a segmental dynamic time warping (S-DTW) algorithm for acoustic pattern discovery in conjunction with a probabilistic model which treats the topic and pseudo-word identity of each discovered pattern as hidden variables. By applying an Expectation-Maximization (EM) algorithm, our method estimates the latent probability distributions over the pseudo-words and topics associated with the discovered patterns. Using this information, we produce informative acoustic summaries of the dominant topical themes of the audio document collection.

Thesis Supervisor: James R. Glass
Title: Senior Research Scientist

Thesis Supervisor: Timothy J. Hazen
Title: Principal Scientist, Microsoft

Acknowledgments

I am beyond grateful to my advisors, Jim Glass and T.J. Hazen, for providing me with the resources and guidance that enabled me to complete this thesis. I am extremely lucky to have the opportunity to work with them.

My family - Nancy, Frank, and Amy Harwath - have provided me with endless love and support and are an enormous source of strength in my life. Thank you for always being there for me.

This work was sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Zero Resource Setting

Current state-of-the-art speech recognition (ASR) systems typically rely on statistical models that require both a large amount of language specific knowledge and a sizable collection of transcribed data. These resources are required for training statistical models that map acoustic observations to phonetic units, creating pronunciation dictionaries mapping phonetic units to words, and estimating language models to provide constraints on the possible sequences of words. But of the 7,000 human languages spoken across the globe, only 50 to 100 can actually support the massive infrastructure of annotated data and expert knowledge required to train state-of-the art speech recognition systems. Recently in the speech community, there has been a push towards developing increasingly unsupervised, data-driven systems which are less reliant on linguistic expertise. In [7], Glass places these paradigms along a spectrum (Figure 1-1) of speech processing scenarios. At one end lies the familiar ASR framework in which near-complete supervision is the norm, and opposing it is the so-called "sensor-based" or "zero resource" learning problem: spoken audio data is available in a specific language, but transcriptions, annotations and prior knowledge for this language are all unavailable. In this scenario completely unsupervised learning techniques are required to learn the properties of the language and build models that describe the spoken audio.

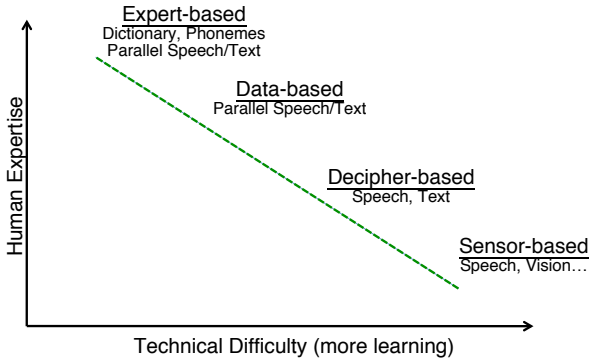

Figure 1-1: Potential ASR learning scenarios as described by [7] We use the terms "sensor-based" and "zero-resource" interchangeably.

In essence, the ultimate goal of zero-resource modeling is to develop completely unsupervised techniques that can learn the elements of a language's speech hierarchy solely from untranscribed audio data. This includes the set of acoustic phonetic units, the sub-word structures such as syllables and morphs, the lexical dictionaries of words and their pronunciations, as well as higher level information about the syntactic and semantic elements of the language. This is an extremely lofty goal, but recent research has begun to investigate solutions to sub-problems at various levels of the hierarchy.

One area of research focuses on the automatic discovery of repeated acoustic patterns in a spoken audio collection. The patterns that are found typically correspond to commonly repeated words or phrases observed in the data. Initial work in this area used a segmental dynamic time warping (S-DTW) algorithm search for repeated acoustic patterns in academic lectures [27]. Improvements to this approach were obtained when raw acoustic features were replaced with model-based posteriorgram features derived from a Gaussian mixture model [35]. Recent techniques for dramatically reducing the computational costs of the basic search have made this acoustic pattern discovery approach feasible on large corpora [18, 19, 20, 34].

Another approach is to first learn acoustic-phonetic models from the audio data. These phonetic units are then used to represent the data before performing any higher level pattern discovery. Approaches of this type include a self-organizing unit (SOU) recognition system which learns an inventory of phone-like acoustic units in an un-

supervised fashion [31], a successive state splitting hidden Markov model framework for discovering sub-word acoustic units [33], and a Bayesian nonparametric acoustic segmentation framework for unsupervised acoustic model discovery [25]. Clustered patterns from a spoken term discovery system have also been used to help unsupervised learning of acoustic models [16, 17].

Independent of the speech technology work being pursued in this area, researchers in linguistics and cognitive science have been interested in the process of language acquisition and have been developing techniques that attempt to learn words by segmenting a collection of phoneme strings. Bayesian approaches have proven to be especially successful for this task [8, 21].

The successful application of the aforementioned algorithms opens the doors for higher level semantic analysis. In [11], n-gram counts of unsupervised acoustic units were used to learn a latent topic model over spoken audio documents. In [4], vector space document modeling techniques were applied to the clustered patterns found by a spoken term discovery algorithm. In [5, 38], similar spoken term discovery algorithms were used to produce acoustic summaries of spoken audio data.

1.2 Problem Statement: Spoken Corpus Summarization

In this thesis, we consider the following problem: suppose we would like to understand the major topical themes within a collection of speech audio documents, without having to listen to each one. If text transcripts or ASR output for each document were available, topic models from Probabilistic Latent Semantic Analysis (PLSA) [14] or Latent Dirichlet Allocation (LDA) [2] could be used to generate a text summary of the corpus as in [10]. In the zero resource setting, these techniques cannot be directly applied. We instead present a method that is similar in spirit, but aims to summarize the topical themes of the corpus by extracting meaningful audio snippets.

For the purpose of generating this kind of summary, we want to associate regions

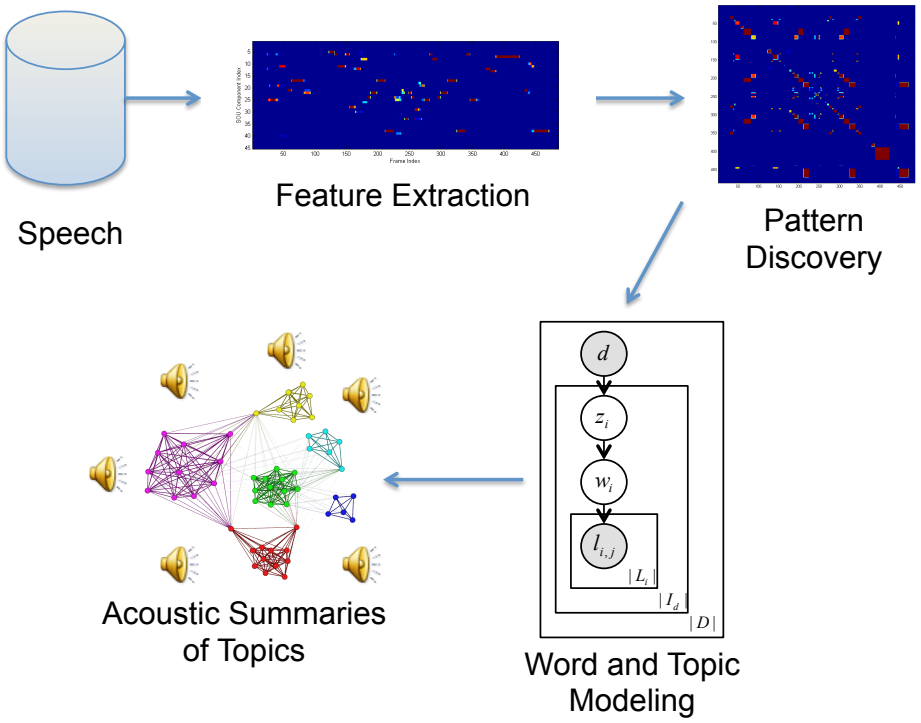

Figure 1-2: Diagram of our unsupervised spoken audio corpus analysis system

of the audio signal with latent topics, analogous to what is done with words in models such as PLSA and LDA. We propose a system (visualized in Figure 1-2) which:

1. Searches the audio corpus for repeated acoustic patterns, often corresponding to repetitions of the same word or phrase.

2. Uses a probabilistic latent variable model to associate these acoustic patterns with latent topics and pseudo-words.

3. Summarizes the topical themes of the corpus by using the model to extract topically meaningful acoustic patterns.

1.3 Thesis Outline

The body of this thesis is organized as follows: In Chapter 2, we present an overview of recent developments in unsupervised acoustic modeling and discuss how the methods apply to our problem at hand. Chapter 3 contains the background information

regarding the unsupervised pattern discovery stage of our system, as well as a description of our implementation. Chapter 4 presents the main contributions of this thesis, a set of methods for characterizing and summarizing the topical themes of a large collection of speech audio. Chapter 5 concludes by discussing possible extensions of the methods we present.

Chapter 2

Unsupervised Acoustic Modeling

2.1 Motivation for Unsupervised Acoustic Models

Mel-Frequency Cepstral Coefficients (MFCCs) are one of the most widely used feature representations used in automatic speech recognition (ASR) systems. Computed by taking the discrete cosine transform of the log magnitudes of a set of filter bank outputs, when applied to a short time segment of speech (usually a 25 millisecond window shifted every 10 milliseconds) they provide a compact estimate of the resonances of the human vocal tract at that instant in time. Additionally, their dimensions are approximately de-correlated by the discrete cosine transform, allowing them to be effectively modeled by diagonal covariance Gaussians commonly used in speech recognizers. Because the shape of the vocal tract varies across speakers, the MFCCs extracted from two different individuals speaking the same words may be very different. To combat this, supervised systems typically employ mixtures of Gaussians along with massive amounts of training data in conjunction with speaker normalization and adaptation techniques such as vocal tract length normalization [1] and maximum likelihood linear regression [26]. Because unsupervised speech processing systems do not have the benefit of training labels, speaker variation becomes a major hurdle. A body of recent work in the zero resource speech processing community has focused on developing speaker independent features and acoustic models specifically for the problem setting in which no labelled data is available [6, 17, 25, 23, 31, 33, 36].

2.2 Phonetic and Gaussian Posteriorgrams

In this thesis, our overarching goal is to characterize the topical content of an audio corpus in an unsupervised fashion. Our approach relies upon first discovering repeated acoustic patterns from the speech of many different people, and so we require representation of the audio signal that is at least somewhat speaker independent. One avenue of research aimed at tackling this speaker independence issue has investigated the use of posteriorgram features and their effectiveness as a more speaker-independent speech representation. In [12], a supervised query-by-example keyword spotting system was developed which utilized posteriorgram features. Because word recognition lattice-based keyword spotting becomes problematic when faced with out of vocabulary search terms, the authors instead aimed to capture the phonetic spelling information of a query term and utilize a dynamic time warping algorithm to match phonetically similar sequences in the corpus of data to be searched. They did this by representing all of the speech data not in terms of acoustic feature vectors such as MFCCs, but rather by computing a posterior probability distribution over phonetic units for every frame of speech audio. Given a set of generative acoustic models θ which model a set of N discrete, phone-like units $u_1, u_2, \ldots, u_N$, an acoustic feature vector $\vec{x}$ can be transformed into a posteriorgram vector $\vec{p}$ in the following way:

$$\vec{p} = [P(u_1|\vec{x}, \theta), P(u_2|\vec{x}, \theta), \ldots, P(u_N|\vec{x}, \theta)]^T \tag{2.1}$$

Using this representation, a dynamic time warping algorithm was applied to align the query term's posteriorgram representation with subsequences of the data to be searched. In [36], the authors presented a similar query-by-example keyword spotting system, but in a completely unsupervised framework. Rather than using phonetic posteriorgram features, the authors demonstrated that Gaussian posteriorgram acoustic features provided increased keyword spotting accuracy across speakers when compared to raw MFCCs. Gaussian posteriorgrams represent each speech frame as a posterior probability distribution across components in a Gaussian Mixture Model (GMM). Given a D dimensional GMM G with N Gaussian components $c_1, c_2, \ldots, c_N$

where the i^{th} component is specified by its weight w_i, mean μ_i, and covariance matrix Σ_i, a feature vector $\vec{x}$ can be represented by its Gaussian posteriorgram vector $\vec{g}$:

$$\vec{g} = [P(c_1|\vec{x}, G), P(c_2|\vec{x}, G), \ldots, P(c_N|\vec{x}, G)]^T \tag{2.2}$$

where $P(c_i|\vec{x}, G)$ represents the posterior probability of the i^{th} component given $\vec{x}$:

$$P(c_i|\vec{x}, G) = \frac{w_i \mathcal{N}(\vec{x}; \mu_i, \Sigma_i)}{\sum_{j=1}^{N} w_j \mathcal{N}(\vec{x}; \mu_j, \Sigma_j)} \tag{2.3}$$

2.3 Lattice Derived Posteriorgrams

Posteriorgrams over phone-like units can also be estimated from the recognition outputs of speech recognition systems. In this case, a recognition lattice is used to represent a set of possible word or phone sequences for a speech utterance. This first requires computing a posterior lattice, which specifies the posterior probability of traversing any given arc in the lattice. Given a lattice where transition arc weights are taken to be likelihood scores, it is straightforward to compute the posterior lattice using the Forward-Backward algorithm. Let I be the set of nodes in the lattice and let $s_{i,j}$ be the weight of the arc starting at node i and ending at node j. The forward variable for node i is computed as

$$\alpha_i = \prod_{k \in I} \alpha_k s_{k,i}, \tag{2.4}$$

and the backward variable is computed as

$$\beta_i = \prod_{k \in I} \beta_k s_{i,k}. \tag{2.5}$$

Assuming the lattice starts at node 0 and ends at node N, the base cases for the recursion relations above are

$$\alpha_0 = 1, \tag{2.6}$$

$$\beta_N = 1. \tag{2.7}$$

Finally, the posterior probability of traversing the arc from node i to node j is equal to

$$\gamma_{i,j} = \frac{\alpha_i s_{i,j} \beta_j}{\beta_0} \tag{2.8}$$

From a posterior lattice, the posterior probability of a word or phone-like unit u occurring at time t can be computed simply by summing the posterior probability of all arcs with label u which cross time t. To compute a posteriorgram vector series for an utterance, we can query the posterior lattice for the probabilities of all units at every 10ms interval.

2.4 Self-Organizing Unit Posteriorgrams

For the experiments in this thesis, we utilize posteriorgram features derived from the recognition lattices provided by the self-organizing unit (SOU) system developed at BBN [31]. The SOU system is very similar to a conventional phone-based HMM speech recognizer, except that acoustic unit labels are learned during training in an unsupervised fashion. The system is initialized by segmenting an audio signal at regions of abrupt spectral change, and then clustering each resulting acoustic segment. The cluster labels become acoustic unit labels, and an HMM for each label is trained. Bigram and trigram language models are also trained over the label sequences, and the resulting acoustic and language models are used to re-recognize all of the training data, producing a new set of segment labels. The process continues, iteratively re-training the acoustic and language models before re-recognizing the training data, and so on.

A subtle albeit very important item to note is the fact that the SOU system contains a silence model. In our experiments, we explicitly ignore the silence unit during the pattern discovery step.

2.5 Other Related Work in Unsupervised Acoustic Model Discovery

Many other approaches towards learning acoustic models in an unsupervised fashion have been explored in the literature. Several techniques investigated by Jansen et al have used the unsupervised pattern discovery architecture introduced by [27] to impose top-down constraints on acoustic sub word unit models in a fashion similar to the way that pronunciation lexicons and word level transcriptions are used in the forced alignment step of training a recognizer. In [17], the authors took clusters of acoustic intervals returned from a pattern discovery search and trained an HMM for each cluster. Then, the states across all HMMs learned were clustered to form acoustic sub word unit models. In [16], the authors took a different approach by first training a bottom-up GMM over a collection of speech feature vectors in a manner similar to the one used by [36]. They next utilized a S-DTW pattern discovery algorithm to align similar words and phrases found in the data. In the last step, the S-DTW alignments were used to cluster the components of the GMM by examining when frames assigned to different Gaussians were aligned to one another by the S-DTW search. The resulting acoustic models formed by clusters of Gaussian components showed significantly increased speaker independence when compared to raw features or the GMM alone.

Recently, Lee and Glass formulated a completely unsupervised Bayesian nonparametric framework for acoustic sub word unit discovery. A key difficulty in many proposed sub word unit discovery algorithms is that of model selection, and most techniques require the number of learned units to be set manually. In [25], the authors attempted to address this in a formal framework by placing prior distributions over their model parameters and then inferring a set of HMMs to represent the sub word units. The authors demonstrated a strong mapping of the learned units to English phones, and showed good results when using the learned units for a keyword spotting task.

A completely different approach was taken by Varadarajan et al in [33] based

upon a successive state splitting algorithm for HMMs. Starting with a collection of speech from one speaker and a single state GMM-HMM model, at each iteration the algorithm considers splitting each HMM state into two new states by computing the gain in data likelihood by splitting. If this gain is above a threshold, the state is split. The algorithm continues splitting the HMM states in this fashion. Although the algorithm was only evaluated on the speech of a single speaker in a single session, the authors demonstrated a strong mapping between the inferred HMM state labels on the test set and the underlying phonetic labels.

Chapter 3

Unsupervised Pattern Discovery

3.1 Dynamic Time Warping

Dynamic Time Warping (DTW) is a well-known algorithm for finding an optimal alignment between two time series, and throughout the 1970s and 1980s was a very popular method of performing speech recognition. One reason for this is the fact that DTW-based recognizers are very easy to build; at the very minimum, all that is necessary is a single spoken example of each word that could be recognized. On simple tasks like digit recognition, DTW can be incredibly effective, especially if the training and test data are spoken by the same speaker. Moreover, DTW makes very few modeling assumptions, and the comparison between a test utterance and the training utterances can be done directly on the feature level. Although there are continued efforts towards implementing large vocabulary continuous speech recognition (LVCSR) systems using DTW [30], statistical frameworks such as Hidden Markov Models (HMMs) have emerged as the dominant approach, thanks in large part to the widely used toolkit HTK [32]. Recently, however, there has been renewed interest in DTW, especially for unsupervised applications in which there exists little or no training data that can be used to fit statistical models [4, 5, 12, 17, 18, 19, 27, 35, 36, 37, 38].

Let $X = x_1, x_2, \ldots, x_M$ and $Y = y_1, y_2, \ldots, y_N$ be two vector-valued time series. An *alignment path* $A = \phi_1, \phi_2, \ldots, \phi_T$ defines a set of frame pairs that matches frames in X with frames in Y. Each $\phi_i = (a_i, b_i)$ represents a pairing between frame x_{a_i} and

frame y_{b_i}. Typically, the following constraints are imposed on A:

1. The alignment path begins at the start of each sequence; that is, $\phi_1 = (1, 1)$

2. The alignment path terminates at the end of each sequence; that is, $\phi_T = (M, N)$

3. The alignment path is monotone and may not move backwards in time; that is, $i < j \implies a_i \leq a_j$ and $b_i \leq b_j$.

4. The alignment path is continuous. That is, the warp path may not advance more than one frame at a time in either X or Y. The path is allowed, however, to advance by one frame in both X and Y simultaneously.

Let $Dist(x, y)$ denote the *distortion* between frames x and y. In practice $Dist(x, y)$ may be any measure of the dissimilarity of x and y, such as Euclidean distance, cosine distance, etc. The total distortion D accumulated along an alignment path A can then be written as:

$$D(A) = \sum_{i=1}^{T} Dist(x_{a_i}, y_{b_i}). \tag{3.1}$$

Let $\hat{A}$ be the optimal path aligning X and Y which minimizes D. Also let C be a matrix with element $C_{i,j}$ equal to the total distortion of the optimal path from (x_1, y_1) to (x_i, y_j). Clearly, $D(\hat{A}) = C_{M,N}$. We can efficiently solve for $\hat{A}$ in $O(MN)$ time using a dynamic programming algorithm that takes advantage of the following insight:

$$C_{i,j} = Dist(x_i, y_j) + min(C_{i-1,j-1}, C_{i-1,j}, C_{i,j-1}) \tag{3.2}$$

Using the above equation, we can construct C column by column, with the additional constraints $C_{0,0} = 0, C_{0,j} = \infty, C_{i,0} = \infty$. By simultaneously constructing a back-pointer table B where $B_{i,j}$ holds a reference to the second-to-last frame pair ϕ along the optimal path from (x_1, y_1) to (x_i, y_j), we can easily recover $\hat{A}$ via lookup.

3.2 Segmental Dynamic Time Warping and Acoustic Pattern Discovery

Dynamic time warping in its basic form provides an efficient mechanism for comparison between two time series, but it can also be modified to reveal similar *subsequences* within the time series. A technique for doing this known as segmental dynamic time warping (S-DTW) was introduced in [27]. The authors applied S-DTW to search a single, long audio waveform for repeating acoustic subsequences, which often corresponded to repeated instances of the same word or short phrase.

S-DTW at its core involves the application of several additional constraints to the vanilla DTW algorithm. The first of these constraints allows the warping path to start and end in the middle of the sequences to be aligned. The second is the addition of a band-width restriction, which prevents the warp path from straying too far away from the diagonal. The band-width restriction for a width R can be expressed as:

$$|(a_k - a_1) - (b_k - b_1)| \leq R,\ 1 \leq k \leq T. \tag{3.3}$$

By beginning the restricted search at various starting frames within each of the two sequences, all possible subsequence alignments can be considered. The two cases defining these possible start coordinates are given by:

$$\phi_1 = ((2R+1)j + 1, 1),\ 0 \leq j \leq \left\lfloor \frac{M-1}{2R+1} \right\rfloor \tag{3.4}$$

$$\phi_1 = (1, (2R+1)j + 1),\ 1 \leq j \leq \left\lfloor \frac{N-1}{2R+1} \right\rfloor \tag{3.5}$$

In [27], the key application of S-DTW explored by the authors was unsupervised acoustic pattern discovery. Given a collection of speech audio, the S-DTW algorithm was employed to find locally optimal alignments between audio segments without the benefit of transcriptions or any acoustic models. These alignments often corresponded to repetitions of the same word or short phrase. An example taken from [27] is shown in Figure 3-1 and shows the alignment paths for all starting coordinates between an

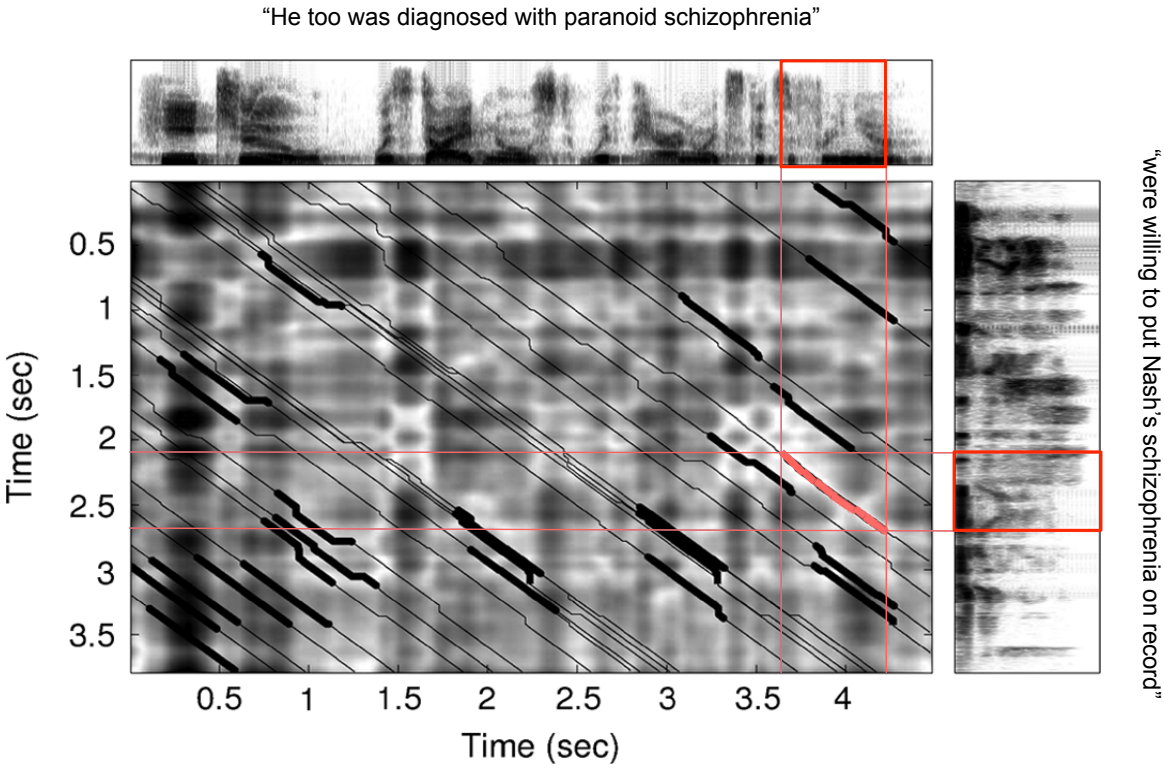

Figure 3-1: A S-DTW alignment between two utterances containing the word "schizophrenia," where distances are based upon spectral features, borrowed from [27]. Each thin black path corresponds to an alignment associated with different start and end coordinates. The thick black part of each path is the LCMA subsequence of the path. After filtering out high distortion LCMA subsequence alignments, only the alignment highlighted in red remains, which aligns the two instances of the word "schizophrenia" while ignoring the rest of both utterances.

utterance of *"He too was diagnosed with paranoid schizophrenia"* and an utterance of *"were willing to put Nash's schizophrenia on record"*. One of the warping paths contains a low-distortion region aligning the two instances of the word "schizophrenia", highlighted in red. To find this subsequence of the local alignment path, the authors employed a length-constrained minimum average (LCMA) subsequence search for each alignment path. This search only retains a contiguous, low-distortion chunk of each alignment path. The result is a collection of alignment path fragments where each fragment aligns an audio interval $[t_1^{(a)}, t_2^{(a)}]$ to another audio interval $[t_1^{(b)}, t_2^{(b)}]$ with a distortion score d, computed by summing the pairwise frame distortions along the alignment fragment. After filtering out alignment fragments with a high distortion, the remaining alignment fragments tend to correspond to repeated instances of the same or similar words.

3.3 Our Implementation

While the authors in [27] and [35] demonstrated the effectiveness of the S-DTW pattern discovery algorithm on audio recordings whose length was on the order of several hours, the biggest hurdle to applying S-DTW to increasingly larger speech corpora is the $O(N^2)$ complexity of the search. To help alleviate this computational cost, Jansen et al have proposed two approximations to the standard S-DTW algorithm which provide dramatic speed gains [18, 19]. The experiments detailed in this thesis use a pattern discovery implementation similar to the one described in [18]. In that work, the authors introduce a two-pass approximation to the full S-DTW search. From a high level, this strategy treats the distance matrix as a 2-D image and applies image filtering techniques to perform a coarse search for approximately diagonal lines. The line segments themselves can be treated as alignment path segments, much the same as when the full S-DTW algorithm is used. The second pass of the algorithm refines the line segments by applying a restricted S-DTW search only to the regions occupied by the discovered diagonal lines.

We will describe the pattern discovery search procedure we use in detail here. Let

$P = \vec{p}_1, \vec{p}_2, \ldots, \vec{p}_M$ and $Q = \vec{q}_1, \vec{q}_2, \ldots, \vec{q}_N$ be the posteriorgram representations of two speech utterances, where each $\vec{p}_i$ and $\vec{q}_i$ are G dimensional vectors representing categorical probability distributions over a discrete set of speech units. That is, the j^{th} element of $\vec{p}_i$ represents the posterior probability that speech frame i of utterance P was generated by speech unit j. The first step is to construct a matrix S where $s_{i,j}$ represents some measure of similarity between $\vec{p}_i$ and $\vec{q}_j$. Any vector space metric or measure may be used, such as Euclidean distance, cosine similarity, dot product similarity, or KL divergence. Because the posteriorgram vectors represent probability distributions, the dot product similarity has an interesting interpretation; $\vec{p}_i \cdot \vec{q}_j$ represents the probability that the i_{th} frame of utterance P and the j_{th} frame of utterance Q were independently generated by the same, single speech unit. Because it is also simple and fast to compute, especially for sparse posteriorgram vectors, we use the dot product similarity in our experiments. Because each $\vec{p}_i$ and $\vec{q}_j$ represent probability distributions, each element is guaranteed to be nonnegative, and their L_1 norms will always be 1. Therefore, $0 \leq s_{i,j} \leq 1$.

One significant advantage that posteriorgram representations offer compared to spectral representations is sparsity. Figure 3-2 shows the spectrogram representation of three instances of the utterance "vocal tract" being spoken, while Figure 3-3 displays the posteriorgram representation of the same three utterances. The fact that most dimensions of the posteriorgram vectors are close to zero means that we can employ a sparse vector-vector product algorithm for the computation of the similarity matrix after flooring elements of the posteriorgram vectors very close to 0. Figure 3-4 displays four similarity matrices computed using a sparse vector-vector product algorithm after applying a minimum similarity threshold to the posteriorgrams. By zeroing out elements of the posteriorgram vectors with a probability less than 0.01, the number of multiply-adds required to compute the similarity matrix is cut by a factor of 17.6 with almost no difference in the resulting matrix. By using a threshold of 0.3, the number of multiply-adds is cut by a factor of more than 44, albeit with some visible degradation in the quality of the similarity matrix.

The next step of the search is to quantize the similarity matrix to binary in order

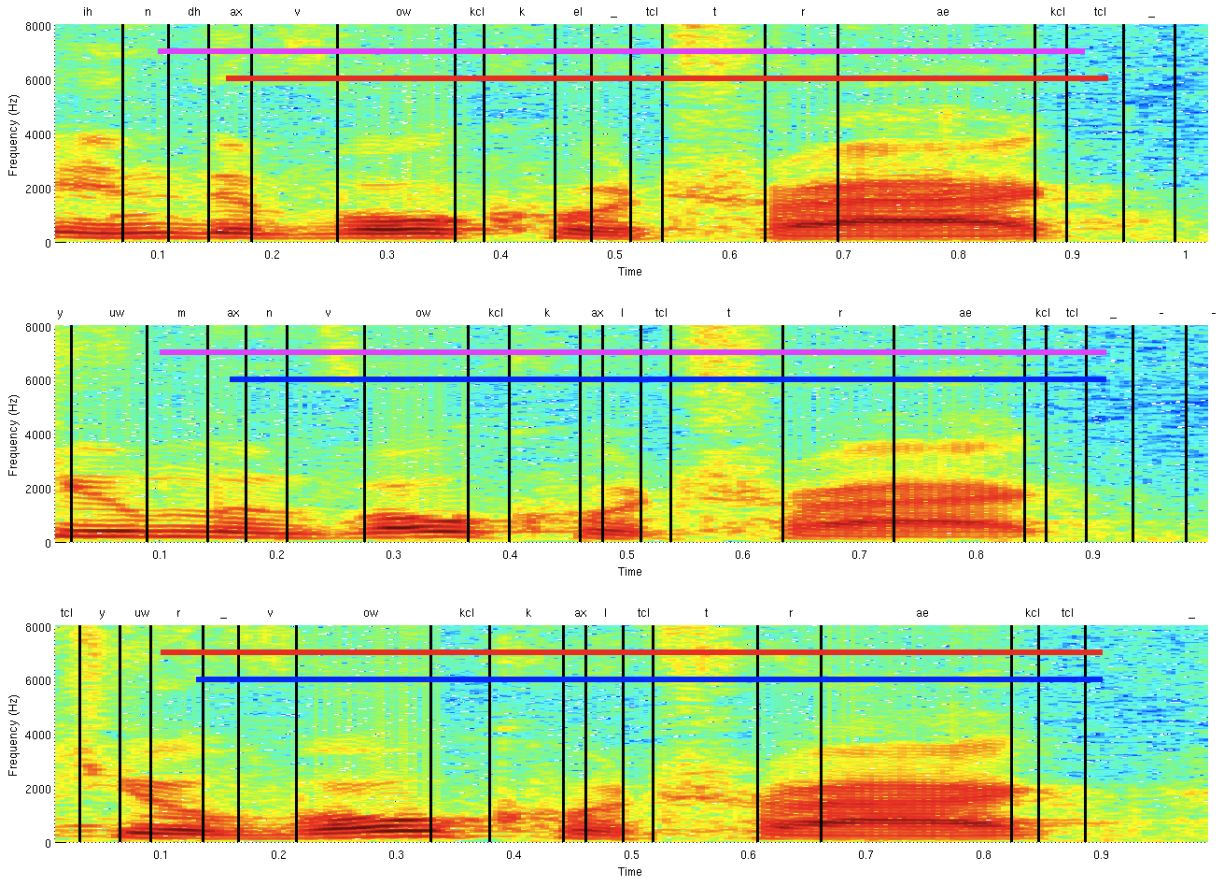

Figure 3-2: Three-way match between three spoken instances of the phrase "vocal tract" shown in their spectrogram representation. Each pair of horizontal sharing the same color (purple, red, and blue) highlights where the pairwise match intervals overlap the speech in time.

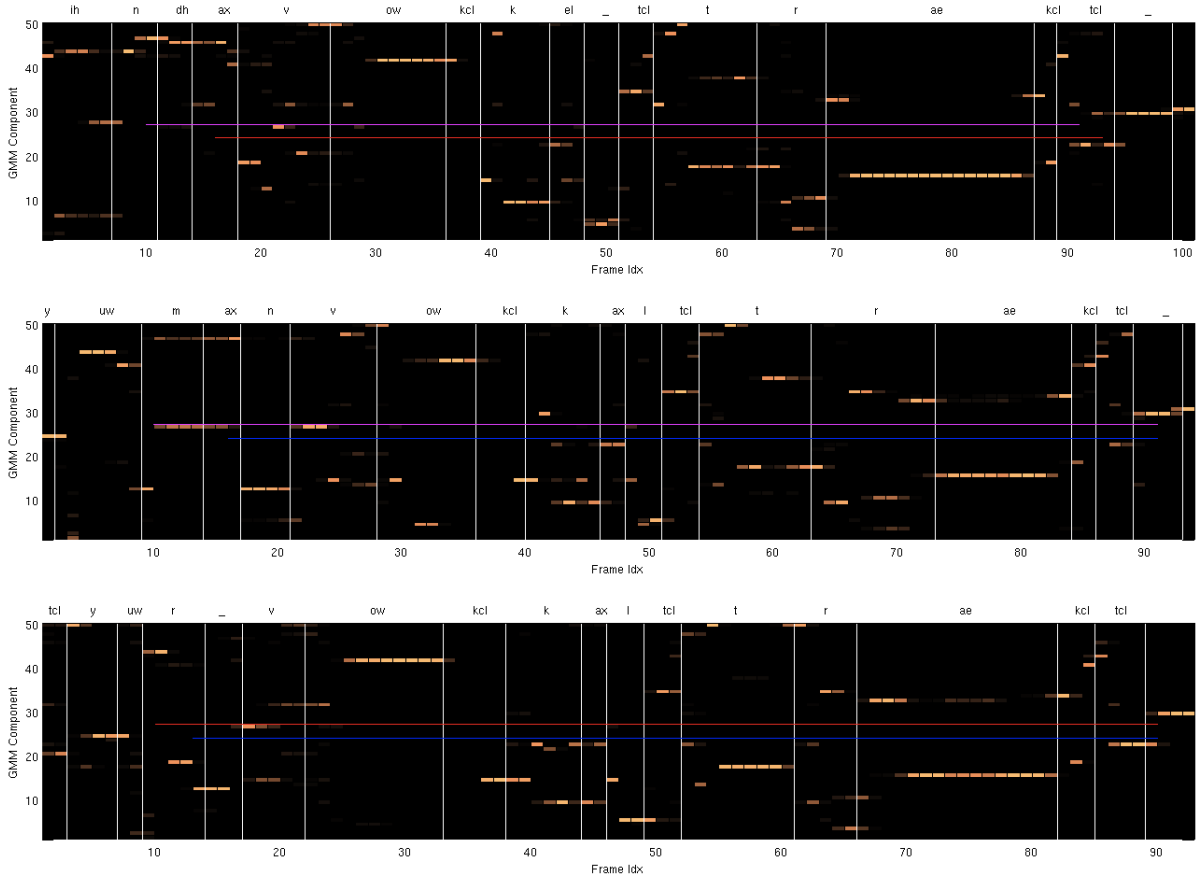

Figure 3-3: Three-way match between the same three spoken instances of the phrase "vocal tract" displayed in Figure 3-2, this time shown in their GMM posteriorgram representation. Lighter color represents higher posteriorgram vector probability with black denoting 0.

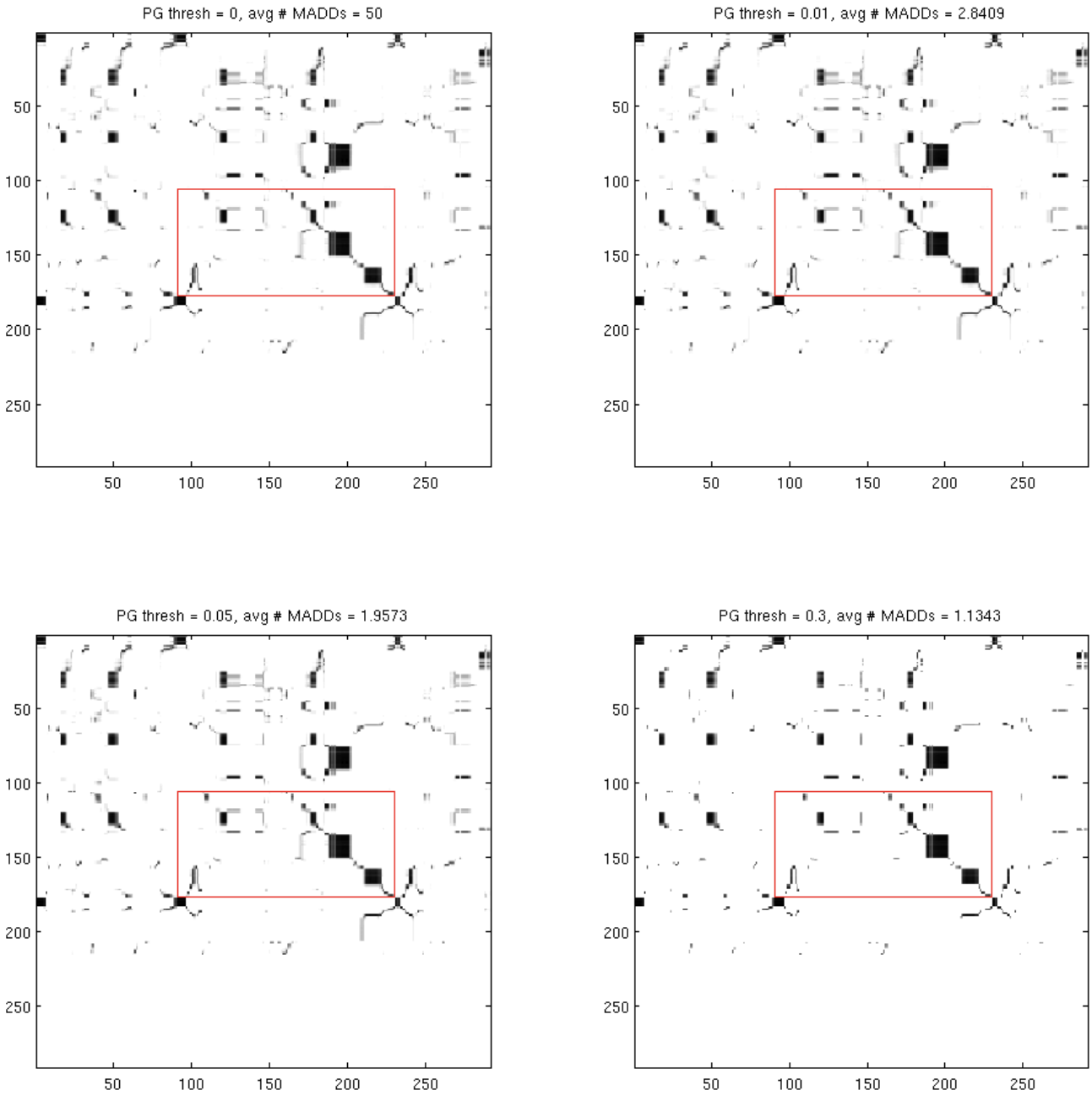

Figure 3-4: Effect of the posteriorgram quantization threshold for values of 0, 0.01, 0.05, and 0.3. Posteriorgram elements with value below the threshold are set to 0, and sparse vector-vector products are used to compute elements of the dotplot. The average number of multiply-adds (MADDs) needed to compute each element of the dotplot for each setting of the threshold are shown. Setting a quantization threshold of 0.3 reduces the number of multiply-adds necessary by a factor of more than 44 while resulting in very little degradation in dotplot fidelity.

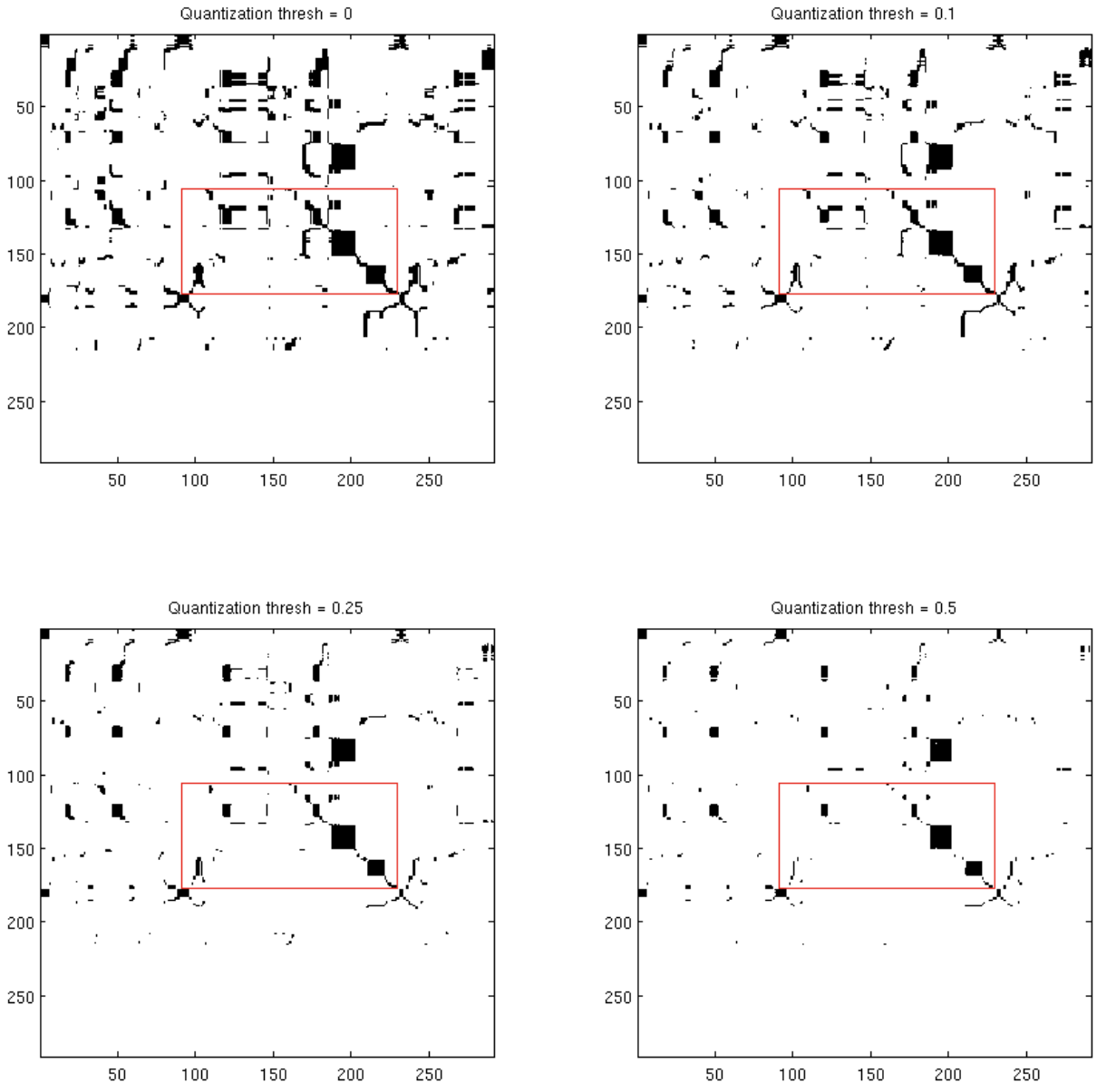

Figure 3-5: Effect of the similarity matrix quantization threshold for values of 0, 0.1, 0.25, and 0.5. Elements of the dotplot matrix with value below the threshold are set to 0, and elements above the threshold are set to 1. Raising the threshold results in more sparseness.

to accommodate the median smoothing filter. We do this by setting each $s_{i,j}$ to 0 if $s_{i,j} < \tau$ for some fixed τ, and 1 otherwise. The effect of various settings of τ on the similarity matrix is shown in Figure 3-5. Even when using relatively small values of τ, the resulting matrix is still quite sparse, so we often use a small threshold in practice. After quantizing the matrix, we apply a nonlinear image filter targeted towards extract diagonal structure. This takes the form of a diagonal median smoothing filter parameterized by μ, the median threshold, and L, the filter lookahead. Given a matrix A of size M by N, applying the filter to A results in a new matrix B of size M by N computed in the following way:

$$b_{i,j} = \begin{cases} 1: & \frac{1}{2L+1} \sum_{k=-L}^{L} a_{i+k,j+k} \geq \mu \\ 0: & \frac{1}{2L+1} \sum_{k=-L}^{L} a_{i+k,j+k} < \mu \end{cases} \tag{3.6}$$

where $a_{i+k,j+k}$ is assumed to be zero when $i+k$ or $j+k$ exceed the bounds of the matrix (i.e. are less than 1, or when either $i+k > M$ or $j+k > N$). Figure 3-6 illustrates the quantization and filtering steps applied to a raw similarity matrix.

Although the diagonal median smoothing filter effectively filters out non-diagonal structure in the similarity matrix, the warp paths we are searching for often do not conform exactly to the 45 degree diagonal assumption made by the filter. This can result in broken or fragmented lines, so we smear the median filtered matrix by convolving it with a constant image patch. Once the image has been smeared, we apply a 1 dimensional Hough transform with θ fixed at 45 degrees. This amounts to simply calculating the sum of every diagonal of the smeared matrix. A simple peak picking algorithm applied to the Hough transform locates the diagonals along which the smeared matrix is searched for contiguous nonzero regions. Each contiguous region found in this fashion results in a diagonal line segment representing the approximate location and orientation of a low-distortion path in the similarity matrix. Figure 3-7 illustrates the smeared matrix along with its Hough transform, in addition to a found diagonal line segment overlaid on the original similarity matrix.

The final step in the search is to warp the diagonal line segments so that they

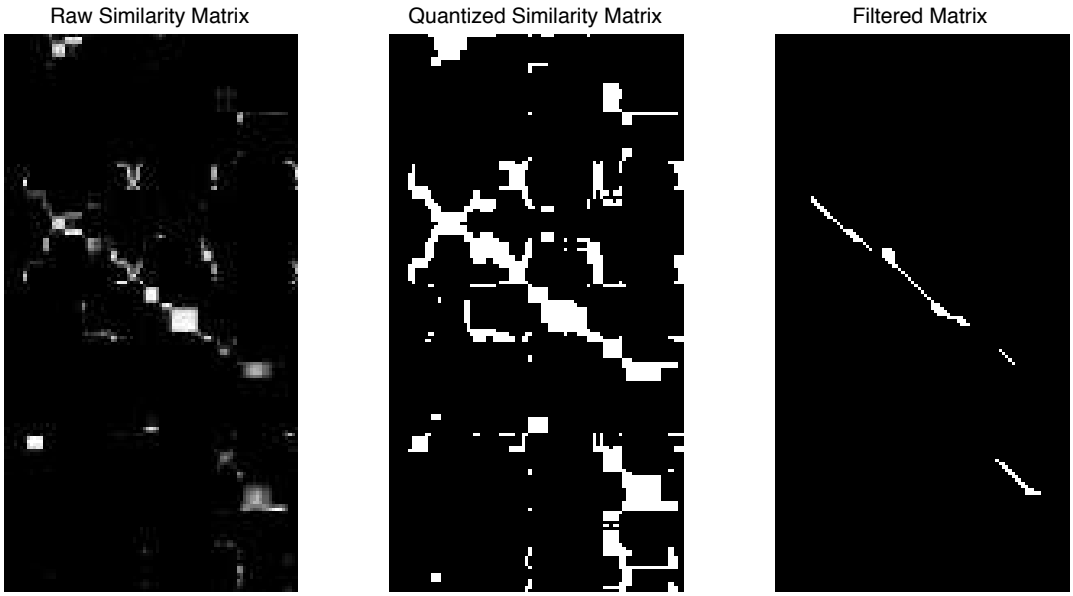

Figure 3-6: Steps 1 through 3 of the pattern search procedure. The raw similarity matrix is first computed by computing the inner product of the posteriorgram vectors belonging to each pair of frames. The matrix is then subject to a binary quantization in the second step. The third step applies a nonlinear diagonal median smoothing filter to reveal diagonal line structure.

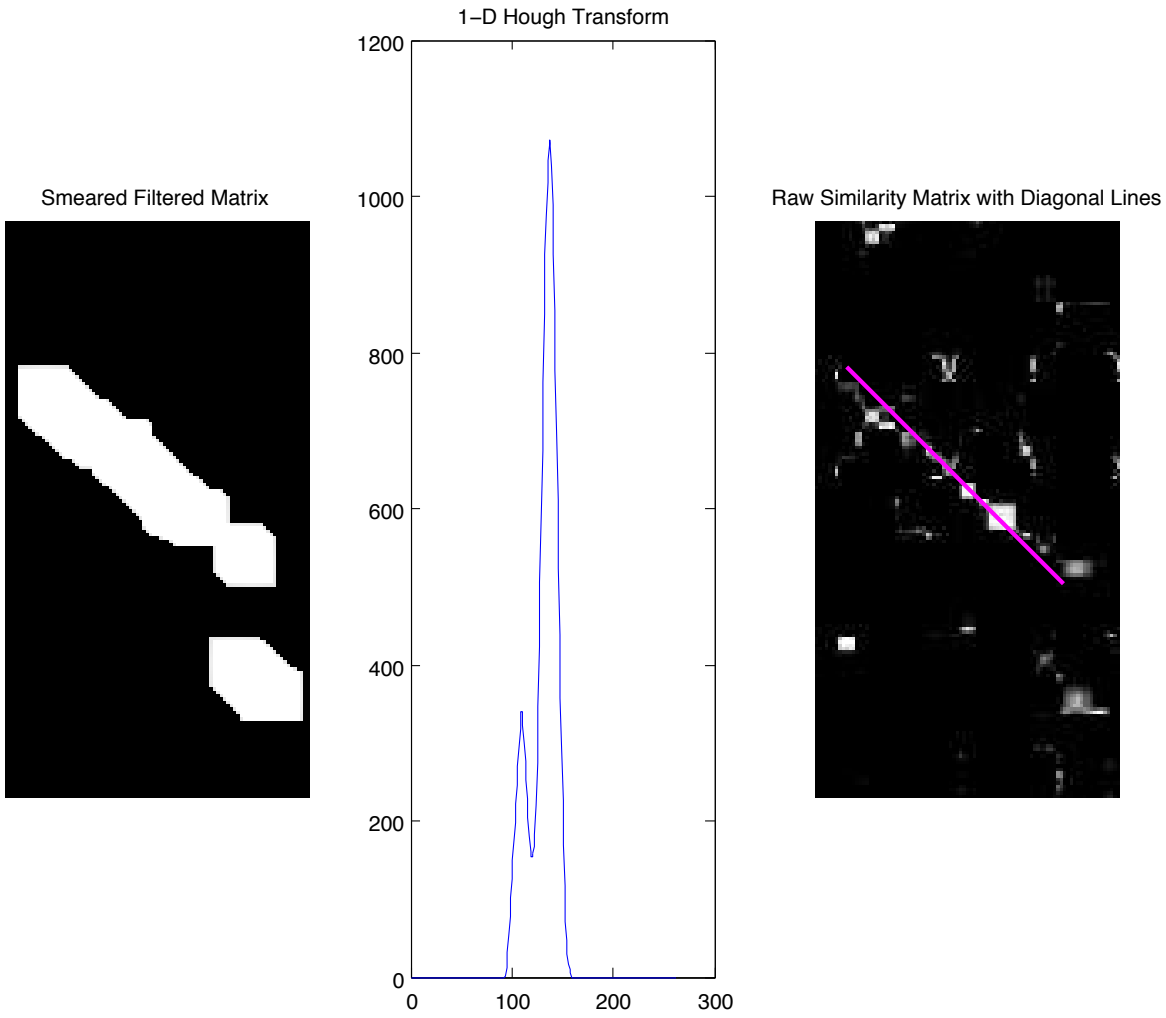

Figure 3-7: Steps 4 through 6 of the pattern search procedure. The nonzero values of the filtered matrix from step 3 are smeared via convolution with an image patch matrix. A 1-D Hough transform is then applied by simply summing every diagonal of the smeared matrix. A simple peak picking algorithm defines the diagonal rays along which the smeared matrix is searched for line segments.

more closely follow the high similarity regions of the raw similarity matrix. We utilize a slightly modified version of S-DTW for this step by introducing the additional constraint that the warp path must pass through the midpoint of its corresponding line segment. Given a line segment with midpoint (m_x, m_y), we perform two separate S-DTW warps. The first warp's start coordinate is taken to be (s_x, s_y) where $s_x = \max(m_x - k, 1)$, $s_y = \max(m_y - k, 1)$, and $k = \min(m_x, m_y)$. The end coordinate of the warp is fixed to be (m_x, m_y), and the minimum cost path is computed. In this step we define the distortion between frames $\vec{p_i}$ and $\vec{q_j}$ as $1 - s_{i,j}$. Rather than keeping the entire alignment path, we are only interested in the low-distortion alignment local to (m_x, m_y); therefore, to determine the fragment of the S-DTW path to retain we walk backward along the path from (m_x, m_y) and accumulate distortion along the way until it exceeds a distortion budget B. The second S-DTW warp is constrained to begin at (m_x, m_y) and end at (e_x, e_y) where $e_x = m_x + l$, $e_y = m_y + l$, and $l = min(M - m_x, N - m_y)$. Again, we walk outwards along the alignment path starting from (m_x, m_y) until the accumulated distortion exceeds B. Merging the two warp path fragments at (m_x, m_y) results in the final alignment. The result of applying the S-DTW refinement to the line segment displayed in Figure 3-7 is shown in Figure 3-8.

3.4 Chapter Summary

In this chapter, we have described the basics of dynamic time warping, a dynamic programming algorithm commonly used to find optimal alignments between time series. We have also described the segmental dynamic time warping algorithm introduced by [27], which can be used to discover acoustic repetitions in an audio stream which commonly correspond to repetitions of the same underlying word or phrase. We have laid out in detail our implementation of the pattern discovery algorithm used in this thesis, which is based upon the 2-pass approximation algorithm introduced in [18], and also takes advantage of the posteriorgram feature representation introduced in [12] and [36].

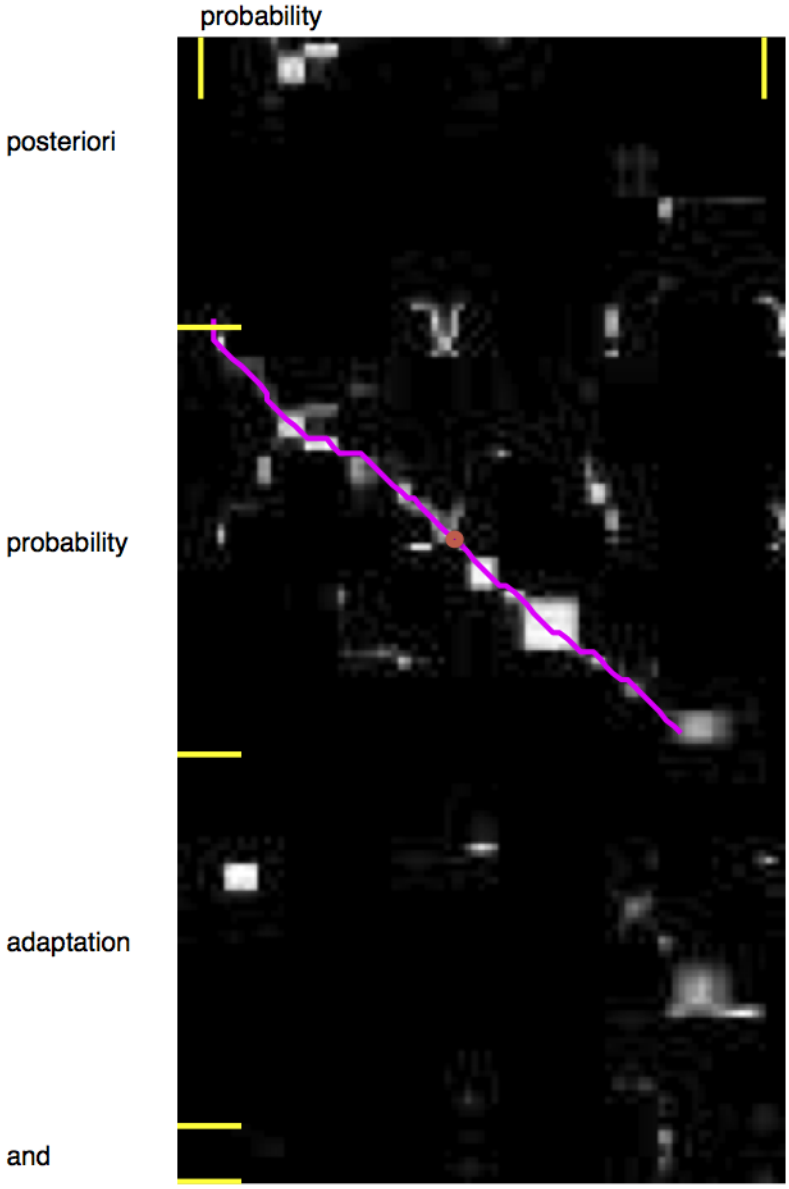

Figure 3-8: The warp path refined from the diagonal line segment shown in Figure 3-7, with the time aligned transcriptions of both utterances shown along the vertical and horizontal axes. Notice how the warp path clings to the low distortion regions of the dotplot as compared to its parent line segment, providing a more accurate estimate of the match distortion.

Chapter 4

Modeling of Latent Topics and Words

4.1 Background on Document Modeling

The notion of a *document* is central to many applications of natural language processing such as search, information retrieval, and automatic summarization. Text documents may take many forms, including news articles, webpages, email messages, and transcriptions of spoken conversations, to name a few. A document may also refer to recorded speech, such as a radio news broadcast, an academic lecture, or a telephone conversation. Many efforts in text mining and information retrieval utilize statistical learning algorithms which model data at the document level, so it is natural to consider how documents may be mapped to feature vectors suitable for learning and inference. In this section, we provide a brief overview of document modeling techniques largely drawn from [9].

The ubiquitous method of extracting feature vectors from text documents is by using a so-called *bag-of-words*. The bag-of-words approach implicitly ignores context information and assumes that the order in which words appear is irrelevant. Formally, let vocabulary V be a set of N_V unique and ordered words,

$$V = \{w_1, \ldots, w_{N_V}\}, \tag{4.1}$$

and let D be a collection of N_D documents,

$$D = \{d_1, \ldots, d_{N_D}\} \tag{4.2}$$

For any $w \in V$ and any $d \in D$, the *term-frequency* of w in d is denoted by $tf(w, d)$ and is equal to the number of times word w appears in document d. Let the N_V dimensional vector $\vec{x}_d$ be the feature vector representing document d. We construct a bag-of-words or term-frequency vector, $\vec{x}_d$, for document d as follows:

$$\vec{x}_d = \left[x_{w_1}, \ldots, x_{w_{N_V}}\right]^T \tag{4.3}$$

$$x_w = tf\,(w, d) \tag{4.4}$$

This method of mapping documents to vectors is sometimes known as *direct modeling*, as individual word occurrences are directly represented in the feature space. Direct modeling of documents is simple, yet effective, and has proven its worth in tasks such as email spam filtering [29]. In addition to finding use as feature vectors in many popular machine learning algorithms, bag-of-words vectors allow documents to be compared to one another using standard vector space measures. A very popular similarity measure for doing so is the cosine similarity. The cosine similarity between vectors $\vec{x}$ and $\vec{y}$ is defined as

$$Sim_{CS}(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}||_2 ||\vec{y}||_2} \tag{4.5}$$

Direct modeling of documents is not without its problems. Function words such as articles, prepositions, and conjunctions tend to be far more frequent than content words like nouns, verbs, and adjectives. Bag-of-words vectors are hence often dominated by counts of words bearing little lexical meaning. In practice, one way of alleviating this problem is by using handcrafted stop-lists which explicitly specify a list of words to be ignored. Other methods attempt to give more weight to some words compared to others. An example of this is *inverse document frequency* (idf)

weighting [22]. This can be expressed as:

$$\lambda_w = idf\,(w) = \log\left(\frac{N_D}{N_{D\cap w}}\right) \tag{4.6}$$

where $N_{D\cap w}$ is the number of documents in D containing the word w. Incorporating these weights modifies Eq. 4.7 to become:

$$x_w = \lambda_w tf\,(w, d) \tag{4.7}$$

Intuitively, when word w frequently appears in many documents, it is more likely to be a function word. In this case, $N_{D\cap w}$ is large relative to N_D, causing λ_w to be small. When $N_{D\cap w}$ is small relative to N_D, λ_w grows larger, reflecting the fact that w only appears in a small subset of the documents in the collection and is likely to be content-carrying or topic-specific.

A more challenging issue to overcome is the fact that the size of most vocabularies are very large, often on the order of tens of thousands of words. It is desirable then to have a reduced dimensionality representation of bag-of-words vectors. Latent topic modeling of documents is one way of doing this, and has been a very active area of research in the natural language processing community in recent years. One popular latent topic model is Probabilistic Latent Semantic Analysis (PLSA) [14]. PLSA is a generative probabilistic model which assumes that the term-frequency vector associated with each document is randomly generated from a mixture of unigram language models. Each language model is associated with a latent topic variable z from a set, $Z = \{z_1, \ldots, z_{N_Z}\}$, of N_Z latent topics. The probability of generating word w from topic z is given by $P(w|z)$. Furthermore, it is assumed that each document possesses a probability distribution over the latent topics, where $P(z|d)$ acts as the mixture weight for topic z in document d. The probability of observing word w in document d can be expressed in terms of these distributions as follows:

$$P(w|d) = \sum_{z\in Z} P(w|z)P(z|d) \tag{4.8}$$

Let $C = \{c_1, \ldots, c_{N_V}\}$ represent document d, where $c_w = tf(w, d)$; that is, C contains the counts of each word appearing in d. Given d, the probability of observing all of the words appearing in the document is:

$$P(C|d) = \prod_{w \in V} \left(\sum_{z \in Z} P(w|z)P(z|d) \right)^{c_w} \tag{4.9}$$

In order to fit a PLSA model to a document collection, maximum likelihood training is typically done using an Expectation-Maximization (E-M) algorithm. The expectation (E) step first computes the posterior probability distribution for each latent topic variable conditioned on the data and previous estimate of the model parameters:

$$P(z|w, d) = \frac{P(w|z)P(z|d)}{\sum_{z' \in Z} P(w|z')P(z'|d)} \tag{4.10}$$

In the maximization (M) step, the model parameters are re-estimated based upon the data and the posterior distribution of each latent variable:

$$P(w|z) = \frac{\sum_{d \in D} tf(w, d)P(z|d, w)}{\sum_{w' \in V} \sum_{d \in D} tf(w', d)P(z|d, w')} \tag{4.11}$$

$$P(z|d) = \frac{\sum_{w \in W} tf(w, d)P(z|d, w)}{\sum_{z' \in Z} \sum_{w \in W} tf(w, d)P(z'|d, w)} \tag{4.12}$$

4.2 Modeling Spoken Audio Documents

Not all documents are text. News broadcasts, telephone calls, lectures, movies, television programs, and YouTube videos are all sources of large amounts of recorded speech audio. The advent of the internet combined with the inexpensive and massive data storage available today make good indexing and retrieval algorithms for audio documents all the more necessary.

The simplest and most straightforward way of handling collections of spoken audio documents is by making use of tools already available - namely, automatic speech

recognition and vector space document modeling techniques. Efforts into this arena have produced very good results on Fisher English data by first estimating word counts using the output lattices from a speech recognizer and then applying a PLSA topic model to the resulting bag-of-words vectors [10]. In the absence of a full speech recognizer, estimated counts of triphone sequences may be used in place of words [11].

When no recognition capability is available, unsupervised methods present an opportunity to map spoken audio documents to discrete symbol sets that may be used for indexing. Segmental dynamic time warping-based acoustic pattern discovery, described in Chapter 3, is a technique growing in popularity for analyzing streams of spoken audio in a completely unsupervised fashion. Its ability to align regions of the audio stream bearing high acoustic similarity to one another can be coaxed into providing a discrete representation of audio documents suitable for the kinds of document modeling techniques presented in Section 4.1.

The research most similar to that contained in this thesis was conduced by Dredze et al in [4]. In that work, the spoken term discovery system outlined in [18] was used to find a set of acoustic matches within an audio document collection. These match intervals were then clustered according to their DTW similarity, and within-document cluster counts were used to form bags-of-words vectors. Experimental results showed competitive performance in both document clustering and document classification tasks. It is important to note, however, that the pattern discovery algorithm used by the authors employed a supervised multi-layer perceptron trained on 250 hours of transcribed data to compute frame-level phonetic posteriorgram vectors. The work presented in this thesis differs in two significant ways. Firstly, our system is completely unsupervised from the ground up. Second, we utilize a probabilistic latent variable model to perform a soft clustering of acoustic match intervals into pseudo-word categories, and the resulting pseudo-word categories into latent topic categories.

4.3 Linked Audio Document Representation

In the absence of any transcriptions or ASR technology, one way to construct a discrete representation of a collection of spoken audio documents is with the aid of an acoustic pattern discovery system. In this section, we describe a method of representing the audio documents within such a collection in terms of a graph structure which reflects pairwise matches between intervals of audio. Assume that a spoken term discovery system (such as the one described in Chapter 3) has been applied to a spoken document collection and returned a set of matches, M. Each element of M is a triple consisting of a distortion score and two matched regions of audio, $(t_1^{(a)}, t_2^{(a)})$, and $(t_1^{(b)}, t_2^{(b)})$. Because we use the dot product between posteriorgram vectors to compute frame distortions, each frame pair's distortion score can range between 0 and 1. The distortion score of a match is computed by summing the distortion score of each frame pair across the path aligning $(t_1^{(a)}, t_2^{(a)})$ and $(t_1^{(b)}, t_2^{(b)})$, then dividing by the length of the alignment path. Since the match distortion score is length normalized and always ranges between 0 and 1, we can conveniently use it to filter out high-distortion matches which may be untrustworthy; in our experiments, all matches with a distortion score above 0.5 are discarded. Additionally, matches that are extremely short in length tend to reflect alignments between similar sub-word units and may not be indicative of a match between two similar words or phrases. To remove these spurious matches, we filter out any match with an average length of less than 0.5 seconds.

Next, we require a means of collapsing overlapping regions into a single interval so as to resolve when multiple matches include the same region of audio. We use a method of doing this similar to the one used in [4] that collapses overlapping regions to the same interval whenever their fractional overlap exceeds a threshold set to 0.75. The result is a collection of intervals, where each interval consists of one or more match regions which overlap in time. For each interval i, we choose the start time, $t_1^{(i)}$, and end time, $t_2^{(i)}$ by averaging the start and end times of all regions collapsed to i. Each interval i is assigned a set of *links* derived from all match regions that overlap

D	The set of all spoken documents
$d \in D$	A single spoken document
I	The set of all audio intervals returned by the spoken term discovery algorithm
$i \in I$	A single interval of audio
I_d	The subset I containing only the intervals which appear in document d
L_i	The set of links to the audio intervals which match interval i according to the spoken term discovery algorithm
$l \in L_i$	An individual link from interval i. Can be thought of as a pointer to some other interval in I which matched interval i.
L_d	The set of links to the audio intervals which match all of the intervals contained in document d
$v^{(d)}$	Bag-of-links vector representing document d
Z	The set of all latent topic variables
$z \in Z$	An individual latent topic variable
W	The set of all pseudo-word variables
$w \in W$	An individual pseudo-word variable

Table 4.1: A compilation and description of all variables used in our latent models.

it. We assign i a link set, $L_i = \{l_{i,1}, l_{i,2}, \ldots, l_{i,|L_i|}\}$, where each $l \in L_i$ takes on as its value the index of some other interval j such that there exists a match in M linking a region of audio overlapping i with a region of audio overlapping j. For each interval, this yields a triple $i = (t_1^{(i)}, t_2^{(i)}, L_i)$. After this process, we are left with a set of $|I|$ intervals, $I = \{i_1, i_2, \ldots, i_{|I|}\}$, with the subset of intervals appearing in document d denoted by I_d. A visual representation of a linked spoken audio document collection is shown in Figure 4-1. In the upcoming descriptions of our models, many variables are employed, and so for notational convenience we provide a summary in Table 4.1.

4.4 PLSA on Bags-of-Links

4.4.1 The PLSA-BoL Model

At a high level, our goal is to characterize the document collection D in terms of a set of latent topics Z, in the same spirit as algorithms such as PLSA applied to

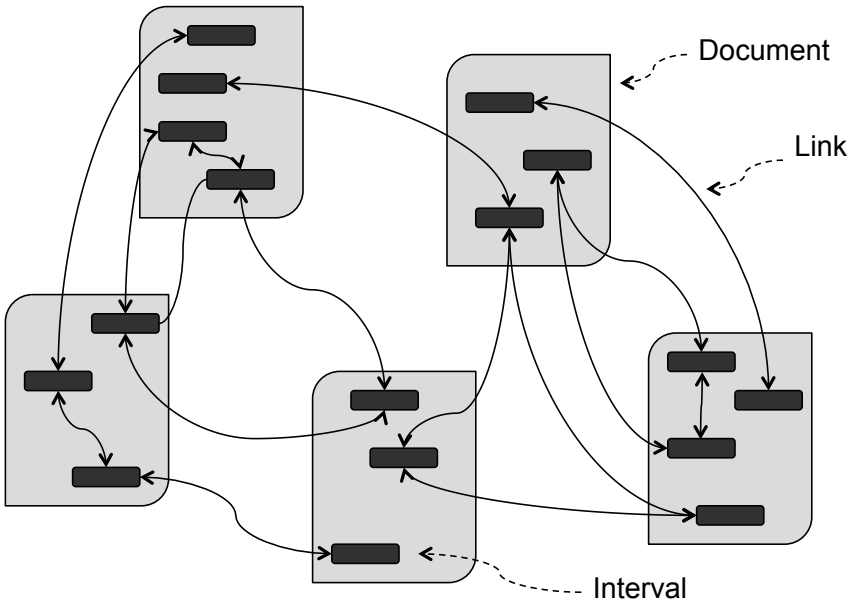

Figure 4-1: An example of a linked audio document corpus. Gray boxes represent documents, and the black rectangles inside them represent the audio intervals they contain. The lines between audio intervals reflect matches discovered by the pattern discovery step.

text documents. We draw inspiration from these text-based document models, but what differentiates our data from text is the fact that we do not know the word-level transcription underlying each interval of audio discovered by the spoken term discovery algorithm. In this section, we present two latent variable models which aim to capture the topical themes of a spoken audio document collection in the absence of any lexical knowledge.

Assume that a pattern discovery algorithm has been applied to a corpus of audio documents, D, providing a set of of intervals, I, as well as a set of target links, L_i, for each $i \in I$. Clearly, we cannot construct a bag-of-words vector for each document since the words in each document are not actually known. An alternative strategy is to iterate over all of the intervals within each document and accumulate their link targets as if they were word counts. From there, each document could be represented as a *bag-of-links*, and standard vector space document modeling approaches may be used.

This model treats each document as a bag-of-links vector $v^{(d)}$, where the j^{th} ele-

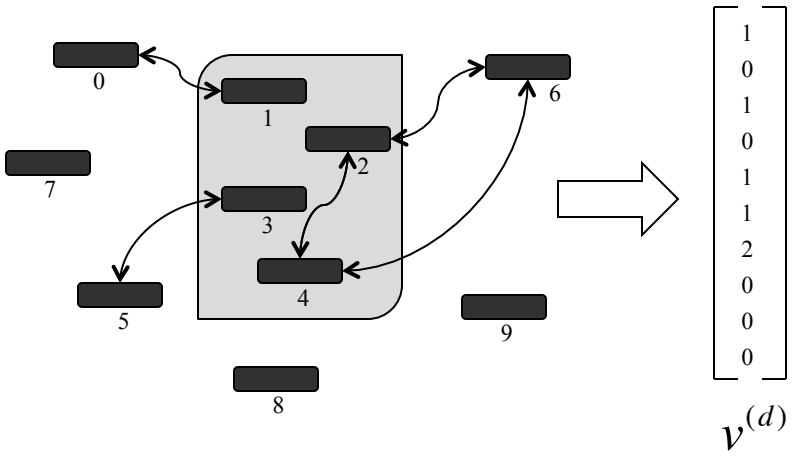

Figure 4-2: Bag-of-links representation of an audio document. Interval 1 is contained inside the document and links to interval 0, so the bag-of-links vector representing the document reflects a count of 1 in its 0th element.

ment of $v^{(d)}$ is equal to the total number of times any interval contained in d matched the j^{th} interval. That is,

$$v_j^{(d)} = \sum_{i \in I_d} \mathbf{1}_{L_i}(j) \tag{4.13}$$

where $\mathbf{1}_{L_i}(j) = 1$ if interval i matched the j^{th} interval and 0 otherwise. This idea for a corpus consisting of 10 match intervals is illustrated in Figure 4-2.

We seek to model the probability of observing a link to interval l from document d using a set of latent topic variables Z:

$$P(l|d) = \sum_{z \in Z} P(l|z)P(z|d). \tag{4.14}$$

The graphical model in plate notation is shown in Figure 4-3, and is in fact equivalent in structure to PLSA. Note that this model assumes that each link originating from d was generated by a different latent topic variable, even if several of these links originate from the same interval of audio within d.

The model parameters, $P(z|d)$ and $P(l|z)$ are estimated using the standard EM update equations for PLSA, enumerated in Eqs. 4.10, 4.11, 4.12, substituting links in place of words. We also apply TF-IDF based stop listing to the bag-of-links vectors, throwing away any interval which was linked to by more than 20% of the documents,

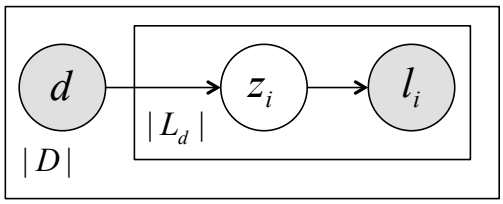

Figure 4-3: The PLSA-BoL model in plate notation

or less than 4 other intervals. This is to eliminate spurious matches, as well as potential stop words, filled pauses, etc. Hierarchical agglomerative clustering of the documents into $|Z|$ clusters is used in the initialization of $P(l|z)$. For this clustering step, the similarity measure used between documents is the cosine similarity between bag-of-links vectors. The initialization is performed by treating each document cluster as an initial topic z, and then finding the maximum likelihood estimate of $P(l|z)$ by only considering the documents clustered into the initial topic z. A pseudo-count of 0.1 for each l is used to smooth the initial $P(l|z)$ distributions before beginning the E-M updates. In these experiments, we assume $|Z| = 6$, the number of true topics, in an effort to re-learn the true topic labels with the latent model.

4.4.2 Summarizing the PLSA-BoL Model

Given a PLSA-BoL model trained on a document collection, we can produce a summary of the topical content of the collection by extracting short audio snippets of speech exemplifying each latent topic. A human user could then listen to these sets of audio snippets and quickly get a gist of what topics are discussed in the collection.

When working with text documents, topic modeling researchers often try to make keyword lists or word clouds reflecting the most significant words in each latent topic learned by a model. To form a summary of a single latent topic z, a common practice is to simply rank all words in the vocabulary according to their posterior probability given the topic, $P(w|z)$. The top N words with the highest posterior probability are then used to form a summary of latent topic z. As an alternative to choosing the words which maximize the posterior probability given a topic, we could instead extract the words which maximize the *weighted point wise mutual information* (WPMI) between

words and topics. The WPMI measure between random variables x and z is defined as:

$$WPMI(x,z) = P(x,z)^{\lambda} \log\left(\frac{P(x,z)}{P(x)P(z)}\right) \tag{4.15}$$

Intuitively, the log( ) factor is large when x and z are more likely to appear together than independently, and the $P(x,z)$ factor weights this by the overall joint probability of x and z. In practice, these factors may weighted more or less against one another by raising them to a power. An example topical summary produced for a collection of text documents from the Fisher corpus by [13] is shown in Table 4.2. In our experiments, to form a summary of latent topic z using the PLSA-BoL model, we rank all of the audio intervals in I according to $WPMI(i,z)$ and extract the top 10 intervals.

4.4.3 PLSA-BoL Experiments

For our summarization experiments, we use a collection of 60 telephone calls from the English Phase 1 portion of the Fisher Corpus [3]. Each call consists of a 10-minute long telephone conversation between two speakers. At the start of each conversation the participants were prompted to discuss a particular topic. The set of calls we use spans 6 of these topic prompts, with 10 calls per prompt. To give a few examples, the prompts for the "Anonymous Benefactor" and "Minimum Wage" topics are shown below:

> "If an unknown benefactor offered each of you a million dollars - with the only stipulation being that you could never speak to your best friend again - would you take the million dollars?"

> Do each of you feel the minimum wage increase - to $5:15 an hour - is sufficient?

SOU posteriorgram representations (described in Chapter 2) for all utterances in all 60 calls were produced by a 45-unit SOU system trained on an independent 60-hour

set of Fisher English data. S-DTW audio segment link detection was applied to the posteriorgram representation of all utterance pairs in the 60 call set. A total of 10,041 link pairs between 3,165 unique audio intervals were discovered and used to train the model, which was set to learn a set of 6 latent topics.

Table 4.3 shows the summaries of the topics learned by the PLSA-BoL model on the 60 call Fisher dataset. The mapping of the latent topics to the true topics, $P(t|z)$, is shown for the closest true topic along with the underlying text transcripts of the intervals extracted to form each latent topical summary. There is a significant overlap between the true topic labels and the latent topics learned, and the extracted intervals are semantically informative with respect to their latent topic. However, many of the intervals forming each topic summary are repeated instances of the same word or phrase. In the next section, we consider a model which has the capability to associate individual match intervals with pseudo-word categories in an effort to alleviate this issue.

Summaries of PLSA Topics	Matching Fisher Topic
dog, cats, pet, animals, fish, bird, feed, puppy, cute, cage	Pets (0.90)
minimum wage, pay, jobs, five fifteen an hour, paid, making, tips	Minimum Wage (0.864)
sports, football, basketball, baseball, game, team, watching, hockey	Sports on TV (0.849)
airport security, plane, fly, september eleventh, flight, airplane, flown	Airport Security (0.523), September 11th (0.351)

Table 4.2: Example latent topic summaries generated using PLSA on text transcripts in [13].

Topic	Text transcripts of extracted intervals	Mapping to true topics (%)
1	minimum wage, minimum wage, minimum wage, minimum wage, ...	Minimum Wage (99.7)
2	think computers, computers, of computers, computer, computer, computers, ...	Computers in Education (99.9)
3	exactly, um, country, exactly, countries, um, countries, exactly, exactly	Illness (37.3), Corporate Conduct (32.2)
4	holidays, holiday, holiday is, holidays, the holidays, holidays, holiday, ...	Holidays (83.1)
5	money, situations, situations, the more money you, friend, educational, four years, situation, situations, make money	Anonymous Benefactor (55.3)
6	weather friends, friends, friends, friends, friends, some friends, friends, kind of friends, to happen, major you know	Corporate Conduct (55.2), Anonymous Benefactor (45.3)

Table 4.3: Latent topic summaries generated using PLSA-BoL.

4.5 The Latent Lexical and Topic Model

4.5.1 Model Overview and Training

While the PLSA-BoL model has the capability to associate links to particular intervals of audio with latent topics, it is not able to infer which intervals of audio may be instances of the same spoken word or phrase. In this section, we introduce a more sophisticated doubly-stochastic model which aims to *jointly* cluster match intervals into pseudo-word categories, and cluster pseudo-word categories into latent topics in a probabilistic fashion. The model assumes that each match interval has a latent word identity $w \in W$, where W is a fixed-size vocabulary of pseudo-words to be learned. The graphical model in plate notation is shown in Figure 4-4, reflecting the generative story assumed for each link set L_i belonging to interval i in document d. For each i in I_d,

1. Draw a latent topic z from $P(z|d)$.

2. Draw a latent pseudo-word w from $P(w|z)$.

3. Draw a set of $|L_i|$ links to other intervals i.i.d. from $P(l|w)$.

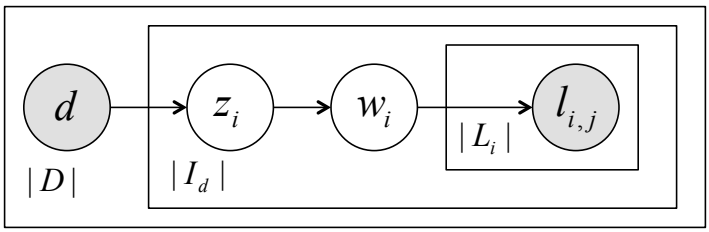

Figure 4-4: The Latent Lexical and Topic Model in plate notation

What differentiates the LLTM from the PLSA-BoL model is the addition of a new set of multinomial distributions, $P(l|w)$. Intuitively, we expect audio intervals containing the same underlying word w to have a tendency to match one another in the S-DTW search. Therefore, $P(l|w)$ should be relatively large for each of these intervals l, and small for intervals not belonging to w. The probability of observing a link set L_i given a document can be expressed mathematically as

$$P(L_i|d) = \sum_{w \in W} \sum_{z \in Z} P(w|z)P(z|d) \prod_{l \in L_i} P(l|w). \tag{4.16}$$

Letting $d(i)$ denote the document containing interval i, the data likelihood can be written as:

$$\mathcal{L} = \prod_{i \in I} \sum_{w \in W} \sum_{z \in Z} P(w|z)P(z|d(i)) \prod_{l \in L_i} P(l|w). \tag{4.17}$$

To find a local maximum of the data likelihood surface, we employ an Expectation-Maximization algorithm. In the E-step, we estimate the joint posterior probability distribution of the pseudo-word variable w and latent topic variable z for interval i appearing in document d. Using Bayes' Rule, we can write this posterior as:

$$P(w, z|d, L_i) = \frac{P(w|z)P(z|d) \prod_{l \in L_i} P(l|w)}{\sum_{z' \in Z} \sum_{w' \in W} P(w'|z')P(z'|d) \prod_{l \in L_i} P(l|w')} \tag{4.18}$$

In the M-step, we use the last estimate of this posterior probability to update the model parameters according to the equations

$$P(l|w) = \frac{\sum_{d \in D} \sum_{i \in I_d} \mathbf{1}_{L_i}(l) \sum_{z \in Z} P(w, z|d, L_i)}{\sum_{l' \in I} \sum_{d \in D} \sum_{i \in I_d} \mathbf{1}_{L_i}(l') \sum_{z \in Z} P(w, z|d, L_i)} \tag{4.19}$$

$$P(w|z) = \frac{\sum_{d \in D} \sum_{i \in I_d} P(w, z|d, L_i)}{\sum_{w' \in W} \sum_{d \in D} \sum_{i \in I_d} P(w', z|d, L_i)} \tag{4.20}$$

$$P(z|d) = \frac{\sum_{i \in I_d} \sum_{w \in W} P(w, z|d, L_i)}{\sum_{z' \in Z} \sum_{i \in I_d} \sum_{w \in W} P(w, z'|d, L_i)} \tag{4.21}$$

Agglomerative clustering of the documents is again used to determine the initial assignment of the topic variable associated with each interval. The $P(l|w)$ and $P(w|z)$ distributions are initialized by pseudo-word category assignments produced by the InfoMap graph clustering algorithm [28] applied to the graph formed by treating the intervals as nodes and their links as edges, although any standard graph clustering algorithm could be used in this step. We take each cluster of intervals and assign it to an initial pseudo-word category w, then find maximum-likelihood initial estimates of $P(l|w)$ and $P(w|z)$ using uniform pseudo-count smoothing added to each element in the support of each distribution.

4.5.2 Summarizing the Topics using the LLTM

The LLTM can be summarized using techniques similar to those used to summarize the PLSA-BoL model, with a few modifications to accommodate the additional latent pseudo-word variables utilized by the model. To summarize a latent topic z using the LLTM, we first rank the pseudo-word categories according to $WPMI(w, z)$ with $\lambda = 0.5$ to choose a representative set of 10 pseudo-words for each latent topic. The choice of λ here was chosen based upon manually examining the summaries produced for various settings of the parameter, but does not seem to have an extremely significant impact on the quality of the summaries produced. We must then extract the interval of audio most representative of each pseudo-word. To do this, we heuristically rank the intervals according to $P(i|w)P(w|d, L_i)$. Here, $P(w|d, L_i)$ represents the posterior probability that interval i belongs to pseudo-word category w, while $P(i|w)$ indicates how likely any interval belonging to pseudo-word category w is to generate a link to interval i.

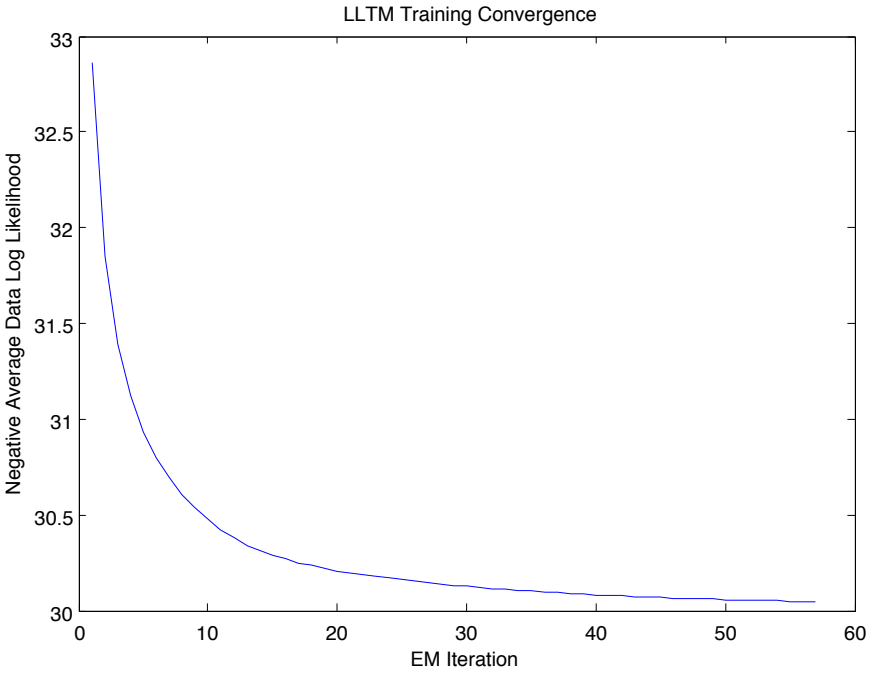

Figure 4-5: Convergence of the LLTM during training.

4.5.3 Experiments and Analysis

In our experiments with the LLTM, we utilized the same 60 call Fisher dataset used to evaluate the PLSA-BoL model, with exactly the same match patterns discovered via S-DTW. Table 4.4 shows the summaries of the learned topics in addition to the mapping of the latent topics to the Fisher topic labels. When training the LLTM model detailed in Table 4.4, we set the number of latent topics equal to the number of true topics, and the number of pseudo-words was set to 581 by the initial clustering provided by InfoMap [28].The final three experimental parameters were the pseudo-count smoothing parameters used when computing initial estimates of the $P(z|d)$, $P(w|z)$, and $P(l|w)$ distributions; these settings were 0.2, 0.02, and 0.01, respectively. The convergence of the data log likelihood is displayed in Figure 4-5.

The LLTM can be used to visually examine the collection of Fisher calls. Figure 4-7 displays a dendrogram of the Fisher conversations where pairwise distances are computed via the cosine distance between their latent topic posterior distributions. Figure 4-6 displays the collection of calls color coded by their dominant latent topic,

along with the text transcriptions of extracted audio summaries of each latent topic.

As can be seen in Tables 4.4, 4.5, and 4.6, the LLTM does largely alleviate the problem of redundant interval selection which was common with the PLSA-BoL model. Many more topically informative words appear in the summaries, although there are still some intervals containing non-informative stop words. Because our system is completely unsupervised top-to-bottom, it does not have the benefit of expert stop lists. Finding a suitable stop word solution is a potential area of improvement for the model.

Topic	Text transcripts of extracted intervals	Mapping to true topics (%)
1	don't think, weather friends, no I, situations, the lottery, very you know, and, benefactor, don't even know who, and, economy, to happen, now um, money, so	Anonymous Benefactor (45.0), Corporate Conduct (27.3)
2	minimum wage, you, yeah I, money, out you'd be, you know, minimum wage jobs, you know people, the, economy, in New York, an hour, five dollars, he has, people working	Minimum Wage (86.1)
3	think computers, if she uses, education, more and, computers, you ah, technical ah, know the computerized, it's just, information, something that's, on there, well that that's, different things, school	Computers in Education (99.7)
4	sicker, C.E.O., stock market, exactly, without the, country, every sick, this guy, in uh in, of cold, like if you, that um, greedy, Zealand you, stomach	Illness (47.7), Corporate Conduct (43.5)
5	I really like, holidays, own holiday, holiday, equality, favorite holiday, considerate, and, the key, recognized, you, new car, keys, like, you like	Holidays (78.8)
6	is actually, I'm twenty, friend, <partial>, I've seen it done, every day, maybe ah, that and all, how, uh, best friend, that's true, increased, children, lazier and	Anonymous Benefactor (72.9)

Table 4.4: Latent topic summaries generated using LLTM.

WPMI	Pseudo-word exemplar
1.2588	think computers
0.7408	if she uses
0.2518	education
0.2260	more and
0.2260	computers
0.2193	you ah
0.2134	technical ah
0.1958	know the computerized
0.1630	(noise)it's just
0.1599	information
0.1599	something that's
0.1599	on there
0.1598	well that that's
0.1516	different things
0.1401	school

Table 4.5: The top audio intervals associated with the top 15 pseudo-words for the latent topic capturing "education", shown with their WPMI scores

Exemplar	Score	Exemplar	Score	Exemplar	Score
think computers	0.019	education	0.073	school	0.184
computers	0.017	education	0.060	schools	0.184
computers	0.016	education	0.060	school	0.147
computer	0.016	education	0.054	school and	0.147
computer	0.015	educational	0.054	school	0.147
computers	0.015	education	0.048	school so	0.110
computers	0.015	indication	0.048	school	0.073
computer	0.015	the educational	0.042	least getting	6e-09
of computer	0.015	education	0.042	know i know	6e-09
a computer	0.015	education	0.036	encyclopedia	6e-09

Table 4.6: The text transcripts of the top 10 audio intervals associated with the first, third, and fifteenth pseudo-word categories for the "education" topic summary shown in Table 4.5. The intervals are scored according to $P(i|w)P(w|d, L_i)$.

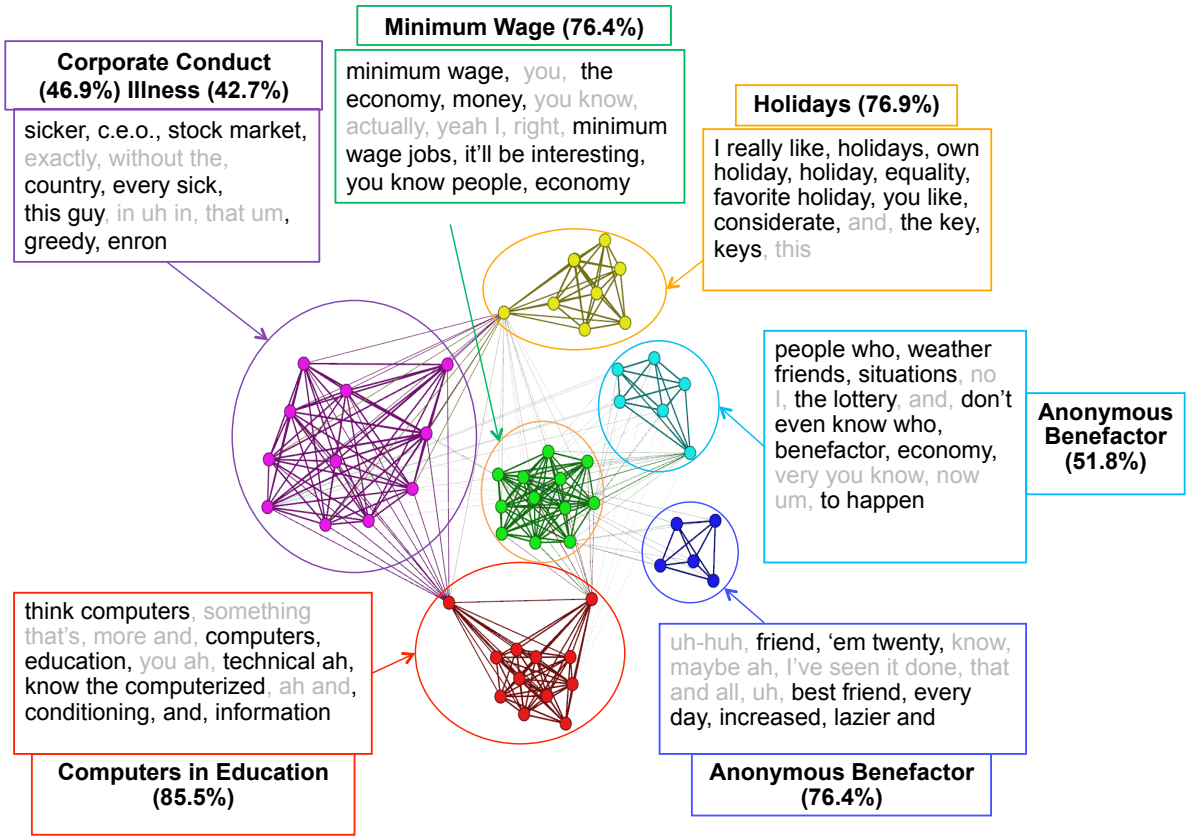

Figure 4-6: A graph displaying the 60 Fisher conversations clustered and color coded by their dominant latent topic. The mapping of each latent topic to its closest true topic is shown in addition to the text transcriptions of a set of extracted short audio snippets summarizing each latent topic.

To more rigorously compare the LLTM with the PLSA-BoL model, as well as to a reasonable baseline, we evaluate the mapping between the learned latent topics and the true topics using the normalized mutual information (NMI) measure:

$$NMI(z,t) = \frac{2 * I(z;t)}{H(z) + H(t)} \tag{4.22}$$

Here $I(\cdot;\cdot)$ denotes mutual information and $H(\cdot)$ denotes entropy. NMI is an information theoretic measure similar to the F-score measure used in detection problems.

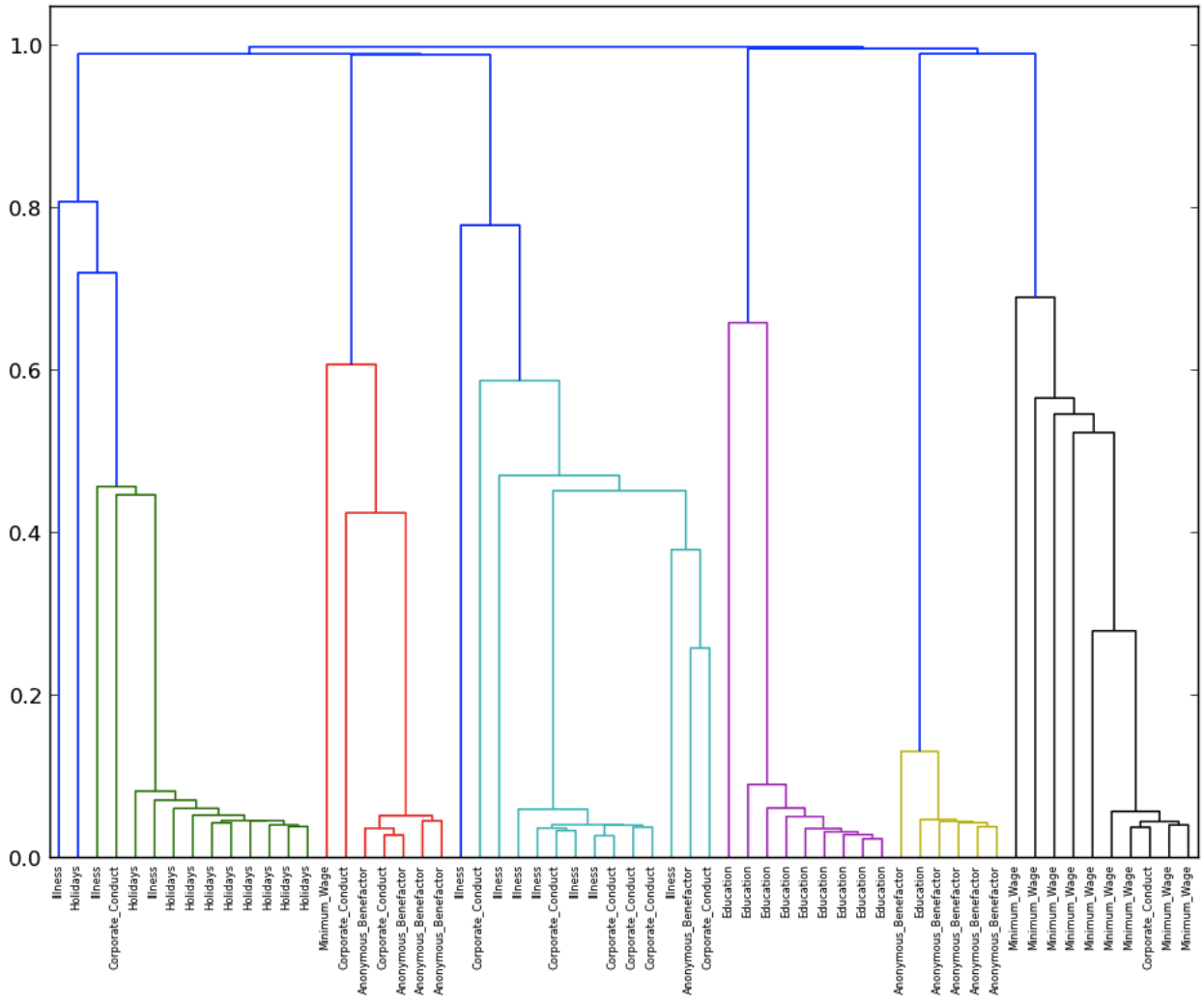

Figure 4-7: Dendrogram formed using the latent topic posterior distributions for the 60 Fisher call collection. Pairwise distances are computed via the cosine similarity between the latent topic distributions of the calls, and the true topic labels are displayed along the bottom.

	Rand.	HAC	PLSA-BoL	LLTM	PLSA-Text
NMI	0.168	0.529	0.529	0.592	0.895

Table 4.7: NMI scores for the various models

Its value ranges between 0 and 1, with 1 representing a perfect mapping between the true topics and the latent topics. Table 4.7 shows the NMI scores for a uniform random assignment of documents to latent topics, the hard agglomerative clustering (HAC) used to initialize all models, both latent models, and a phrase-based PLSA model applied to text transcripts of the data [13]. While the PLSA-BoL model does not beat the hard-clustering baseline, the LLTM significantly outperforms them both, closing 17.2% of the performance gap between the baseline and the text-based system.

4.6 Chapter Summary

This chapter began by providing a brief overview of vector space text document modeling techniques. We then motivated our next steps by describing a scenario in which we wish to apply latent document models to spoken audio documents, but have no access to a speech recognizer or labeled data. A method of collapsing acoustic patterns discovered by a segmental dynamic time warping-like algorithm into pseudo-terms which may be treated like word occurrences was introduced. We then described a latent model similar to PLSA which operates on document vectors representing the degree of similarity between the acoustic patterns located in a document to all of the acoustic patterns contained within a dataset. Experimental results confirm that this model is able to relate instances of semantically similar acoustic patterns into topical categories, but is unable to relate instances of lexically similar patterns to one another. We introduced a novel doubly-stochastic topic model which takes into account the latent lexical identity of each discovered acoustic pattern, and present the E-M update equations for the model; methods for summarizing the topics learned by such a model are also described. Experimental results on a corpus of 60 topically-labeled telephone calls from the Fisher corpus are presented, demonstrating the ability of the model

to jointly infer latent lexical identities for each acoustic pattern interval, as well as latent topics covering the set of latent lexical identities.

Chapter 5

Conclusion

5.1 Summary of Contributions

In Chapter 1, we have described and motivated the so-called "zero-resource" learning problem in speech research and detailed previous work in this field. We presented the subproblem of unsupervised spoken audio corpus analysis, as well as a novel framework for tackling the problem. In Chapter 2, we presented an overview of some recent research efforts in the realm of unsupervised acoustic modeling and motivated its importance to our application. In Chapter 3, we described background work on segmental dynamic time warping based pattern discovery as well as more recent improvements to the algorithms. We then gave a detailed explanation of the implementation that was used in our experiments.

The main contributions of this thesis are predominantly detailed in Chapter 4. Namely, we explore several ways of characterizing the latent topical content of a collection of speech audio in a completely unsupervised fashion. We apply the standard PLSA model to a novel representation of spoken audio documents, and then formulate a novel probabilistic topic model which explicitly models the underlying, unknown lexical identity of each speech audio interval discovered by a S-DTW-like algorithm. We demonstrate the model's utility by applying it to a 10-hour collection of Fisher English telephone calls and showing that the learned topics (and hence the implied clustering of the documents) highly overlap the true topic labels. Furthermore, we

demonstrate the ability of the model to produce highly informative yet very concise summaries of the latent topics, enabling a human listener to get the gist of the topical themes of the hours-long collection in less than a minute.

5.2 Future Directions

We have presented a system capable of taking as its input nothing more than a collection of unlabeled, untranscribed spoken audio documents in a possibly unknown language and producing topically informative audio summaries of the entire collection. It does this by discovering acoustic repetitions throughout the data, inferring an underlying word category to each repetition, as well as inferring a latent topic distribution associated with each underlying word category and each document as a whole. We suggest here several key ways in which the methodology may be improved.

5.2.1 Improvements in Speed and Scalability

Possibly the most important question that future work may tackle is that of scalability. Even taking into account the algorithmic optimizations we utilize for the pattern discovery step, the computational complexity still remains $O(n^2)$. Jansen and Van Durme suggest an approximation algorithm based upon locality sensitive hashing which computes an approximation of the distance matrix in $O(n\ logn)$ time, which would facilitate the application of our methodology to larger datasets.

Another method of improving the speed and scalability of the proposed system lies within the considerable processing power of massively parallel graphics processing units (GPUs). Zhang and Glass demonstrated that a S-DTW-based keyword spotting algorithm could be efficiently implemented on a GPU, resulting in a 55x speedup when compared to a CPU implementation. The implementation of a similar algorithm capable of an exhaustive S-DTW search would likely see similar gains. In addition to improving the speed at which acoustic patterns may be discovered, the E-M updates for the LLTM may also benefit from being parallelized on a GPU.

5.2.2 Improvements in Representation of Acoustics

Because the focus of our work was on modeling the underlying lexical and topical identities of a collection of short speech intervals, we did not do an exhaustive comparison of different feature representations of the acoustics. However, some investigation into this arena has revealed that different acoustic representations do in fact have a large influence over the capability of a DTW-based pattern discovery algorithm to uncover high precision matches [15]. In the extreme case, perfect precision and recall of a pattern discovery algorithm would be nearly identical to achieving perfectly accurate speech recognition. Therefore, any downstream processing of the discovered patterns, such as the models described in this paper, would greatly benefit from higher quality matching.

5.2.3 Improvements in Topic and Word Modeling

One challenging issue facing unsupervised lexical modeling is the stop word problem. Although the topical summaries produced by our models contain mostly informative audio snippets, they are often peppered with instances of uninformative function words, filled pauses, disfluent speech, and so on. Although we do employ a TF-IDF based stop listing procedure in an attempt to remove these words, it is clear that there is room for improvement. Supervised text-based systems often employ expert stop word lists whose sole purpose is to filter out this sort of "junk," and it is reasonable to consider whether unsupervised measures may be developed to serve this purpose.

Refined probabilistic graphical models which rely on fewer parameters would be faster to train while simultaneously requiring less data. It is also conceivable that phoneme-like-unit segmentation within each discovered pattern, such as that introduced by [24], could be used to infer a set of pronunciations for each discovered pattern and remove the need to model every possible pairwise link between the match intervals. The latent topic and word levels of the model could also be integrated into the Bayesian acoustic unit discovery framework presented by [25].

5.2.4 New Application Areas

Although we have made the assumption that our method is language agnostic, we have thus far only applied it to English spoken audio. In order to validate this assumption, it is necessary to evaluate our methods on non-English data. It may be particularly interesting to choose a resource impoverished language for which ASR technology is nonexistent.

5.3 Parting Thoughts

Automatic speech recognition technology continues to improve and is now becoming so widespread that most of us carry ASR-capable smartphones everywhere we go. However, it is important to remember that of the approximately 7,000 human languages spoken across the world, only an estimated 50 to 100 possess a sufficient amount of labelled data to train a recognizer; it doesn't seem fair to simply ignore those languages and their speakers. Furthermore, by focusing our efforts on developing a unified, generalizable framework to teach computers to recognize and understand human language on their own, it is likely that researchers will reach a deeper understanding of how humans learn. While building speech systems less reliant on expert knowledge is by no means an easy task, it opens up new avenues of research filled with exciting new problems waiting to be solved.

Bibliography

[1] A. Andreou, T. Kamm, and J. Cohen, "Experiments in vocal tract normalization," *CAIP Workshop: Frontiers in Speech Recognition II,* 1994.

[2] D. Blei, A. Ng and M. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research* vol. 3, pp. 993-1022, 2003.

[3] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: A resource for the next generation of speech-to-text," in *Proc. of International Conf. on Language Resources and Evaluation,* Lisbon, May 2004.

[4] M. Dredze, A. Jansen, G. Coppersmith, and K. Church, "NLP on spoken documents without ASR," in *Proc. of EMNLP,* 2010.

[5] R. Flamary, X. Anguera, and N. Oliver, "Spoken wordcloud: Clustering recurrent patterns in speech," in *Proc. of CBMI,* 2011.

[6] A. Garcia and H. Gish, "Keyword spotting of arbitrary words using minimal speech resources," in *Proc. of ICASSP,* Toulouse, 2006.

[7] J. Glass, "Towards Unsupervised Speech Processing," Keynote, *Proc. ISSPA*, Montreal, July 2012.

[8] S. Goldwater, T. Griffiths, and M. Johnson, "A Bayesian framework for word segmentation: exploring the effects of context," *Cognition,* vol. 112 pp. 21-54, 2009.

[9] T. Hazen, "Direct and latent modeling techniques for comparing spoken document similarity," *in Proc. of IEEE Workshop on Spoken Language Technology,* 2010.

[10] T. Hazen, "Latent topic modeling for audio corpus summarization," in *Proc. of Interspeech,* Florence, August 2011.

[11] T. Hazen, M. Siu, H. Gish, S. Lowe, and A. Chan, "Topic modeling for spoken documents using only phonetic information," in *Proc. of ASRU,* 2011.

[12] T. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. ASRU,* 2009.

[13] T. Hazen and F. Richardson, "Modeling multiword phrases with constrained phrase trees for improved topic modeling of conversational speech," in *IEEE Spoken Language Technology Workshop,* Miami, December 2012.

[14] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. of Conf. on Uncertainty in Artificial Intelligence,* Stockholm, 1999.

[15] Aren Jansen, Emmanuel Dupoux, Sharon Goldwater, Mark Johnson, Sanjeev Khudanpur, Kenneth Church, Naomi Feldman, Hynek Hermansky, Florian Metze, Richard Rose, Michael Seltzer, Pascal Clark, Ian McGraw, Balakrishnan Varadarajan, Erin Bennett, Benjamin Borschinger, Justin Chiu, Ewan Dunbar, Abdellah Fourtassi, David Harwath, Chia-ying Lee, Keith Levin, Atta Norouzian, Vijayaditya Peddinti, Rachael Richardson, Thomas Schatz, and Samuel Thomas, "A Summary of the 2012 CLSP Workshop on Zero Resource Speech Technologies and Models of Early Language Acquisition," to appear in *Proc. of ICASSP,* 2013.

[16] A. Jansen, S. Thomas, and H. Hermansky, "Weak top-down constraints for unsupervised acoustic model training," to appear in *Proc. of ICASSP,* 2013.

[17] A. Jansen and K. Church, "Towards unsupervised training of speaker independent acoustic models," in *Proc. of Interspeech,* Florence, 2011.

[18] A. Jansen, K. Church and H. Hermansky, "Towards spoken term discovery at scale with zero resources," in *Proc. of Interspeech,* Makuhari, September 2010.

[19] A. Jansen and B. Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Proc. of ASRU,* 2011.

[20] A. Jansen and B. Van Durme, "Indexing raw acoustic features for scalable zero resource search," in *Proc. of Interspeech,* 2012.

[21] M. Johnson, "Unsupervised word segmentation for Sesotho using adaptor grammars," in *Proc. ACL SIG on Computational Morphology and Phonology,* 2008.

[22] K. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation,* 28:(1) 11-21, 1972.

[23] L. Lamel, J. Gauvain, and G. Adda, "Unsupervised acoustic model training," in *Proc. of ICASSP,* 2002.

[24] A. Lee and J. Glass, "A comparison-based approach to mispronunciation detection," in *Proc. of SLT*, 2012.

[25] C. Lee and J. Glass, "A nonparametric Bayesian approach to acoustic model discovery," in *Proc. of ACL,* Jeju, 2012.

[26] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language,* 1995.

[27] A. Park and J. Glass, "Unsupervised pattern discovery in speech," *IEEE Trans. on Audio, Speech, and Language Processing,* vol. 16, no. 1, pp. 186-197, 2008.

[28] M. Rosvall and C.T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. of National Academy of Science, USA,* vol. 105 pp. 1118-1123, 2008.

[29] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, "A Bayesian approach to filtering junk e-mail," in *AAAI Workshop of Learning for Text Categorization,* 1998.

[30] T. Sainath, B. Ramabhadran, D. Nahamoo, D. Kanevsky, D. Van Compernolle, K. Demuynck, J. Gemmeke, J. Bellegarda, S. Sundaram, "Exemplar-Based Processing for Speech Recognition: An Overview," in *IEEE Signal Processing Magazine* 29(6): 98-113, 2012.

[31] M. Siu, H. Gish, S. Lowe, A. Chan, "Unsupervised audio pattern discovery using HMM-based self-organized units," in *Proc. of Interspeech,* 2011.

[32] P. Woodland, S. Young, "The HTK tied-state continuous speech recognizer," in *Proc. of Eurospeech,* 1993.

[33] B. Varadarajan, S. Khudanpur, and E. Dupoux, "Unsupervised learning of acoustic sub-word units," in *Proc. of ACL-08 HLT, Short Papers,* pp. 165-168, 2008.

[34] Y. Zhang, K. Adl, and J. Glass, "Fast spoken query detection using lower-bound dynamic time warping on graphical processing units," in *Proc. of ICASSP*, Kyoto, March 2012.

[35] Y. Zhang and J. Glass, "Towards multi-speaker unsupervised speech pattern discovery," in *Proc. of ICASSP,* Dallas, March 2010.

[36] Y. Zhang and J. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. ASRU,* 2009.

[37] Y. Zhang, R. Salakhutdinov, H. Chang, and J. Glass, "Resource configurable spoken query detection using deep Boltzmann machines," in *Proc. of ICASSP,* Kyoto, March 2012.

[38] X. Zhu, G. Penn, and F. Rudzicz, "Summarizing multiple spoken documents: finding evidence from untranscribed audio," in *Proc. of ACL,* Singapore, August 2009.