Unsupervised Pattern Discovery in Speech: Applications to Word Acquisition and Speaker Segmentation

by

Alex Seungryong Park

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

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2006

Author .....
Department of Electrical Engineering and Computer Science
September 30, 2006

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

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Unsupervised Pattern Discovery in Speech: Applications to Word Acquisition and Speaker Segmentation

by Alex Seungryong Park

Submitted to the Department of Electrical Engineering and Computer Science
on September 30, 2006,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

We present a novel approach to speech processing based on the principle of pattern discovery. Our work represents a departure from traditional models of speech recognition, where the end goal is to classify speech into categories defined by a pre-specified inventory of lexical units (i.e. phones or words). Instead, we attempt to discover such an inventory in an unsupervised manner by exploiting the structure of repeating patterns within the speech signal. We show how pattern discovery can be used to automatically acquire lexical entities directly from an untranscribed audio stream.

Our approach to unsupervised word acquisition utilizes a segmental variant of a widely used dynamic programming technique, which allows us to find matching acoustic patterns between spoken utterances. By aggregating information about these matching patterns across audio streams, we demonstrate how to group similar acoustic sequences together to form clusters corresponding to lexical entities such as words and short multi-word phrases. On a corpus of academic lecture material, we demonstrate that clusters found using this technique exhibit high purity and that many of the corresponding lexical identities are relevant to the underlying audio stream.

We demonstrate two applications of our pattern discovery procedure. First, we propose and evaluate two methods for automatically identifying sound clusters generated through pattern discovery. Our results show that high identification accuracy can be achieved for single word clusters using a constrained isolated word recognizer. Second, we apply acoustic pattern matching to the problem of

speaker segmentation by attempting to find word-level speech patterns that are repeated by the same speaker. When used to segment a ten hour corpus of multi-speaker lectures, we found that our approach is able to generate segmentations that correlate well to independently generated human segmentations.

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

Acknowledgments

Having completed the writing of this thesis document, it is both a relief and a pleasure to give thanks to the many individuals and organizations who have advised, encouraged, and supported me throughout my education.

First and foremost, I am grateful to my advisor, Jim Glass, for his invaluable guidance and support throughout the course of my doctoral research. I am indebted to him for introducing me to spoken language research and for giving me the freedom to explore a wide variety of topics during my time as a student. In so doing, he has taught me how to grapple with the difficult problem of when to dig deeper, and when to dig elsewhere.

Many thanks to my thesis committee: Victor Zue and Regina Barzilay, who gave me lots of excellent feedback and detailed comments about this work. Victor encouraged me to keep a larger audience in mind with my work, while Regina pointed out many ways that the research and the writing could be strengthened. Victor and Regina both provided much-needed perspective on this work and I feel fortunate to have benefited from their experience and knowledge.

I am grateful to all of the members of the Spoken Language Systems Group for making work such a collegial and enjoyable experience. In particular, T. J. Hazen advised me for my master's thesis and I have continued to count on him as an unofficial second advisor ever since. He is an amazing resource on everything recognition related and an extraordinary researcher. I am indebted to Lee Hetherington and Scott Cyphers, who (among others) have created an excellent computational infrastructure and library of research tools that were indispensable for much of the work in this thesis.

Karen Livescu, Han Shu, and Min Tang, all recent graduates of the SLS group provided much advice and many interesting research discussions. Special thanks also to Jon Yi, for answering so many of my questions and for contributing many great ideas that shaped some of the work in this thesis. Thanks also to the other present and former SLS students that I have been fortunate enough to work with: Ken Schutte, Mitchell Peabody, Tara Sainath, John Lee, Ed Filisko, Ghinwa Choutier, Chih-yu Chao, Alex Gruenstein, and Ernie Pusateri. I would also like to thank my office mates, Han-Pang Chiu, Claire Monteleoni, and Luis Berra, for making it a pleasure to come to work every day.

I am grateful to the members of the MIT faculty who have influenced my education throughout my time here. Professors Sara Billey and Don Troxel recommended me to the Ph.D. program. Professor Munther Dahleh was extremely helpful to me as my graduate academic counselor. Professors Denny Freeman and Ken Stevens served on my Research Qualifying Examinations committee. I would also like to thank Professors Duane Boning and Munther Dahleh for allowing me to develop my teaching skills as a teaching assistant in their classes.

Outside of MIT, my education was enriched by various summer internships. I worked as an intern at Speechworks (now Nuance) in 2002, where I was fortunate to be mentored by Johan Schalkwyk, who helped me to understand the OSR system and provided plenty of interesting problems to work on and papers to read. I spent the summer of 2003 as a research intern at ATR in Kyoto in the Spoken Language Translation group headed by Satoshi Nakamura. Thanks to Frank Soong, Tor Myrvoll, and Sakriani Sakti for many interesting lunchtime discussions and for providing some perspective on the history of speech research.

Thanks to my friends, roomates, and fellow students, who have made my life outside of lab fun and enjoyable: Craig Gerardi, Bobby Middleton, Dora Kelle, Cindy Zia, Fong Keng, Jolie Chang, Nori Yoshida, Chia Hao La, Sybor Wang, Xuemin Chi, Keith Santarelli, Kazutaka Takahashi, Tin Kyaw, Danish Khatri, Roy Lee, Oak Son, Agatha Lee, Alex Andalis, Rich Possemato, Albert Chan, Karyn Lu, Albert Huang, Des Adler, Annie Kim, and many others.

I am extremely grateful to Jane Wu for all of her patience, encouragement, and advice over the past several years. She has been a constant source of support, both in research and in life, cheering me on when things were good, and cheering me up when they were not.

Finally, I would like to express my deepest gratitude to my immediate family for their unconditional love and support. My brothers, Justin and Keith, keep me strong with the knowledge that I can always count on them. Over the years, my parents have been my guardians, role models, teachers, and friends, somehow knowing which I needed most. In difficult times, my mom has always been there to help me take things in stride, offering philosophical advice and comfort (much of it prefaced by: "You know, Buddha says ..."). My dad, who I credit for introducing me to math and science, has always believed in me, has never demanded of me, and has intuitively known that the most comforting way to start any phone conversation was: "Let's not talk about anything related to when you're going to graduate...". Mom and Dad, everything I am, I owe to you. Thank you.

Support for this research was provided in part by the National Science Foundation under grant #IIS-0415865.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Every day, humans communicate with each other using the shared code of spoken language. Although opinions differ as to the degree of innate language learning *ability* possessed by humans [11, 21, 25, 66], the degree of variation in languages across different cultures indicates that linguistic knowledge itself is acquired through interaction with the environment, and through exposure to spoken language [45, 58, 57, 91]. Because of the natural and interactive way in which we learn how to speak, we rarely give thought to the mechanisms behind this development. Despite much study, the mechanisms through which humans learn the intricacies of language and acquire an inventory of words is still not well understood.

The process of human language acquisition is of interest to us because it represents an existence proof in our pursuit of speech recognition and understanding by machines. Over the last several decades, significant progress has been made in developing automatic speech recognition (ASR) systems which

are now capable of performing large vocabulary continuous speech recognition [87, 37, 24, 68, 48]. In spite of this progress, the underlying paradigm of most approaches to speech recognition has remained the same. The problem is cast as one of classification, where input data (speech) is segmented and classified into a pre-existing set of known categories (words). Discovering where these word entities come from is typically not addressed.

At the algorithmic level, the prevailing methodology used in most speech recognition systems is both *supervised* and *static*. By supervised, we mean that the systems rely on an abundance of labeled training data, consisting of actual speech audio data along with human transcriptions of the underlying orthography. While some systems perform semi-supervised learning by using automatically labeled data in addition to human transcribed data for training, the overall paradigm is still one in which the machine learns how to map input data to a set of predetermined labels. By static, we mean that the parameters and settings of the speech recognition system are specified and learned prior to deployment, and are not modified thereafter. Although systems use adaptation to adjust system parameters in response to the recognition input, recognition output is mostly determined by what is learned in the initial training and design phase.

While this prevailing approach to speech recognition has led to steadily decreasing word error rates in the past, it has also resulted in systems that are vulnerable to different types of mismatch that constitute many of the difficulties faced by the ASR community. The out-of-vocabulary (OOV) word problem is an example of mismatch caused by differences in the chosen lexicon and the set of words observed during testing [8]. Inconsistent recognition performance for different speakers can be attributed to the mismatch between the acoustics of speakers in the training data and the observed speaker [49]. Even the phenomenon of speech recognizers hypothesizing English word sequences in response to foreign language speech can be classified as a mismatch issue. Additional sources of mismatch that can dramatically affect speech recognition accuracy include environmental conditions [44], speaking style [3], and language usage [87].

Problems due to mismatch aside, there are more philosophical concerns with the prevailing model of recognition that we have described above. The level of supervision required for designing and training a recognition system is intuitively unsatisfying when one considers the ability of humans to learn spoken language without having access to labeled corpora of speech training data. In addition to automatically determining where word boundaries occur through exposure to continuous speech, humans are also able to detect unknown words and add them to their vocabulary so that they can be used and understood in subsequent conversations. In other words, learning and application of learned knowledge are not separated – one enforces the other. However, for the most part, automatic speech recognizers can only be *trained*, and are not inherently designed to learn from the data they are meant to classify.

In this thesis, we depart from the traditional model of speech recognition that we have described so far to consider the idea of speech processing from the perspective of pattern discovery. Instead of focusing on the recognition problem, that is, the organization of speech into categories derived from a pre-existing inventory of lexical units, we attempt to discover this inventory automatically by exploiting structure within the speech signal.

1.2 Background

Spoken language exhibits structure at many levels. Ideas and concepts are composed of phrases, which are composed of words, which are in turn composed of phonological units. On the surface however, continuous speech is a simple time varying acoustic signal, seldom including pauses between words or consistent cues to indicate phrase boundaries. In the process of acquiring facility with spoken language, one of the major challenges infants must overcome is the problem of how to segment and organize the raw speech signal into the patterns that form the building blocks of language. Our goal in this work is to overcome the same challenge algorithmically. The inspiration for our unsupervised approach to speech processing comes from two sources, one directly related to the study of human language acquisition, and the other in a field far removed from speech.

The first source of inspiration for our approach is conceptual in nature and concerns the human language acquisition process. Developmental psychologists studying infant language learning have devoted considerable effort to understanding how word entities can be discerned when speech is presented as a continuous stream of diverse sounds. Saffran *et al.* found that infants are able to detect the statistical properties of syllable co-occurrence, presumably to aid in word boundary detection [100, 99]. In an experiment involving 8-month old infants, the researchers presented the subjects with continuous streams of coarticulated consonant-vowel syllables. Four three-syllable nonsense word entities were defined and used in random sequences to generate the speech stream. Aside from the differences in syllable transitional probabilities that resulted from the word structure in the speech stream, no other acoustic or prosodic cues for word boundary detection were present in the speech signal. After only two minutes of exposure, the infants were able to distinguish between syllable sequences corresponding to word entities and those composed of random three-syllable sequences. The experiments provide evidence that humans use information about recurring patterns in speech to acquire words, which was one of the initial factors that motivated our work.

Our second source of inspiration is implementational in nature and is directly related to current research in bioinformatics [96, 12], particularly comparative genomics. The goal of much research in comparative genomics is to find patterns corresponding to genes, regulatory regions, and structurally important

sequences from massive amounts of genomic DNA or protein sequence data. Unlike speech, the lexicon of interesting subsequences is often not known ahead of time, so these items must be discovered from the data directly. Pattern discovery is made possible by the observation that functionally important biological sequences are more likely to be preserved across the genomes of different specimens than non-essential sequences. By aligning sequences to each other and identifying patterns that repeat with high recurrence, these preserved sequences can be readily discovered. Since there are a finite inventory of fundamental units that comprise a biological sequence – nucleic acids for DNA and amino acids for proteins – alignment reduces to simple string matching with penalties for insertion, deletion, and substitution.

We can apply similar observations from comparative genomics back to speech. That is, patterns of speech sounds are more likely to be consistent within word or phrase boundaries than across. By aligning continuous utterances to each other and finding similar sequences, we can potentially discover frequently occurring words with minimal knowledge of the underlying speech signal. The fundamental assumption in this approach is that acoustic speech data displays enough regularity to make finding matches possible. In order to bring the speech problem closer to the bioinformatics formulation, a simplifying abstraction would be to transform the speech data into an intermediate representation resembling a biological sequence by using a phonetic recognizer to output a sequence of phone units. As we will discuss in the next section, this technique has been attempted by researchers in previous approaches to the word acquisition problem. However, this type of translation is sensitive to the training data and units used in the phonetic recognizer, and also requires supervision in the specification of elementary subword units. Instead, we describe an approach that uses a modification of dynamic time warping to directly compare utterances against each other at the acoustic level.

1.3 Related Work

There have been a variety of research efforts that are related to the work presented in this thesis. We can roughly categorize these works into two major groups: applications of pattern discovery principles to domains outside of natural language processing, and unsupervised learning techniques within the field of natural language processing.

1.3.1 Pattern Discovery

The works summarized in this section represent a variety of different fields, ranging from computational biology to music analysis to multimedia summarization. There is a common underlying theme in all of this research: the application of

pattern discovery principles to sequence data. We briefly describe work in each of these fields below.

In computational biology, research in pattern discovery algorithms is motivated by the problem of finding *motifs* (biologically significant recurring patterns) in biological sequences. Although the large body of proposed approaches is too large to list here, a survey of the more important techniques is described in [104] and [32]. The class of algorithms most relevant to our work are based upon sequence comparison, where multiple sequences are compared to one another to determine which regions of the sequence are recurring. Since biological sequences can be abstractly represented as strings of discrete symbols, many of the comparison techniques have roots in string alignment algorithms. In particular, a popular approach to alignment is the use of dynamic programming to search an edit distance matrix (also known as a distance matrix, position weight matrix, or position specific scoring matrix) for optimal global alignments [78, 47] or optimal local alignments [109, 119]. The distance matrix is a structure which generates a distance or similarity score for each pair of symbols in the sequences being compared. We make use of distance matrices for alignment in this thesis as well, although the sequences we work with are derived from the audio signal, and are therefore composed of real-valued vectors, not discrete symbols.

Distance matrices are also used extensively by researchers in the music analysis community. In this area of research, the music audio is parameterized as a sequence of feature vectors, and the resulting sequence is used to create a self-distance matrix. The structure of the distance matrix can then be processed to induce music structure (i.e., distinguish between chorus and verse), characterize musical themes, summarize music files, and detect duplicate music files [69, 16, 18, 26, 46, 88]. We carry over the use of distance matrices for pattern discovery in music audio to our own work in speech processing.

Recently, Xie [122] proposed a framework for discovering temporal patterns in multimedia streams using a graphical modeling approach. The task of pattern discovery was cast as one of finding the optimal parameters of a hierarchical hidden Markov model (HMM), where the recurring patterns are modeled as HMMs connected to each other via transitions in a higher-level Markov chain. Xie performed pattern discovery experiments on broadcast video of speech and news using relatively coarse features such as dominant color intensity and motion intensity for video, and volume, zero crossing rate, and spectral roll-off for audio. The types of patterns found corresponded to relatively high-level temporal phenomena such as break segments or play segments in sports matches.

1.3.2 Unsupervised Language Acquisition

Another major category of related work is research concerning unsupervised learning in the area of natural language processing. We summarize recent work

in this area in the context of whether the work is done at the supralexical, sublexical, or lexical levels.

Grammar Induction

At the supralexical level, the problem of unsupervised language acquisition has been well-studied by many researchers in the field of natural language understanding [123, 90, 22, 15, 17]. The primary objective of much of this research has been the statistical induction of syntactic structure from unannotated text. Solan *et al.* have developed a system for producing structured grammars from unparsed and unannotated text [111, 110]. This work is conceptually similar to our own in that grammar induction is cast as a problem of pattern discovery, except that words are used as elementary units. More recently, Klein has proposed statistical methods for unsupervised dependency and phrase structure parsing from corpora of sentences that are represented as sequences of word classes (such as parts of speech and semantic fields) [61].

The work mentioned above has focused primarily on the domain of language acquisition in the sense of understanding and grammar, with the assumption of a pre-existing lexicon. In each of the cited examples, the system or algorithm has taken words or word classes in textual form as input, making such approaches unsuitable for direct application to untranscribed speech data.

Subword Unit Selection

At a lower level of granularity than the grammar induction task is the problem of automatically finding sublexical patterns in speech data. Researchers in the speech recognition community have devoted considerable effort to this task, which is primarily motivated by the goal of determining units for automatic speech recognition [40, 6, 53, 106, 62, 114, 113, 82]. The majority of proposed approaches consist of an acoustic segmentation phase to break the speech signal into short segments, followed by a clustering phase to group the segments into syllable- or phoneme-size units. Recent work in this area includes research by Bacchiani, who introduced an automatic method for deriving subword units and word pronunciations directly from speech segmented at the word level [35, 5, 4]. In that work, subword units were determined by first performing acoustic segmentation on word-length utterances, then clustering the resulting segments using a maximum likelihood objective function. This procedure resulted in a unit inventory along with a lexicon of pronunciations based on these units.

Although we briefly consider the problem of pattern discovery at the subword level in Chapter 4, the majority of the work in this thesis is concerned with finding word and phrase level patterns. However, the method we use for word discovery can easily be modified to find patterns of shorter duration by chang-

ing parameters used for sequence alignment (described in Chapter 3). In this context of finding subword units, our approach differs from the other techniques described in this section primarily in the acoustic segmentation phase. Whereas other approaches use local acoustic measures such as spectral change to segment speech as a preprocessing step, our algorithm is based upon simultaneously finding recurring segments as well as the corresponding boundaries through direct comparison to other speech segments.

Word Acquisition

The area of unsupervised language learning research that is most closely related to our work concerns the problem of knowledge acquisition at the lexical level.

For text processing, Ando and Lee have presented an unsupervised method for segmenting continuous text in languages such as Japanese or Chinese, where textual word boundaries are typically absent [2]. In that work, character n-gram frequencies were used to determine word boundary placement. Boundaries are more likely to be placed at points where the n-gram frequency of a segment is less than the n-gram frequency of adjacent segments. This algorithm is based on the observation that n-grams which cross word boundaries occur more rarely than n-grams which are found within words.

Beyond text processing, interest has grown in the area of unsupervised word acquisition from continuous speech, with notable work being performed by Roy, de Marcken, Brent, and Venkataraman. We briefly describe each of these works below.

Most recently, Roy *et al.* have proposed a model for lexical acquisition by machine using multimodal inputs, including speech [97, 98]. Roy used a recurrent neural network trained on transcribed speech data to output a stream of phoneme probabilities for phonemically segmented audio. Words were learned by pairing audio and visual events and storing them as lexical items in a long term memory structure.

In [29] and [28], de Marcken demonstrated how to learn words from phonetic transcriptions of continuous speech by using a model-based approach to lexicon induction. The algorithm iteratively updates parameters of the model (lexicon) to minimize the description length of the model given the available evidence (the input corpus).

Brent proposed a model-based dynamic programming approach to word acquisition by considering the problem as one of segmentation (i.e., inferring word boundaries in speech) [13, 14]. In his approach, the input corpus is presented as a single unsegmented stream. The optimal segmentation of the corpus is found through a dynamic programming search, where an explicit probability model is used to evaluate each candidate segmentation. A similar strategy is used by

Venkataraman in [117], although the utterance level representation of the corpus is used as a starting point rather than viewing the entire corpus as a single entity. The estimation of probabilities used in the segmentation algorithms of Brent and Venkataraman differ, but the overall strategies of the two techniques are conceptually similar.

We note here that each of the above examples used a phonological lexicon as a foundation for the word acquisition process, and none of the techniques described were designed to be applied to the speech signal directly. The algorithms proposed by de Marcken and Roy, both depend on a phonetic recognition system to convert the continuous speech signal into a set of discrete units. The systems of Brent and Venkataraman were evaluated using speech data phonemically transcribed by humans in a way that applied a consistent phoneme sequence to a particular word entity, regardless of pronunciation.

1.4 Contributions

The primary focus of this work concerns the unsupervised processing of speech data to automatically extract words and linguistic phrases. Our work differs substantially from other approaches to unsupervised word acquisition in that it operates directly on the acoustic signal, using no intermediate recognition stage to transform the audio into a symbolic representation. Although the inspiration for our methods is partially derived from experiments in developmental psychology, we make no claims on the cognitive plausibility of these word acquisition mechanisms in actual human language learning. While we attempt to associate discovered word entities with known vocabulary items, we refrain from attempting to associate meaning with these words.

The main contributions of this thesis are summarized below:

(1) We demonstrate how to find subsequence alignments between the spectral representations of pairs of continuous utterances. In so doing, we propose a variation of a well-known dynamic programming technique for time series alignment, which we call segmental dynamic time warping (DTW). This task is motivated by the assumption that common words and phrases between utterance pairs are likely to be acoustically similar to each other.

(2) We characterize the matching accuracy of subsequence alignments that are found via segmental DTW along the dimensions of alignment duration and average distortion. At both the word level and phone level, we demonstrate that lower distortion and longer duration are indicative of correct matches.

(3) We show how recurring speech patterns in an audio stream can be found and clustered together by representing the audio stream as an abstract adjacency graph. These discovered speech pattern clusters are shown to

correspond to words and phrases that are relevant to the audio streams from which they are extracted.

(4) We propose and evaluate two methods for cluster identification using lightweight speech recognition methods. Both methods take advantage of the constraint that all cluster members share a consistent lexical identity.

(5) In a related, but slightly different direction from the rest of this work, we show how alignment distortions can be used to compare utterances for the purpose of speaker segmentation. This alignment-based comparison strategy is shown to perform well at finding major speaker change points in multi-speaker lectures.

1.5 Outline

The remainder of this document is organized as follows: Chapter 2 provides the background for many of the experiments conducted in this work, including description of the speech corpora used, and some of the preliminary signal processing methods used. Chapter 3 describes the segmental dynamic time warping (DTW) algorithm, an adaptation of a widely known dynamic programming technique, which is designed to find matching acoustic patterns between spoken utterances. In Chapter 4, we present experiments to evaluate the accuracy of the matching alignment paths that are produced by the segmental DTW algorithm, both at the phonetic level, and the word level. In Chapter 5, we employ clustering techniques to discover patterns that correspond to words and phrases in speech by aggregating the alignment paths that are produced by the segmental DTW algorithm. Methods for automatically identifying these discovered patterns are described and evaluated in Chapter 6. In Chapter 7, we apply the segmental DTW technique to the problem of speaker segmentation and evaluate segmentation performance on several multi-speaker lectures. Finally, in Chapter 8, we summarize the main points of this thesis, describe our contributions and suggest directions for future work.

Chapter 2
Experimental Background

This chapter provides background information for experiments presented in later chapters. Information about the speech corpora used in the experiments, as well as processing methods for preparing the data are described.

2.1 Speech Corpora

2.1.1 The Ice Cream Corpus

In this dissertation, experiments involving phone-level transcriptions of speech data are carried out using the Ice Cream corpus (ICC), a set of phonetically compact sentences read by a single speaker [33]. The Ice Cream corpus consists of 720 unique sentences, each spoken once. Each utterance is between 1.3 and 3.2 seconds long, and contains between 7 and 14 words. Two example sentences from the corpus are shown in Figure 2-1. Time-aligned orthographic and phonetic transcriptions for each utterance are provided by human transcribers.

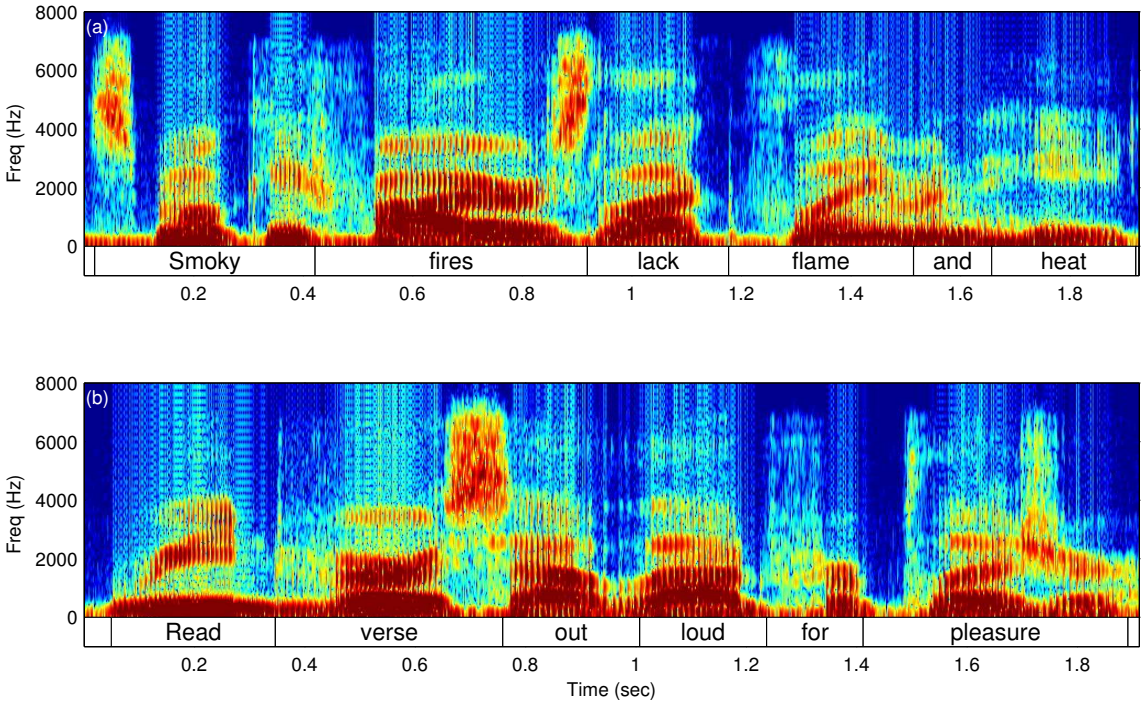


Figure 2-1: Two example ICC sentences with time aligned orthographies.

2.1.2 MIT Lecture Corpus

Word-level experiments in this thesis which are presented in subsequent chapters, as well as illustrative examples used throughout this thesis are taken from an extensive corpus of academic lectures recorded at MIT. At present, the lecture corpus includes more than 300 hours of audio data recorded from 8 different courses and over 80 seminars given on a variety of topics such as poetry, globalization, psychology, and science. The amount of data in the corpus continues to grow, as new seminars and courses are continually being recorded and added. Many of the seminar lectures are, at the time of this writing, publicly available on the MIT World website [74]. Likewise, the audio and video of many of the course lectures are also available online as part of the MIT Open Courseware initiative [73]. In most cases, each lecture takes place in a classroom environment, and the audio is recorded with an omni-directional microphone (as part of a video recording). The lectures used for experiments or examples in this thesis are described in Table 2.1. In this section, we give an overview of some characteristics of the lecture data.

Lecture Characteristics

One of the unique characteristics of the lecture corpus is the quantity of speech data that is available for any particular speaker. Unlike other sources of speech data, the academic lectures and a large portion of the seminars are primarily comprised of a single speaker addressing an audience for up to an hour or more at a time. Though some lectures adopt a panel format, where several speakers take turns acting as the main speaker, the amount of data per speaker remains significant - between five minutes to half an hour. In many of the seminar-style lectures, the main talk is followed by a question and answer (Q & A) session with the audience asking questions of the speaker or the panel.

Word usage and speaking style are two other factors that distinguish the lecture data in this corpus from other commonly used speech corpora such as Switchboard, a corpus of telephone conversations [43], or Broadcast News data [36, 83]. For the most part, course lectures tend to have relatively small vocabularies which make frequent use of subject specific words that may not be commonly used in everyday speech. An analysis of 80 lectures taken from three undergraduate courses in math, physics, and computer science revealed that each one hour lecture contained between 5K and 12K total words, with an average of approximately 7K words. However, the number of unique words used per lecture ranged from 500 to 1,100 words, with an average of 800 words [42]. A graph of the out-of-vocabulary (OOV) rates for these courses as a function of training vocabulary size is shown in Figure 2-2. The graph reveals that even with a vocabulary as large as 10K words taken from the most frequently occuring words in Switchboard, the OOV rate remains approximately 10% for each of

Title	Speaker	Duration
1. The World is Flat	**Thomas Friedman**	**1 hr 15 mins**
New York Times columnist and author Thomas Friedman discusses his latest book, *The World is Flat*.		
2. A Beautiful Mind	**Silvia Nasar**	**1 hr 15 mins**
Journalism professor and author Silvia Nasar discusses her biography of the mathmetician John Nash.		
3. ASR Lecture 2	**Victor Zue**	**1 hr 25 mins**
The second lecture from a course on automatic speech recognition. The topic of this lecture is the acoustics of speech production.		
4. ASR Lecture 6	**James Glass**	**1 hr 18 mins**
The sixth lecture from a course on automatic speech recognition. The topics of this lecture are distortion measures and vector clustering algorithms.		
5. ASR Lecture 19	**Timothy Hazen**	**1 hr 14 mins**
The nineteenth lecture from a course on automatic speech recognition. The topic of this lecture is speaker adaptation.		
6. Physics II Lecture 3	**Walter Lewin**	**51 mins**
The third lecture from a course on electricity and magnetism. The topic of this lecture is the electric fields and electric flux.		
7. Linear Algebra Lecture 2	**Gilbert Strang**	**47 mins**
The second lecture from a course on linear algebra. The topic of this lecture is matrix solution by elimination techniques.		

Table 2.1: Description of lectures used for experiments in this thesis.

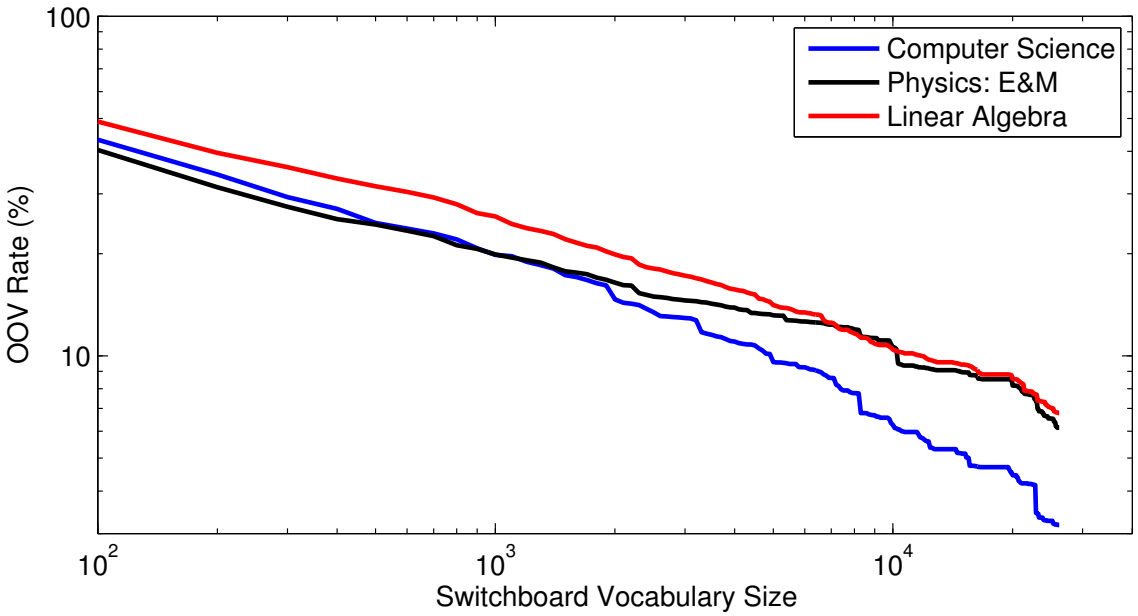


Figure 2-2: OOV rate versus training vocabulary size for three academic courses. The training vocabulary source is a frequency ranked lexicon taken from the Switchboard corpus.

the different courses. This analysis indicates that the lexicon for any particular lecture may not be wide ranging, but has high usage of a relatively small set of subject specific words.

In terms of language usage, the speaking style observed in the lecture data is less formal than that observed in Broadcast News or read speech such as TIMIT [124], but is less spontaneous than in true conversational speech sources such as Switchboard. An example of the type of speech found in one of the MIT World seminars is shown in Table 2.2. The transcript deviates significantly from patterns typically observed in formal written text, exhibiting artifacts such as filled pauses (1,2,3), false starts (3), sentence fragments (4), and sentence planning errors (5).

Segmentation

Unlike the data in the Ice Cream corpus, lectures are typically recorded as a single stream of audio often over 1 hour in length, with no supplementary indicators of where one utterance stops and another begins. For many of the processing steps undertaken in later chapters, we require a set of discrete utterances in order to compare utterances to one another. In order to subdivide the audio stream into discrete segments of continuous speech, we use a basic phone recognizer to

(1)	So let me simply conclude with [uh] an insight that was imparted me by Carly Fiorina from H P before she [um] lost her job.
(2)	[uh] Carly got all this actually.
(3)	[um] I – you know – I don't know about the business side but she was very smart [um] about all of this and she said to me,
(4)	"You know Tom, everything we called the IT revolution? The information technology revolution, these last twenty years? Sorry to tell you, that was just the warm up act."
(5)	That has just been the sharpening, forging – forging, sharpening, and distribution of the tools of collaboration into this new platform

Table 2.2: A segment of speech taken from a lecture, "The World is Flat", delivered by Thomas Friedman.

identify regions of silence in the signal. Silent regions with duration longer than 2 seconds are removed and the portions of speech in between those silences are used as the isolated utterances. The use of a phone recognizer is not a critical prerequisite for this segmentation procedure, since we only use the output to make a speech activity decision at each particular point in time. In the absence of a phone recognizer, a less sophisticated technique for speech activity detection can be substituted in its place. Segmentation statistics for the lectures described in Table 2.1 are shown in Table 2.3. Most of the utterances produced during the segmentation procedure are short, averaging durations of less than 3 seconds. The segmentation procedure is also conservative enough that segmentation end points are rarely placed in the middle of a word.

Lecture	Length	# Segments	Avg. Duration (s)	Max. Duration (s)
1	1 hr 15 mins	2089	1.66	6.76
2	1 hr 15 mins	1725	2.17	6.00
3	1 hr 25 mins	2798	1.34	6.61
4	1 hr 18 mins	2042	1.78	7.26
5	1 hr 14 mins	1905	1.70	7.40
6	51 mins	1510	1.33	5.08
7	47 mins	1260	1.54	7.06

Table 2.3: Segmentation statistics for the lectures described in Table 2.1.

2.2 Signal Processing

In subsequent chapters, we will treat spoken utterances as time series of spectral vectors. Here, we give a brief overview of the signal processing steps used to generate the spectral representation of speech that we will use throughout this

thesis.

Although there are a number of spectral representations that are widely used in the speech research community [92], we use whitened Mel-scale cepstral coefficients (MFCCs). Our choice of representation is motivated primarily by our need for a distance measure which is able to quantify the distortion between two spectral feature vectors. The process of whitening decorrelates the dimensions of the feature vector and normalizes the variance in each dimension. These characteristics of this spectral representation make the standard unweighted Euclidean distance metric a reasonable choice for comparing two feature vectors, as the distance in each dimension will also be uncorrelated and have equal variance.

When captured by a microphone, a speech signal is stored as a digitized waveform, $x[t]$, which is essentially a one dimensional time series of data samples collected at a fixed sampling rate. In most of the data considered in this thesis, the default sampling rate was 16 kHz. The process of converting $x[t]$ into whitened MFCC vectors, $\mathcal{X} = \mathbf{x}_1, \ldots, \mathbf{x}_{N_x}$, can be summarized in the following steps:

(1) Subtract the mean of the waveform and normalize the magnitude of the waveform,

$$x_1[t] = \frac{x[t] - \bar{x}[t]}{\max |x[t]|}. \tag{2.1}$$

(2) Pre-emphasize the normalized waveform,

$$x_2[t] = x_1[t] - 0.97 x_1[t-1]. \tag{2.2}$$

(3) Calculate the short-time Fourier transform with a frame interval of 10 ms, a 25.6 ms Hamming window, and a 256 point discrete Fourier transform.

$$X_{stft}(n,k) = \sum_{m=-\infty}^{\infty} x_2[n] w[n-m] e^{j\omega_k m}, \qquad \omega_k = \frac{2\pi}{256} k, \tag{2.3}$$

where $X_{stft}(n,k)$ is the value of the k-th spectral component at time n and w is the Hamming window.

(4) Calculate the Mel-frequency spectral coefficients (MFSCs) from the STFT representation by weighting the spectral energy from the STFT by the mel-scale filters shown in Figure 2-3.

$$X_{mfsc}(n,l) = \frac{1}{A_l} \sum_{k=-\infty}^{\infty} \left| V_l(k) X_{stft}(n,k) \right|^2 \tag{2.4}$$

where A_l is the energy under the l-th filter, $V_l(k)$.

(5) Compute the logarithm of the MFSCs.

$$X_{lmfsc}(n,l) = 10\log_{10}(X_{lmfsc}(n,l)). \tag{2.5}$$

(6) Take the discrete cosine transform of the log MFSCs to get the corresponding Mel-frequency cepstral coefficients (MFCCs) [27].

$$X_{mfcc}(n,m) = \frac{1}{R}\sum_{k=0}^{R-1} X_{lmfsc}(n,l)\cos\left(\frac{2\pi}{R}km\right), \tag{2.6}$$

where R is the number of mel filters.

(7) Finally, "whiten" the MFCC vectors to produce d-dimensional vectors which have uncorrelated dimensions and unity variance along each dimension [10]. The first step in this whitening process is to compute the correlation matrix, $\mathcal{C}$, using all of the MFCC vectors to be processed.

$$\mathcal{C}(i,j) = \frac{\sigma_{ij}}{\sigma_i\sigma_j}. \tag{2.7}$$

where σ_{ij} is the estimated covariance of dimensions i and j, and σ_i is the standard deviation of the i-th dimension. Next, the eigenvalues, λ_i, and eigenvectors, $\mathbf{v}_i$ of $\mathcal{C}$ are computed and used to generate the whitening rotation matrix, $\mathcal{R}$, where each entry is given by

$$\mathcal{R}(i,j) = \frac{\mathbf{v}_i(j)}{\sqrt{\lambda_i}}. \tag{2.8}$$

The resulting output vectors that we use as our final representation are given by

$$\mathcal{X} = \mathbf{x}_1, \ldots, \mathbf{x}_{N_x} \tag{2.9}$$

where $\mathbf{x}_k$ is just a rotated version of the k-th MFCC vector,

$$\mathbf{x}_k = \mathcal{R}X_{mfcc}(n) \tag{2.10}$$

2.3 The SUMMIT Speech Recognizer

In parts of this thesis, we make use of the output of an automatic speech recognizer, either for phone recognition in the segmentation component, or during cluster identification. The speech recognition system used in these situations is the SUMMIT speech recognizer. In this section, we give a broad overview of the SUMMIT recognition framework. Since this thesis does not rely upon modifying the functionality or architecture of SUMMIT, we defer a more detailed description of the probabalistic framework to [41].

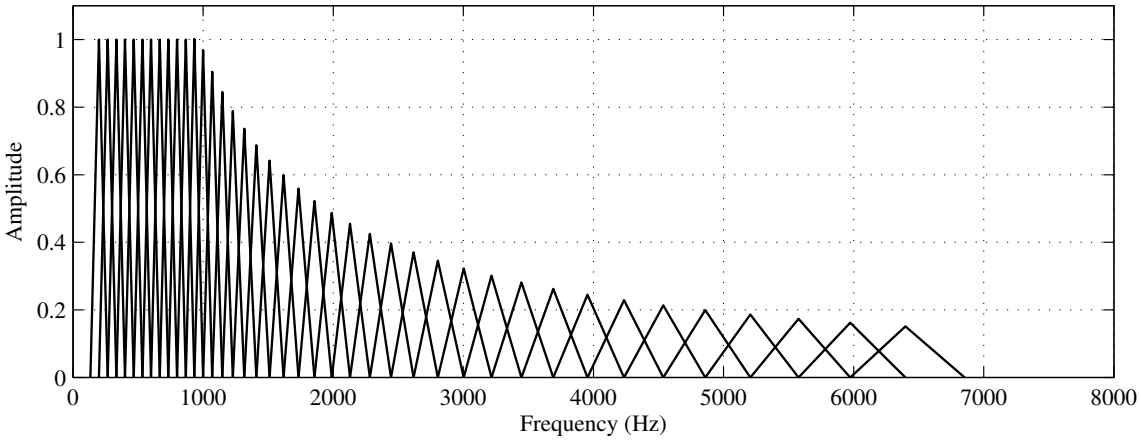


Figure 2-3: The 40 triangular mel-scale filters used to weight the STFT vectors, $X_{stft}(n)$. The filter spacings and widths implement a mel-warping of the frequency axis, where low frequency filters have constant bandwidth and linear spacing of center frequencies, while high frequency filters have wider bandwidths and logarithmically spaced center frequencies. These features are characteristic of auditory critical bands, which exhibit better frequency resolution at lower frequencies.

The SUMMIT speech recognizer is a landmark-based automatic speech recognition system that incorporates multiple levels of knowledge of the speech signal in the process of generating a word hypothesis for a given utterance.

At the acoustic level, diagonal covariance Gaussian mixture models are used to model the acoustic feature vectors that are extracted from the speech signal. One way that SUMMIT differs from other state-of-the-art recognizers is in its use of landmark-based features. Rather than extracting feature vectors at a constant frame rate (e.g., every 10 milliseconds), a heuristic landmark detector preprocesses the signal and hypothesizes points of interest in the speech signal from which features are extracted. These landmarks usually correspond to regions of significant acoustic change either between phonetic units (transitional) or within a phonetic unit (internal). The acoustic models used are context-dependent diphones that are trained on labeled training data [112].

At the phonological level, a set of manually and automatically determined phonological rules are used to specify what types of surface pronunciations can result from a given sequence of phonemes [52, 50, 51]. Lexical knowledge is incorporated via a dictionary of pronunciations which list a word and the set of acceptable pronunciations in terms of phonemes. Finally, linguistic knowledge is provided in the form of a language model that uses textual data to assign probabilities to various word sequences.

The multiple knowledge sources described above are combined using a cascade of

finite state transducers (FSTs) [89, 107]: C, the mapping of context-dependent phones to the acoustic model diphones, P, the phonological rules, L, the lexicon, and G, the grammar. The final combined FST is the composition of these four individual transducers: $C \circ P \circ L \circ G$. During recognition, a search is performed on the combined FST to produce the hypothesized word sequence.

For experiments in this thesis requiring the use of the SUMMIT speech recognizer, the acoustic model training data is taken from a set of lectures with speakers disjoint from Lectures 1, 4, 5, and 6 in Table 2.1. Vocabulary selection and language model training data for the grammar vary according to the usage of the recognizer and are described prior to each specific experiment.

Chapter 3

Segmental DTW: Algorithm

This chapter motivates and describes a dynamic programming algorithm which we call segmental dynamic time warping (DTW) [84, 86]. Segmental DTW takes as input two continuous speech utterances and finds matching pairs of subsequences. This algorithm serves as the fundamental building block for the pattern discovery methodology described in subsequent chapters.

3.1 Motivation and Background

The primary motivation for the acoustic pattern matching technique discussed in this chapter can be better understood by considering the example shown in Figure 3-1. The utterances shown in this example, and throughout this chapter are from the lecture given to the MIT Mathematics department by Silvia Nasar, author of the John Nash biography, "A Beautiful Mind". Upon visual inspection, one can observe that the first two utterances, (a) and (b), are qualitatively similar to each other, and that both are qualitatively different from the third utterance, (c). More specifically, utterances (a) and (b) consist of roughly the same set of spectral events occurring in the same order. On the other hand,

utterance (c) contains some similar spectral events to those found in (a) and (b), but the order of these events is significantly different. One of our goals in this work is to demonstrate that knowledge of the similarity of (a) to (b), and in turn, their difference from (c), is of value for speech processing, even when the underlying lexical *identities* of each utterance is unavailable. The ability to use such knowledge to structure observed speech data is a crucial first step for being able to perform unsupervised learning from unlabeled data and classification of subsequent input speech. In the case of speech, evaluating the similarity of one sequence to another is a difficult problem. For sequences where the elements consist of distinct symbols, such as strings, confusion matrices or exact match can be used. When comparing sounds directly, however, the concept of distance is less clear.

How then, can we compare the acoustic observation sequences computed from a set of speech segments against one other? A naïve solution would be to represent each utterance as a fixed-length feature vector and to compute similarities using a distance metric between feature vectors. Although this approach might work if words were uttered with fixed duration and timing, in practice, significant temporal variations occur in each spoken instance of a word. Another solution would be to transform the sequences into an intermediate representation that is more amenable to comparison, such as a sequence of strings, but this would be sensitive to the transformation function and would also require training data to discover the mapping of acoustics into strings. A more direct technique for comparing variable length sequences is to consider the distortion between the sequences when they have been appropriately aligned. The technique we turn to for this purpose is dynamic time warping (DTW), which has been widely studied in the speech recognition community [101, 102, 77, 94] and other fields such as time series data mining [59] and handwriting recognition [76]. We note here that the use of DTW is well-suited to the lecture data processed in this work, since many variables are controlled for, such as speaker, microphone, and acoustic environment.

3.2 Dynamic Time Warping

In 1971, Sakoe and Chiba first illustrated the use of dynamic time warping for speech recognition [101]. Until the widespread adoption of Hidden Markov Models by the speech recognition community some years later [55, 7, 63, 93], DTW was widely used in a variety of speech processing applications. In this section, we briefly review the technique and discuss how we make use of it in our work.

As originally proposed, DTW was intended for aligning examples of isolated words to reference templates. For two utterances, $\mathcal{X}$ and $\mathcal{Y}$, the optimal alignment path between the two, $\hat{\phi}$, is computed, and the distortion between the two

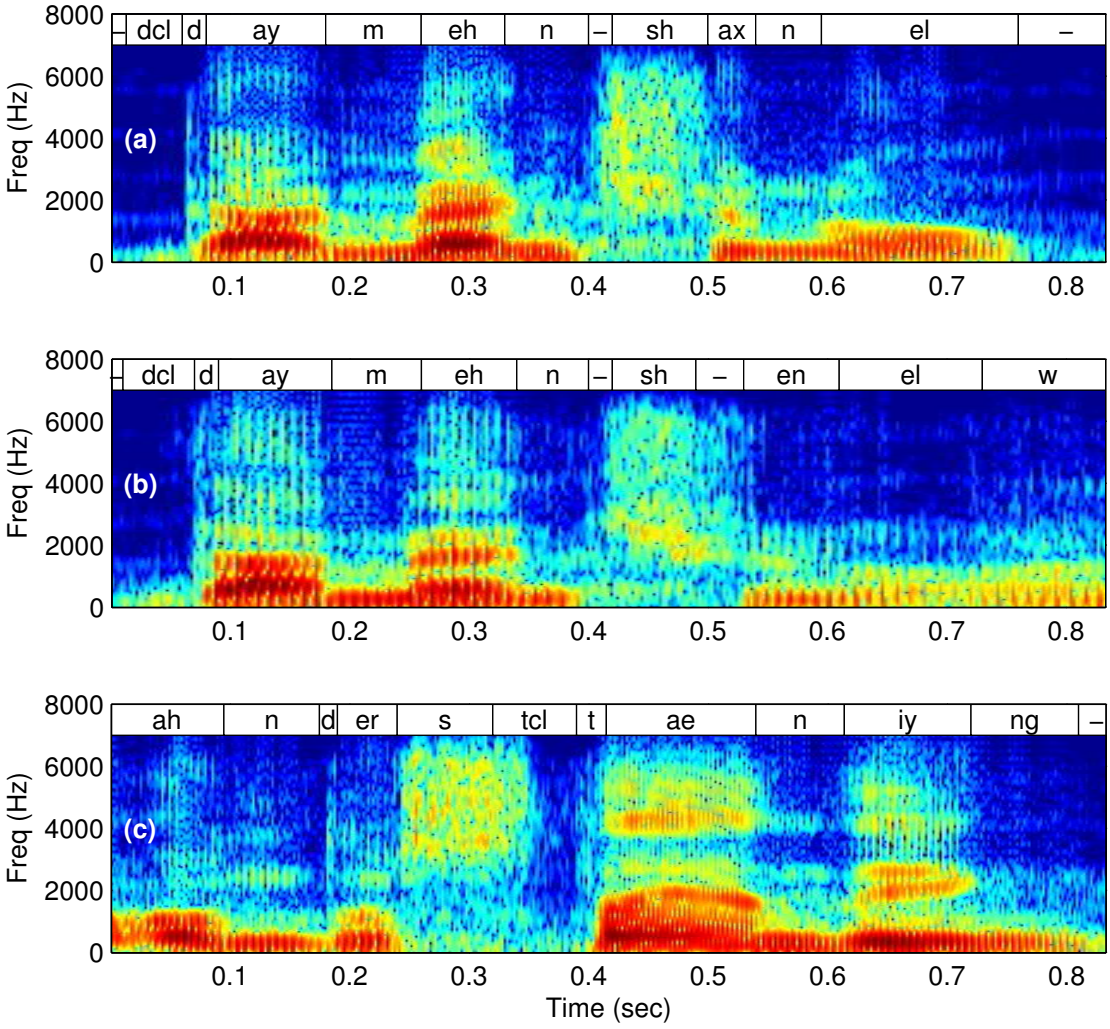


Figure 3-1: Spectrograms for three isolated word utterances spoken in a conversational context. The first two utterances are of the word "dimensional". The third utterance is of the word "understanding".

utterances along that path, $d_{\hat{\phi}}(\mathcal{X}, \mathcal{Y})$ is used to compare the two. We describe this procedure more fully in the next section.

3.2.1 Formal Description of DTW

Given the frame level spectral representation of two utterances

$$\mathcal{X} = \mathbf{x}_1, \ldots, \mathbf{x}_{N_x} \tag{3.1}$$

and

$$\mathcal{Y} = \mathbf{y}_1, \ldots, \mathbf{y}_{N_y}, \tag{3.2}$$

we define a warping relation, or warp path, ϕ, to be an alignment which maps $\mathcal{X}$ to $\mathcal{Y}$ while obeying several constraints. The warping relation can be written as a sequence of ordered pairs,

$$\phi = \big(i_k, j_k\big) \quad k = 1, \ldots, T, \tag{3.3}$$

that represents the mapping,

$$\begin{aligned} \mathbf{x}_{i_1} &\leftrightarrow \mathbf{y}_{j_1} \\ \mathbf{x}_{i_2} &\leftrightarrow \mathbf{y}_{j_2} \\ &\vdots \\ \mathbf{x}_{i_T} &\leftrightarrow \mathbf{y}_{j_T} \end{aligned}$$

In the case of global alignment, ϕ maps all of sequence $\mathcal{X}$ to all of sequence $\mathcal{Y}$. This implies endpoints at the beginning and end of each utterance, i.e. $(i_1, j_1) = (1, 1)$ and $(i_T, j_T) = (N_X, N_Y)$. An example of such a warp path is shown in Figure 3-2. We note that in order for such a mapping to be a valid alignment, some local constraints must be enforced. First, the aligned sequences must retain their original ordering and progress forward in time,

$$i_k \leq i_{k+1}, \quad j_k \leq j_{k+1}. \tag{3.4}$$

Second, for our particular implementation, we choose to ensure that no frames are skipped along the warp path,

$$i_{k+1} \leq i_k + 1, \quad j_{k+1} \leq j_k + 1. \tag{3.5}$$

Conditions 3.4 and 3.5 are referred to as the Monotonicity and Continuity conditions, respecively. Both constraints are adhered to for the warp path in Figure 3-2, as each step progresses monotonically towards the bottom right corner, and no steps are skipped along the way.

Given a valid warp path, we now have a way to compare two utterances of

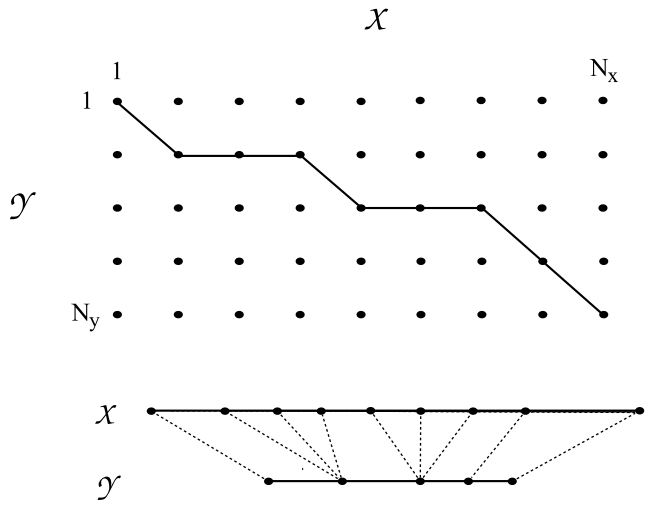


Figure 3-2: An example warp path aligning sequences $\mathcal{X}$ and $\mathcal{Y}$ of lengths N_x and N_y, respectively. The warp path ϕ in this case is the sequence of ordered pairs: (1,1) (2,2) (3,2) (4,2) (5,3) (6,3) (7,3) (8,4) (9,5). The alignment corresponding to the warp path is displayed in the lower part of the figure.

different lengths by using the accumulated distortion between aligned frames,

$$D_\phi(\mathcal{X}, \mathcal{Y}) = \sum_{k=1}^{T} d(\mathbf{x}_{i_k}, \mathbf{y}_{j_k}). \tag{3.6}$$

We denote the optimal warping relation, $\hat{\phi}$, as the one that minimizes the accumulated distortion while simultaneously satisfying Conditions 3.4 and 3.5,

$$\hat{\phi} = \arg\min_{\phi} D_\phi(\mathcal{X}, \mathcal{Y}) \tag{3.7}$$

Initially, the problem of finding the optimal path may seem a difficult one, considering the large search space of possible warping relations. Fortunately, the optimal path problem exhibits optimal substructure, thus lending itself well to a dynamic programming solution. This can be seen by working backwards: consider the optimal global warp path, $\hat{\phi}$ aligning $\mathcal{X}$ to $\mathcal{Y}$. The last coordinate in this path must be $\mathcal{Q} = (N_x, N_y)$. Applying the continuity condition, the preceding coordinate must then be one of

$$\begin{aligned} \mathcal{P}_1 &= (N_x - 1, N_y), \\ \mathcal{P}_2 &= (N_x, N_y - 1), \\ \mathcal{P}_3 &= (N_x - 1, N_y - 1), . \end{aligned}$$

Therefore, the optimal path ending at $\mathcal{Q}$ will be an extension of one of the

optimal paths ending at $\mathcal{P}_1$, $\mathcal{P}_2$, or $\mathcal{P}_3$. Define the accumulated distortion of the optimal path from $(1,1)$ to the coordinate, $\mathcal{P}$, to be $\mathcal{D}(\mathcal{P})$,

$$\mathcal{D}(\mathcal{P}) \triangleq \min_{\phi} \sum_{k=1}^{T'} d(\mathbf{x}_{i_k}, \mathbf{y}_{j_k}), \tag{3.8}$$

where

$$\mathcal{P} = (i_{T'}, j_{T'}). \tag{3.9}$$

Then the optimal global distortion can be found from the optimal distortions ending at the preceding points,

$$D_{\hat{\phi}}(\mathcal{X}, \mathcal{Y}) = \mathcal{D}(\mathcal{Q}) = \min\big[\mathcal{D}(\mathcal{P}_1), \mathcal{D}(cP_2), \mathcal{D}(\mathcal{P}_3)\big] + d(\mathbf{x}_{N_x}, \mathbf{y}_{N_y}). \tag{3.10}$$

This procedure can then be carried out recursively for $\mathcal{D}(\mathcal{P}_1)$, $\mathcal{D}(\mathcal{P}_2)$, and $\mathcal{D}(\mathcal{P}_3)$, down to the starting coordinate $(1,1)$, whose distortion we define to be

$$\mathcal{D}(1,1) = d(\mathbf{x}_1, \mathbf{y}_1). \tag{3.11}$$

Once the optimal distortion has been computed, the optimal path, $\hat{\phi}$ can be found by backtracking through the sequence of transitions made during the distortion computation.

3.3 Segmental DTW

Dynamic time warping, as described in the previous section, is most suitable for finding the optimal global alignment and the associated distortion between two whole word exemplars. This is a consequence of path constraints which fix the starting and ending points of the alignment path to be $(1,1)$ and (N_x, N_y), or some slight variation thereof, respectively. When the utterances that we are trying to compare happen to be isolated words, this approach is a suitable way to directly measure the similarity of two utterances at the acoustic level. However, if the utterances consist of multiple words sequences, the distances and paths produced by optimal global alignment may not be meaningful. Although DTW was applied, with some success, to the problem of connected word recognition via a framework called level building, this technique still required the existence of a set of isolated word reference templates [77]. In that respect, the problem has significant differences to the one in which we are interested. Consider the pair of utterances shown in Figure 3-3:

(1) "He too was diagnosed with paranoid schizophrenia"

(2) "... were willing to put Nash's schizophrenia on record"

Even in an optimal scenario, a global alignment between these two utterances

would be forced to map speech frames from dissimilar words to one another, making the overall distortion difficult to interpret. This difficulty arises primarily because each utterance is composed of a different sequence of words, meaning that the utterances can not be considered from a global perspective. However, (1) and (2) do share similarities at the local level. Namely, both utterances contain the word "schizophrenia". Identifying and aligning such similar local segments is the problem we seek to address in this section. Our proposed solution is a *segmental* variant of DTW that attempts to find subsequences of two utterances that align well to each other. Segmental DTW is comprised of two main components: a local alignment procedure which produces multiple warp paths that have limited temporal variation, and a path trimmimg procedure which retains only the lower distortion regions of an alignment path.

3.3.1 Local Alignment

In this section we modify the basic DTW algorithm in several important ways. First, we incorporate global constraints to restrict the allowable shapes that a warp path can take. Second, we attempt to generate multiple alignment paths for the same two input sequences by employing different starting and ending points in the DTW search.

The need for global constraints in the DTW process can be seen by considering the example in Figure 3-4. The shape of the path in the figure corresponds to an alignment that indicates that $\mathcal{X}$ is not a temporally dilated form of $\mathcal{Y}$, or vice versa. A more rigid alignment would prevent an overly large temporal skew between the two sequences, by keeping frames from one utterance from getting too far ahead of frames from the other. The following criterion, proposed by Sakoe and Chiba, accomplishes this goal. For a warp path, originating at (i_1, j_1), the k-th coordinate of the path, $\mathcal{P}_k = (i_k, j_k)$, must satisfy,

$$\left|\left(i_k - i_1\right) - \left(j_k - j_1\right)\right| \leq R. \tag{3.12}$$

The constraint in 3.12 essentially limits the path to a diagonal region of width $2R + 1$. This region is shown in Figure 3-4, for a value of $R = 2$. Depending on the size of R, the ending point of the constrained path may not reach (N_x, N_y). As we will note later, an alignment path resulting in unassigned frames in either of the input utterances may be desireable in cases where only part of the utterances match.

In addition to limiting temporal skew, the constraint in Eq. 3.12 also introduces a natural division of the search grid into regions suitable for generating multiple alignment paths with offset start coordinates as shown in Figure 3-5. For utterances of length N_x and N_y, with a constraint parameter of R, the start

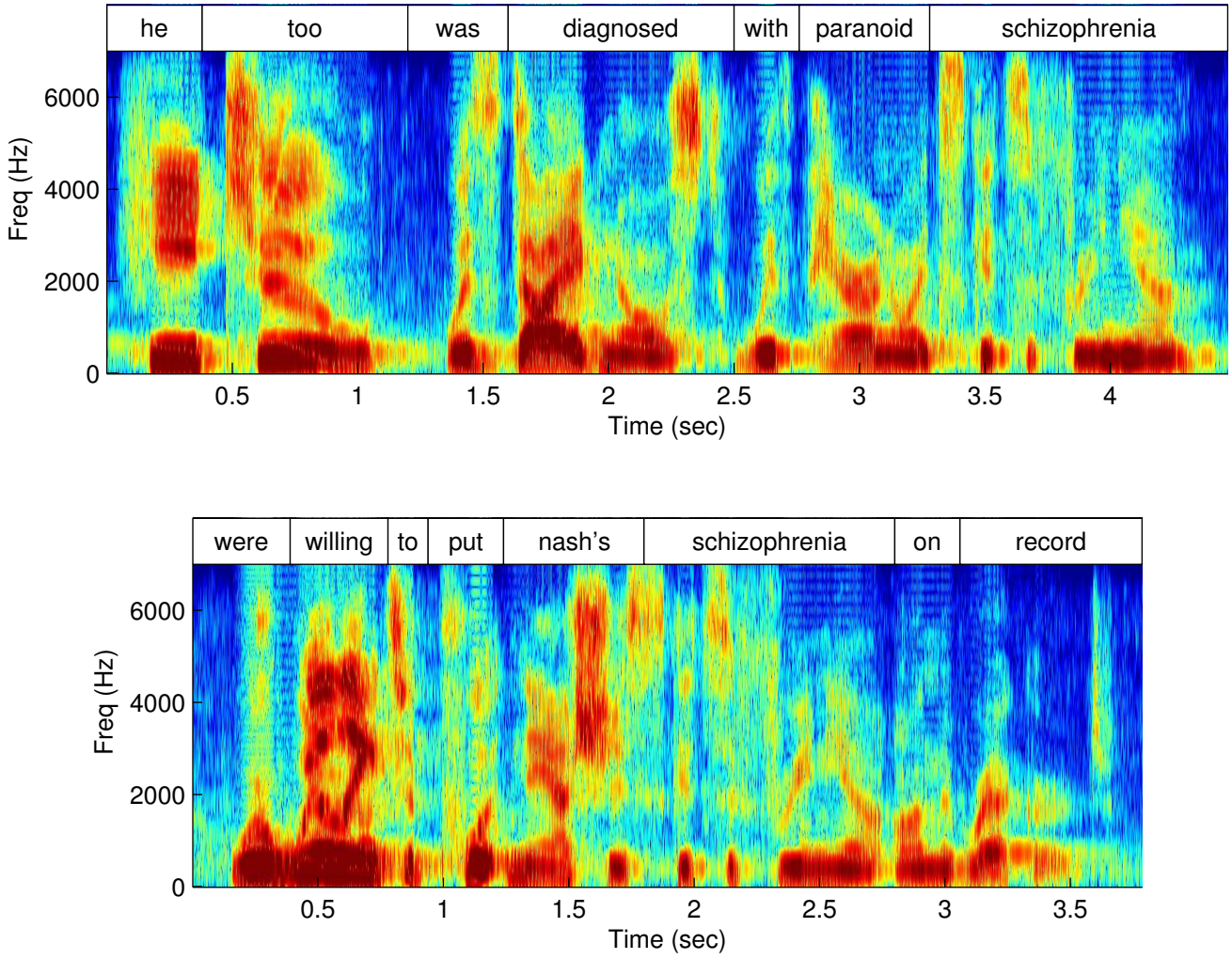


Figure 3-3: Spectrograms for two utterances spoken by a female speaker with their time-aligned orthographies. The upper utterance is the phrase, "he too was diagnosed with paranoid schizophrenia". The lower utterance is the phrase, "were willing to put Nash's schizophrenia on the record".

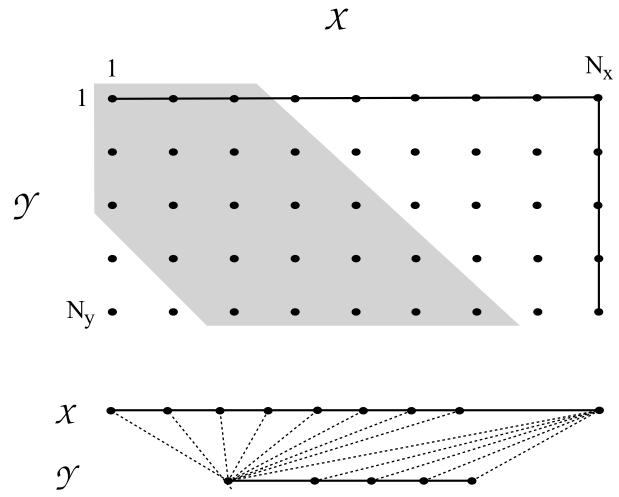


Figure 3-4: A non-ideal warp path that can result from unconstrained alignment. For this path, all frames from $\mathcal{X}$ are mapped to the first frame of $\mathcal{Y}$, and all frames from $\mathcal{Y}$ are mapped to the last frame of $\mathcal{X}$. The alignment corresponding to the warp path is displayed in the lower part of the figure. The shaded region of the graph represents the allowable set of path coordinates following the band constraint in Eq. 3.12 with $R = 2$.

coordinates will be

$$\big((2R+1)k+1, 1\big), \qquad 0 \le k \le \left\lfloor \frac{N_x - 1}{2R+1} \right\rfloor$$
$$\big(1, (2R+1)k+1\big), \qquad 1 \le k \le \left\lfloor \frac{N_y - 1}{2R+1} \right\rfloor .$$

Based on these coordinates, the total number of diagonal regions will be

$$N_R = \left\lfloor \frac{N_x - 1}{2R+1} \right\rfloor + \left\lfloor \frac{N_y - 1}{2R+1} \right\rfloor + 1. \tag{3.13}$$

Therefore, we have a set of N_R diagonal regions, each defining a range of alignments between the two utterances with different offsets but the same temporal rigidity. Within each region, we can use dynamic time warping to find the optimal local alignment, $\hat{\phi}_r$, where r is the index of the diagonal region.

3.3.2 Path Refinement

At this stage, we are left with a family of local warp paths, $\hat{\phi}_r$, for $r = 1, \ldots, N_R$. Because we are only interested in finding portions of the alignment which are similar to each other, the next step is to refine the warp path by discarding parts

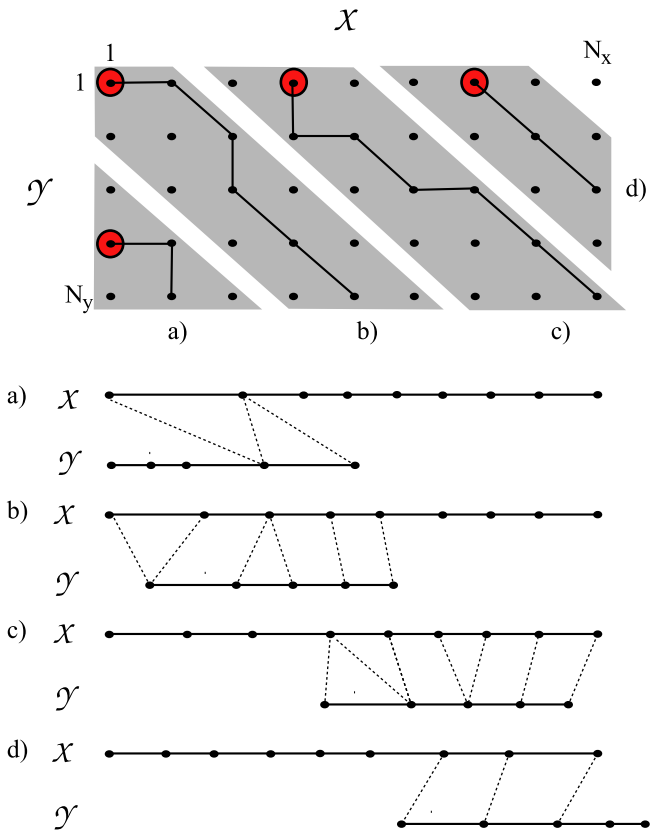


Figure 3-5: Multiple alignment paths resulting from applying the band constraint with R = 1. Starting coordinates for each region are shown in red. The alignments corresponding to each diagonal region are shown below the grid.

of the alignment with high distortion. We accomplish this by first identifying the length-constrained minimum average (LCMA) distortion *fragment* of the local alignment path. We then *extend* the path fragment to include neighbouring points falling below a particular threshold.

The problem of finding the LCMA distortion fragment can be described more generally as follows. Consider a sequence of positive real numbers

$$S = \langle s_1, ..., s_N \rangle \tag{3.14}$$

and a length constraint parameter, L. Then the length constrained minimum average subsequence, LCMA(S, L) is a consecutive subsequence of S with length at least L that minimizes the average of the numbers in the subsequence. In our work, we make use of an algorithm proposed by Lin *et al.* for finding LCMA(S, L) in $O(N \log(L))$ time [67].

Recall that every warp path ϕ is a sequence of ordered pairs,

$$\phi = (i_1, j_1), \ldots, (i_T, j_T). \tag{3.15}$$

Associated with each warp path, is a distortion sequence whose values are real and positive,

$$\delta(\phi) = d(\mathbf{x}_{i_1}, \mathbf{y}_{j_1}), \ldots, d(\mathbf{x}_{i_T}, \mathbf{y}_{j_T}). \tag{3.16}$$

The minimum distortion warp path fragment, φ, is a subsequence of ϕ that satisfies

$$\delta(\varphi) = \text{LCMA}(\delta(\phi), L). \tag{3.17}$$

The minimum length criterion plays a practical role in computing the minimum average subsequence. Without the length constraint, the minimum average subsequence would typically be just the smallest single element in the original sequence. Likewise, for our application, it has the effect of preventing spurious matches between short segments within each utterance. The length criterion also has important conceptual implications. The value of L essentially serves to control the granularity of repeating patterns that are returned by the segmental DTW procedure. Small values of L will lead to many short, subword patterns being found, while large values of L will return fewer, but more linguistically significant patterns such as words or phrases. In the remainder of this chapter, we show example outputs that are produced when segmental DTW is applied to pairs of utterances.

3.4 Example Outputs

In this section, we step through the segmental DTW procedure for an example pair of utterances. We begin by revisiting the example presented at the beginning of Section 3.3. The distance matrix for the two utterances from Figure 3-3 is

displayed in Figure 3-6. In this distance matrix, each cell corresponds to the Euclidean distance between frames from each of the utterances being compared. The cell at row i, column j, corresponds to the distance between frame i of the first utterance and frame j of the second utterance. The local similarity between the utterance portions containing the word "schizophrenia" are evident in the diagonal band of low distortion cells stretching from the time coordinates $(1.6, 0.9)$ to $(2.1, 1.4)$. From the distance matrix, a family of constrained warp paths is found using dynamic time warping as shown in Figure 3-7. The width parameter which constrains the extent of time warping is set to $R = 10$ frames, at a 5 millisecond analysis rate, which corresponds to a total allowable offset of 105 milliseconds. The warp paths are overlaid with their associated length constrained minimum average path fragments. The length parameter used in this example is $L = 100$, which corresponds to approximately 500 ms. The coloring of the warp path fragments correspond to the average distortion of the path fragment, with bright red fragments indicating low distortion paths and darker shades indicating high distortion paths. Typically, there is a wide range of distortion values for the path fragments found, but only the lowest distortion fragments are of interest to us, as they indicate potential local matches between utterances.

A three-dimensional view of the lowest distortion path is shown in Figure 3-8, with the associated path fragment highlighted in red. This figure illustrates how the path distortion varies through time along a particular alignment. An alternate view of the distortion path, including a frame-level view of the individual utterances, is shown in Figure 3-9. This view of the distortion path highlights the need for extending the path fragments discovered using the LCMA algorithm. Although the distortion remains low from the onset of the word "schizophrenia" in each utterance, the LCMA path fragment (shown in red) starts almost 500 ms after this initial drop in distortion. In order to compensate for this phenomenon, we allow for path extension using a distortion threshold based on the values in the path fragment, for example within 10% of the distortion of the original fragment. The extension of the fragment is shown in Figure 3-9 as a white line.

Although the endpoints of the extended path fragment in Figure 3-9 happen to coincide with the common word boundaries for that particular example, in many cases, the segmental DTW algorithm will align subword sequences or even multi-word sequences. This is because, aside from fragment length, the segmental DTW algorithm makes no use of lexical identity when searching for an alignment path.

An example of a subword match can be seen by observing the distance matrix for the phrases "generation", and "all that recognition", as shown in Figure 3-10. Although the utterances contain no common words, the distance matrix reveals a low distortion band matching the last part of each utterance. In this case, the resulting path fragment will match the suffix "tion" and the preceding vowel from both utterances. An example illustrating how multi-word matches

can arise is shown in Figure 3-11. Here, the utterances being compared are

(1) "act one of John Nash's drama"

(2) "now fortunately for John Nash, and I think, the rest of us".

The low distortion region for this distance matrix corresponds to the multi-word phrase "John Nash". The point of Figures 3-10 and 3-11 is to illustrate that segmental DTW is, in the end, a low-level algorithm. That is, with no knowledge of the constituent lexical items in the audio stream, we can not hope to identify whether a single fragment endpoint corresponds to the boundary of a word or of a subword unit. Two possible ways to make use of these path fragments are to incorporate higher level knowledge via human interaction, or aggregate the outputs of many other path fragments. We investigate both of these ideas in subsequent chapters.

3.5 Summary

In this chapter, we presented a method for finding similar regions between two utterances directly at the acoustic level. The method used is an extension of the well-known DTW algorithm used for whole word alignment. In contrast to global alignment, we make use of multiple constrained local alignments to account for the possibility of subsequence matches. The warp paths generated by this segmental DTW procedure is able to find common subsequences between multi-word utterances at the subword, word, and phrase level. In the next chapter, we will demonstrate how the segmental DTW algorithm can be applied beyond utterance pairs, to entire sets of audio segments representing full-length audio streams and provide some quantitative evaluations of how well it is able to match common words between utterances.

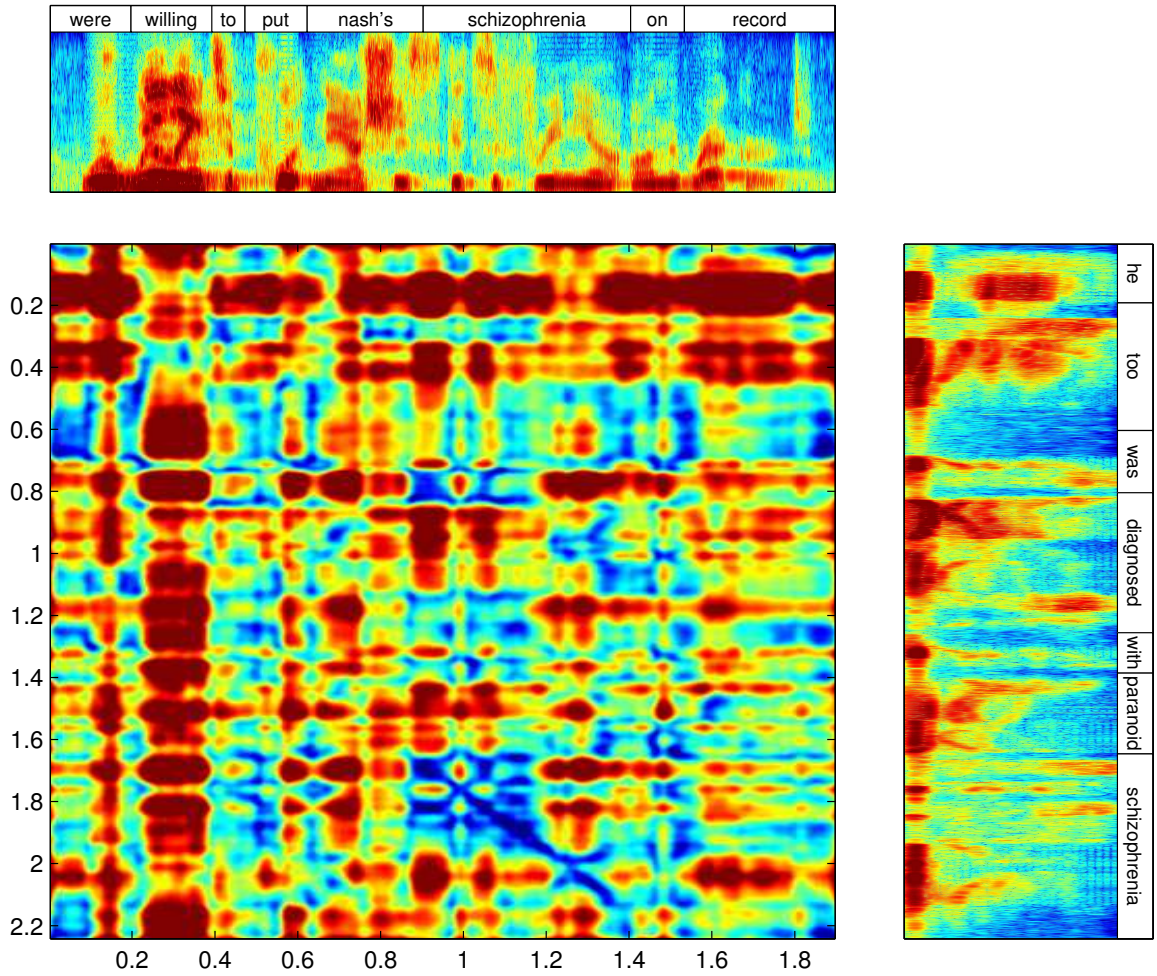


Figure 3-6: Distance matrix for two utterances spoken by a female speaker and their corresponding spectrograms. The first utterance, shown horizontally across the top, is the phrase, "were willing to put Nash's schizophrenia on the record". The second utterance, shown vertically along the right-hand side, is the phrase, "he too was diagnosed with paranoid schizophrenia".

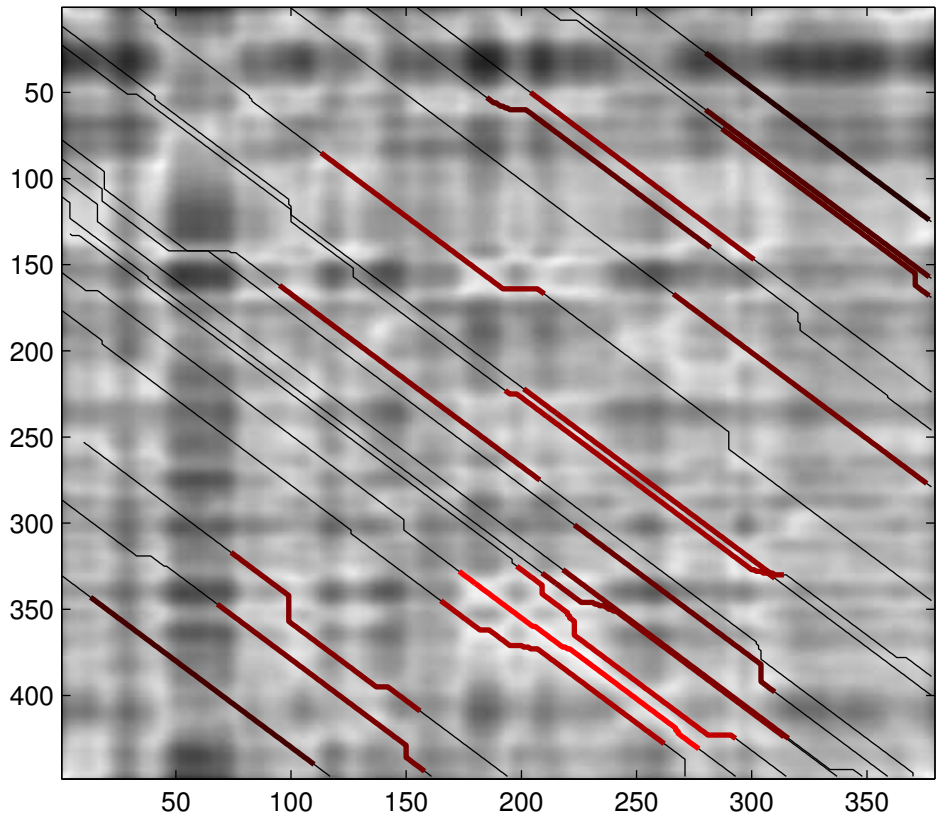


Figure 3-7: The family of constrained warp paths $\hat{\phi}_r$ with R = 10 for the distance matrix in Figure 3-6. The frame rate for this distance matrix is 200 frames per second. The associated LCMA path fragments, with L = 100, are shown as part of each warp path. The color of each path fragment is an indicator of the average distortion for that path fragment, with black corresponding to higher distortion and red corresponding to lower distortion.

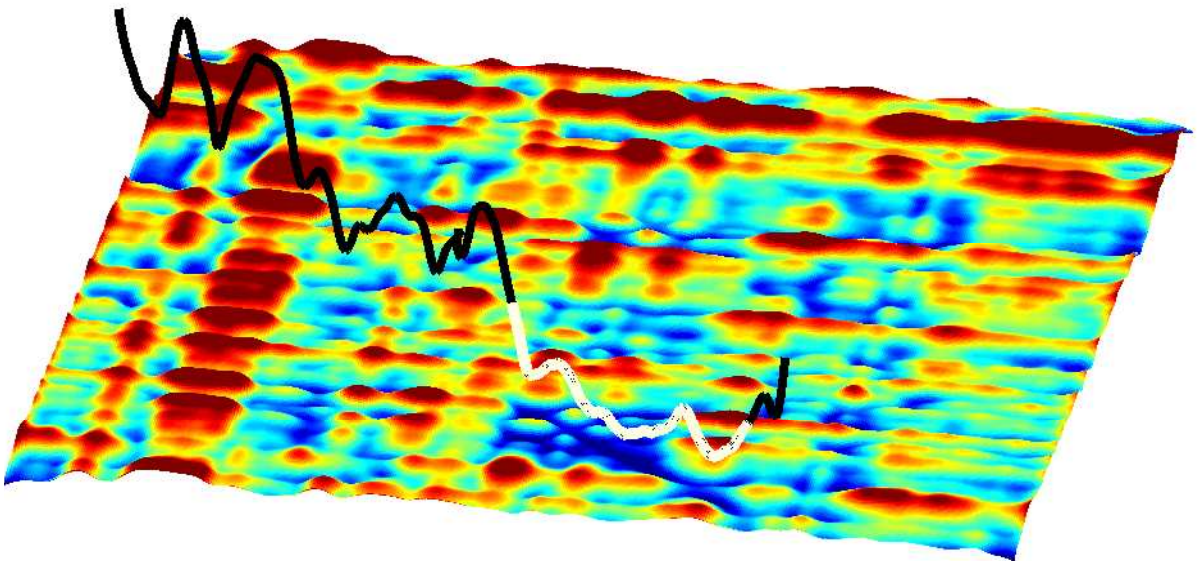


Figure 3-8: Three dimensional relief view of the distance matrix from Figure 3-6 overlaid with the lowest distortion warp path from Figure 3-7.

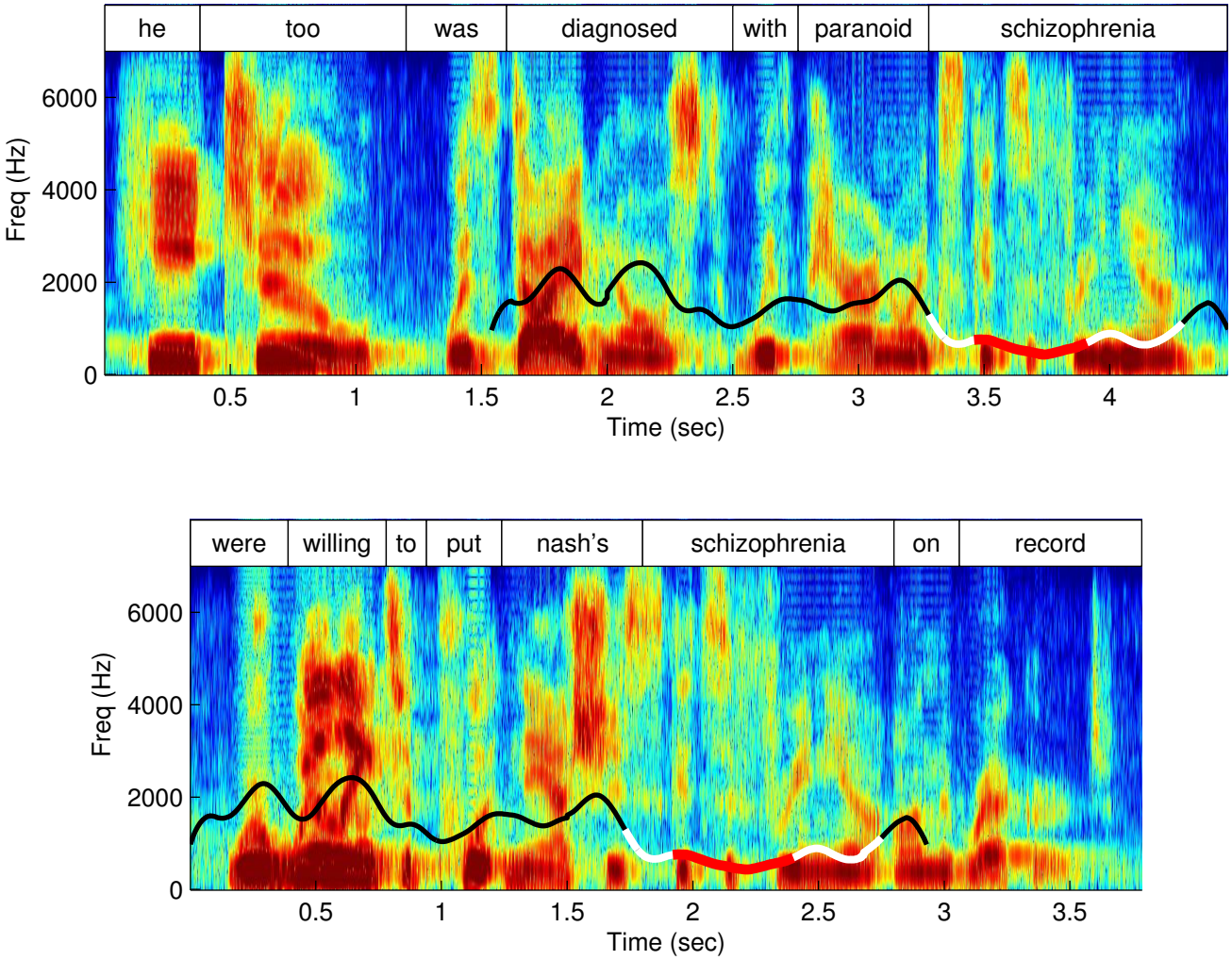


Figure 3-9: Utterance level view of the warp path from Figure 3-8. The red line corresponds to the LCMA fragment for this particular warp path, while the white line corresponds to the fragment resulting from extending the LCMA fragment to neighboring regions with low distortion.

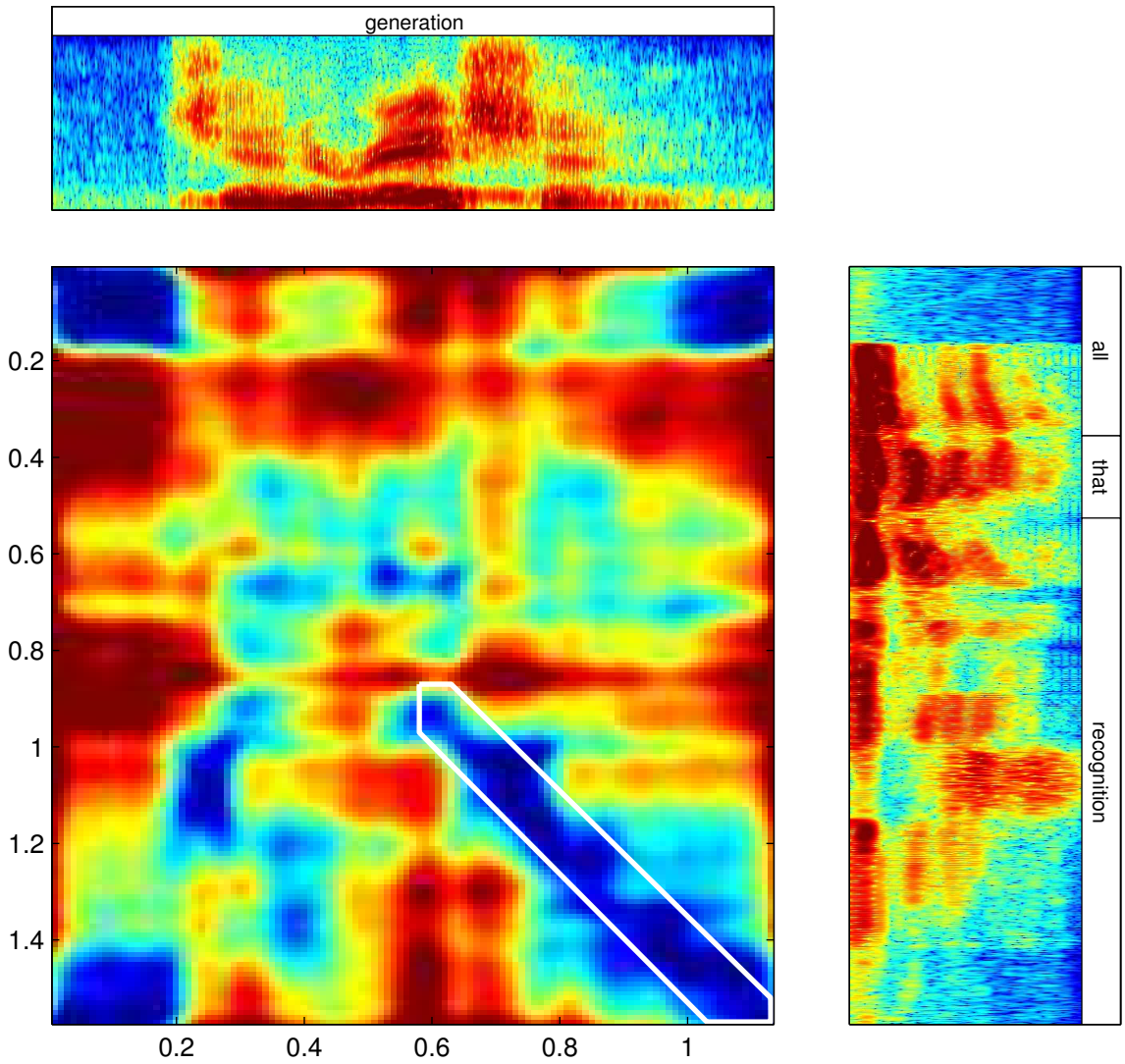


Figure 3-10: Distance matrix for two utterances spoken by a female speaker and their corresponding spectrograms. The first utterance, shown horizontally across the top, is the word, "generation". The second utterance, shown vertically along the right-hand side, is the phrase, "all that recognition".

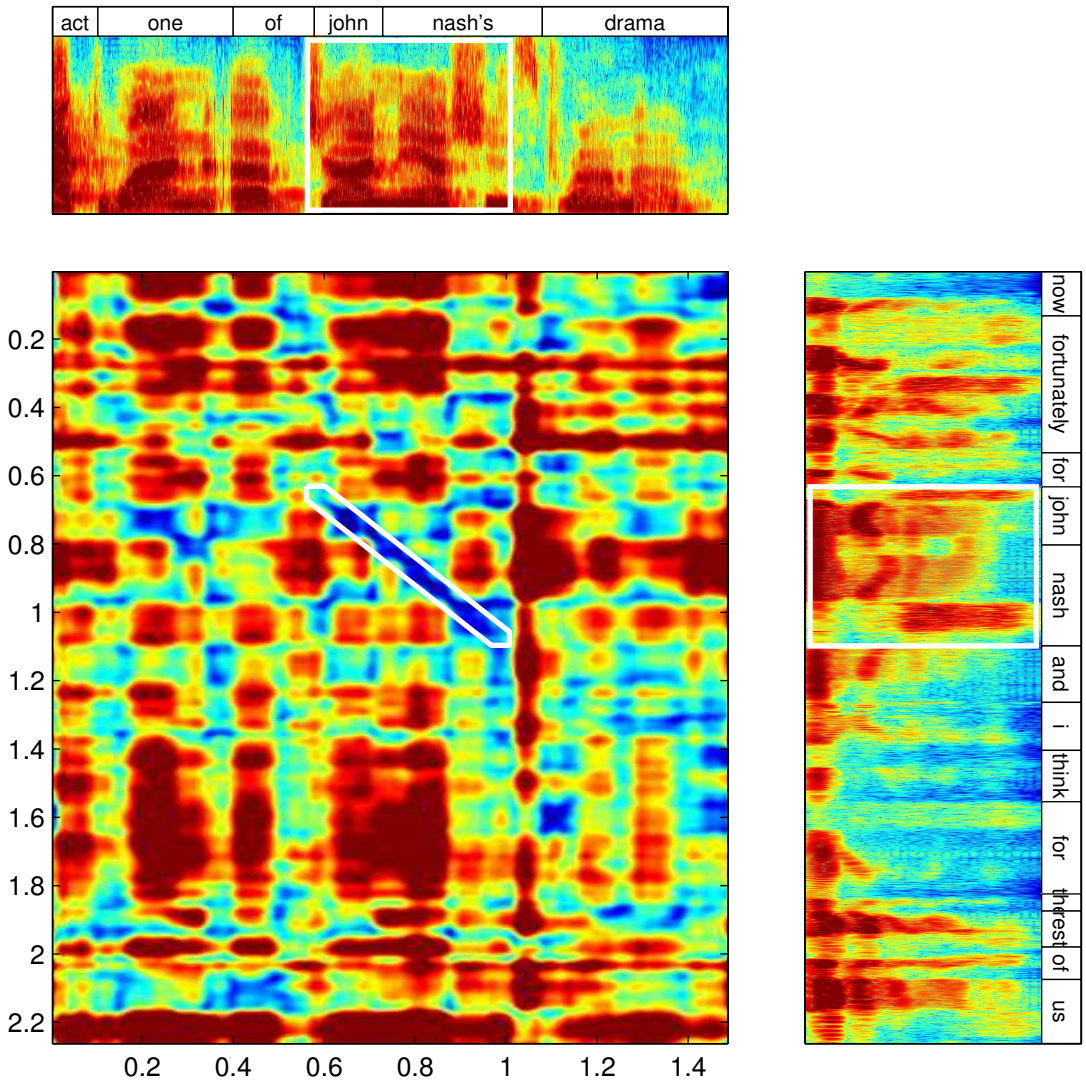


Figure 3-11: Distance matrix for two utterances spoken by a female speaker and their corresponding spectrograms. The first utterance, shown horizontally across the top, is the phrase, "act one of John Nash's drama". The second utterance, shown vertically along the right-hand side, is the phrase, "now fortunately for John Nash, and I think, the rest of us".

Chapter 4

Segmental DTW: Analysis

The goal of this chapter is to provide a quantitative understanding of what alignment path fragments are and what distinguishes 'good' fragments from 'bad' fragments, in the context of finding acoustically similar sequences. To that end, we conduct a series of experiments to examine path characteristics and matching accuracy when evaluated on relatively large sets of data. In the first part of this chapter, we evaluate the accuracy of matches that are found at the subword level by comparing phone level transcriptions of short alignment paths using the single-speaker Ice Cream corpus. In the second part of this chapter, we evaluate matches induced by longer path fragments at the word level using several lectures from the lecture corpus.

4.1 Segmental DTW on Multiple Utterances

In the previous chapter, we explained the steps involved in using segmental DTW to find similar segments of speech between two multi-word, or possibly single word, utterances. Extending this operation to multiple utterances, like the ones constituting the Ice Cream corpus, or the ones generated by segmenting an audio

lecture, requires performing segmental DTW on each pair of utterances in the set. For a set of N segments, the number of segmental DTW operations will be

$$\#\text{SegDTW} = \binom{N}{2}. \tag{4.1}$$

As discussed in the previous chapter, each application of the segmental DTW algorithm will itself yield multiple warp path fragments, with the exact number depending on the duration of the utterances being compared and the size of the band constraint parameter R. Over the entire set of utterances, therefore, we can expect to have a large number of warp path fragments generated from performing pairwise segmental DTW. In practice however, only a fraction of the path fragments that are generated are actually candidates for reasonable alignments. This is because the majority of path fragments are the result of choosing the optimal warp path between two segments of speech that do not align well together. The average path distortion can be used as a criterion for determining the reliability of a particular path fragment. While performing segmental DTW, we typically use a distortion threshold to prune away path fragments with high distortion.

One concern regarding this stage of processing is that performing path detection over all pairs of utterances might be computationally prohibitive. Equation 4.1 indicates that the number of operations will be at least quadratic in the number of utterances. However, we have observed that in practice, the amount of time required for the comparisons is similar to performing recognition. Moreover, because each pairwise comparison is independent of the other comparisons, the entire computation can be easily parallelized for further gains. We will discuss computational issues further in Chapter 8.

4.2 Path Fragments: Phonetic Analysis

The first experiment we conducted was designed to observe how well the segmental DTW algorithm discovered matches between acoustic sequences at the subword level. For this the phone matching experiment, we performed segmental DTW on the entire set of 720 utterances in the Ice Cream corpus.

The Ice Cream corpus includes manually determined phonetic transcriptions for each of the utterances in the corpus. The 61 phone labels used for phonetic transcription of the corpus are shown in Table 4.1. For evaluation purposes, these 61 labels are typically collapsed into 39 phone class labels prior to being compared in classification experiments [60]. The mapping of the 61 phone labels into into the 39 scoring classes is shown in Table 4.2.

In computing the segmental DTW path fragments, we used whitened MFCC spectral vectors computed at a 10 millisecond analysis rate. We imposed a

IPA	ARPAbet	Example	IPA	ARPAbet	Example
ɑ	aa	bob	ɨ	ix	debit
æ	ae	bat	iʸ	iy	beet
ʌ	ah	but	ǰ	jh	joke
ɔ	ao	bought	k	k	key
ɑʷ	aw	bout	k˺	kcl	k closure
ə	ax	about	l	l	lay
əʰ	ax-h	potato	m	m	mom
ɚ	axr	butter	n	n	noon
ɑʸ	ay	bite	ŋ	ng	sing
b	b	bee	ɾ̃	nx	winner
b˺	bcl	b closure	oʷ	ow	boat
č	ch	choke	oʸ	oy	boy
d	d	day	p	p	pea
d˺	dcl	d closure	-	pau	pause
ð	dh	then	p˺	pcl	p closure
ɾ	dx	muddy	ʔ	q	glottal stop
ɛ	eh	bet	f	r	ray
l̩	el	bottle	s	s	sea
m̩	em	bottom	ʃ	sh	she
n̩	en	button	t	t	tea
ŋ̩	eng	washington	t˺	tcl	t closure
-	epi	epenthetic silence	θ	th	thin
ɝ	er	bird	ʊ	uh	book
eʸ	ey	bait	uʷ	uw	boot
f	f	fin	ü	ux	toot
g	g	gay	v	v	van
g˺	gcl	g closure	w	w	way
h	hh	hay	y	y	yacht
ɦ	hv	ahead	z	z	zone
ɪ	ih	bit	ʒ	zh	azure
-	h#	utterance final			

Table 4.1: IPA and ARPAbet symbols for the 61 phones occurring in the Ice Cream corpus with example words indicating their pronunciation.

ARPAbet	Example	ARPAbet	Example
1	iy	20	n en nx
2	ih ix	21	ng eng
3	eh	22	v
4	ae	23	f
5	ax ah ax-h	24	dh
6	uw ux	25	th
7	uh	26	z
8	ao aa	27	s
9	ey	28	zh sh
10	ay	29	jh
11	oy	30	ch
12	aw	31	b
13	ow	32	p
14	er axr	33	d
15	l el	34	dx
16	r	35	t
17	w	36	g
18	y	37	k
19	m em	38	hh hv
39	bcl pcl dcl tcl gcl kcl q epi pau h#		

Table 4.2: Mapping of 61 phones from Table 4.1 into 39 classes used for evaluation.

minimum length constraint of $L = 5$ (50 milliseconds) for each path fragment and used a band constraint factor of $R = 2$ (20 milliseconds). With a pruning threshold of 2.0 on the path distortion, a total of 921K path fragments were retained after the segmental DTW phase. The distributions of average path distortion and path length for these fragments is shown in Figure 4-1. The histograms shown indicate that the set of retained path fragments are heavily populated by fragments with high distortion and short path length. Although path distortions range from a minimum of 0.93 through to the prune threshold of 2.0, the majority of fragments have high distortion. Likewise, path lengths range between 5 and 57 frames, but over 80% of the paths are shorter than 10 frames. Since the average phone length in this corpus was close to 7 frames, many of the path fragments consisted of multi-phone sequences.

The matching accuracy of each path fragment was evaluated by computing the frame level accuracy of each fragment. Manually labeled phonetic transcriptions for each utterance were used for this scoring procedure, which is illustrated in Figure 4-2. Every step in the warp path fragment was categorized as either correct or incorrect based on whether the phone transcription on either side of the alignment belonged to the same class according to the mapping in Table 4.2. This scoring procedure essentially measures the precision of the frame pairs that are returned by the segmental DTW algorithm.

Our reasons for considering only precision and not recall are twofold. From a pragmatic perspective, the number of true matching frame pairs in the corpus is large enough we would have to significantly increase the number of paths found in order to effect a meaningful difference in recall. More philosophically, our goal in performing information discovery is fundamentally different from what we might want to accomplish by performing information retrieval. Since we seek to use the information about matching paths to help us find structure in the data, the cost of a missed detection is less than the cost of a false positive. While the former contributes nothing to our body of knowledge, the latter can corrupt it. Thus, in the context of our task, it is more important to have high precision among the frame pairs that we discover than to exhaustively find all matching frame pairs.

4.2.1 Analysis

Out of approximately 7.4M total frames matched across the 921K path fragments, the overall frame accuracy was 57.9%, with 4.3M of these frames being matched correctly. The frame accuracy, however, is strongly related to both the path length and the path distortion. In Figure 4-3, the average frame accuracy of groups of path fragments are plotted according to length and average distortion. The highest frame accuracy, 76%, is obtained for the 10K fragments which had an average distortion of 1.553 and lengths greater than 11 frames.

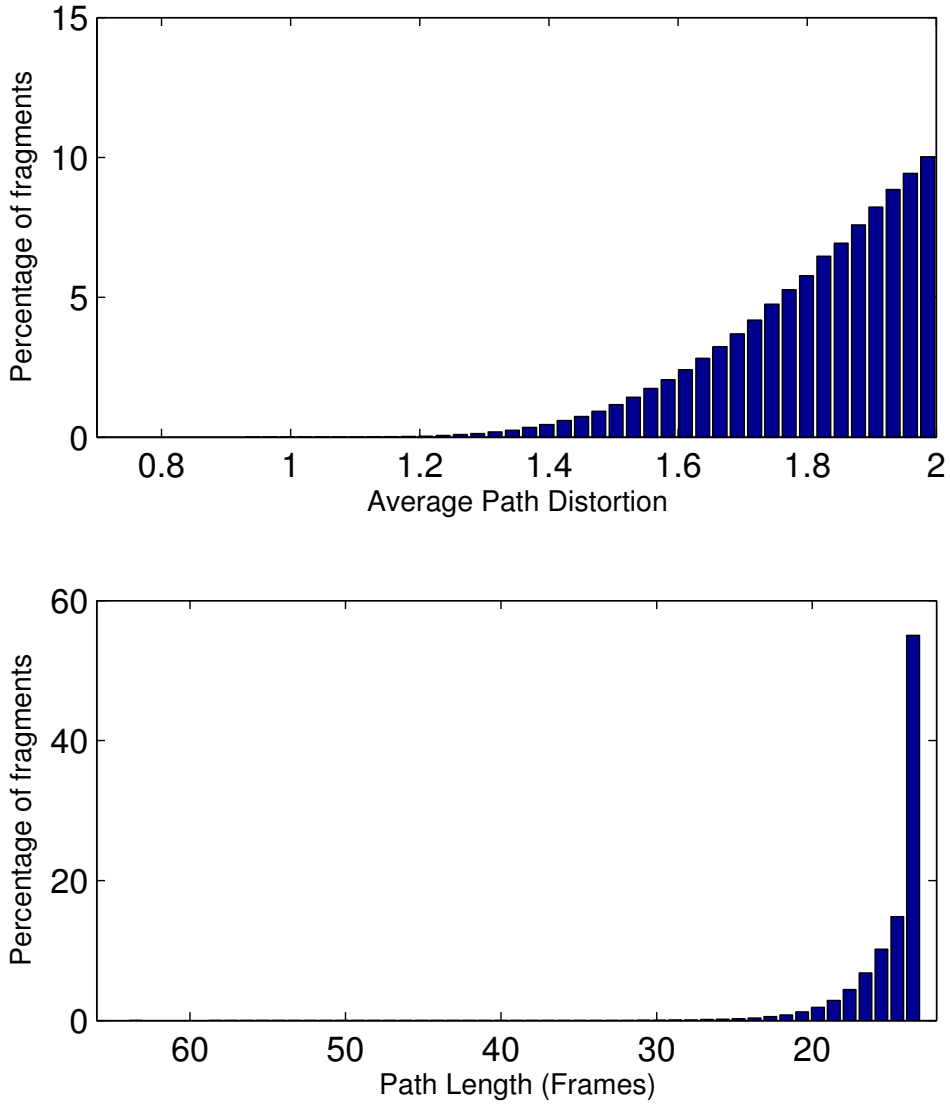


Figure 4-1: Histograms showing the distribution of the 921K path fragments according to average path distortion and path length. The distortion threshold of 2 was used to prune the generated fragments and is therefore the upper limit for fragments in the distortio histogram.

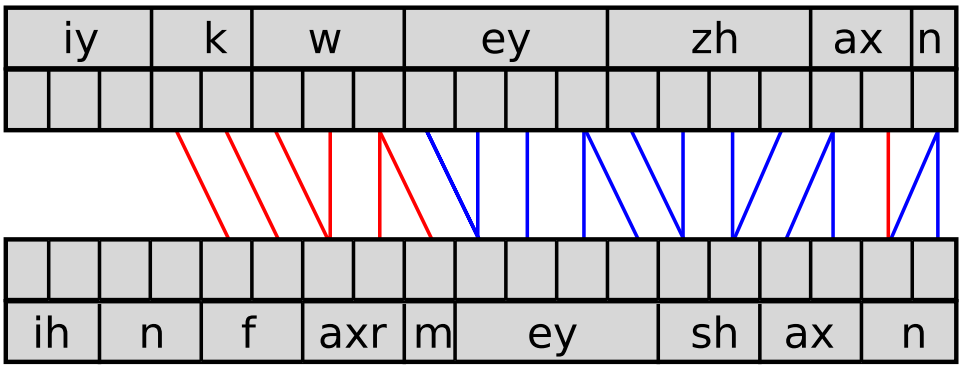


Figure 4-2: Illustration of the alignment scoring procedure. In the above example, fragments of the words "equation" and "information" are aligned together. The blue lines indicate correctly frame alignments, while the red lines indicate incorrect alignments. The alignment of "sh" with "zh" is considered correct as they both belong to the same class according to the mapping in Table 4.2.

The lowest frame accuracy, 48.3%, is obtained for the 10K fragments which had an average distortion of 1.998 and length equal to 8 frames. This set of results indicate that when distortion is held constant, longer paths typically have better frame accuracy. Furthermore, when length is held constant, the paths with lower distortion had better frame accuracy.

In order to obtain a more detailed understanding of the *types* of matches found, we analyzed a set of low distortion paths more closely. For the 97K path fragments with distortion less than 1.6, 856K frames were matched and the overall frame accuracy was 68.7 %. We then examined the distribution of these correctly matched frames amongst the various phone classes to learn which classes were more likely to induce matches via the segmental DTW algorithm.

First, we estimated the number of reference frame matches across phone classes by using manual phonetic transcriptions. The number of inter-utterance token matches of each phone class was computed by counting the pairwise matches of each phone occurrence between all pairs of utterances. This number, the token match count, was then multiplied by the average frame duration of each phone class to get an approximate value of the reference frame match count across the corpus for that particular class. The total number of these reference frame matches was 83M frames, with approximately 50% of these frame matches belonging to the final phone class in Table 4.2, which consists of silence-like units such as closures and epenthetic silences. For brevity, we will refer to this as the 'CL' class.

Out of the 588K correct frame matches produced by the segmental DTW algorithm, 84.5% of these were of the 'CL' class. This percentage is significantly larger than the expected 50% that one would expect if matches were equally

drawn from amongst all phone classes. A similar bias can be observed in other classes by viewing the phone class recall rates, as shown in Figure 4-4. Rather than being equally distributed across the phone classes, we observe that recall rates are significantly higher for labial stop bursts ('p' and 'b'), alveolar fricatives and affricates ('sh', 'jh', 'ch'), and the 'CL' class. In general, we can see that stops and fricatives have higher recall rates than vowels.

One possible explanation for the proportionally higher recall rates for the 'CL', labial burst, and affricate classes may be in the consistency of their realization. Since the 'CL' class is predominantly made up of silent regions, there is likely to be less variation among the various occurences of this class, which may result in lower distortion for warp paths along these regions. This consistency observation can also hold true for labial bursts, which typically have very little energy, and the 'ch/jh/sh' classes, which have very stable and distinct spectral profiles that are relatively invariant within a single speaker. The disparity in recall rates for these classes may also be explained by characteristics of their spectral representation which result in them having smaller within class distortion. This hypothesis is partially validated by observing the within class average frame distortions plotted in Figure 4-5. The phone classes with the lower within class distortion values tend to be the same classes which yield higher recall rates.

The uneven distribution of recall rates across the various phone classes indicates that path fragments found using a short length constraint are biased towards finding particular types of phone classes. While this may be an artifact of the spectral representation used, it may also be a result of larger inherent variability in the acoustic realization of some phone classes over others. The limited variety of matches found indicates that using segmental DTW alone may not be suitable for inducing an inventory of subword units from a speech corpus. However, this experiment did show that high frame accuracy is correlated to both path length and distortion. In the next section, we consider path fragments at a coarser granularity by increasing the minimum length constraint.

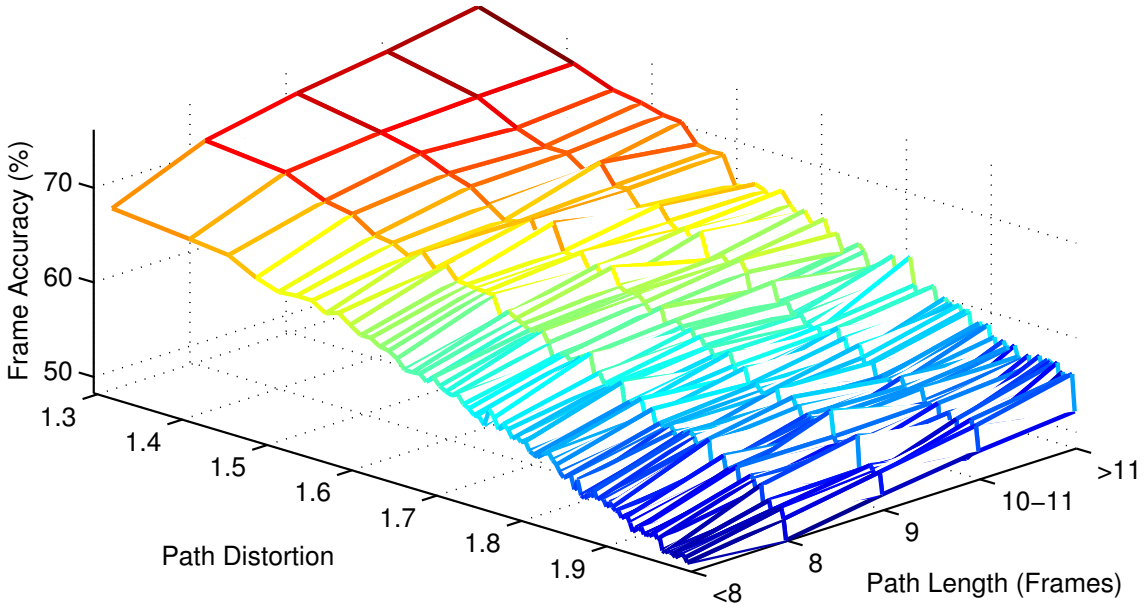


Figure 4-3: Phone accuracy in terms of percentage of frames correctly matched versus path distortion and path length.

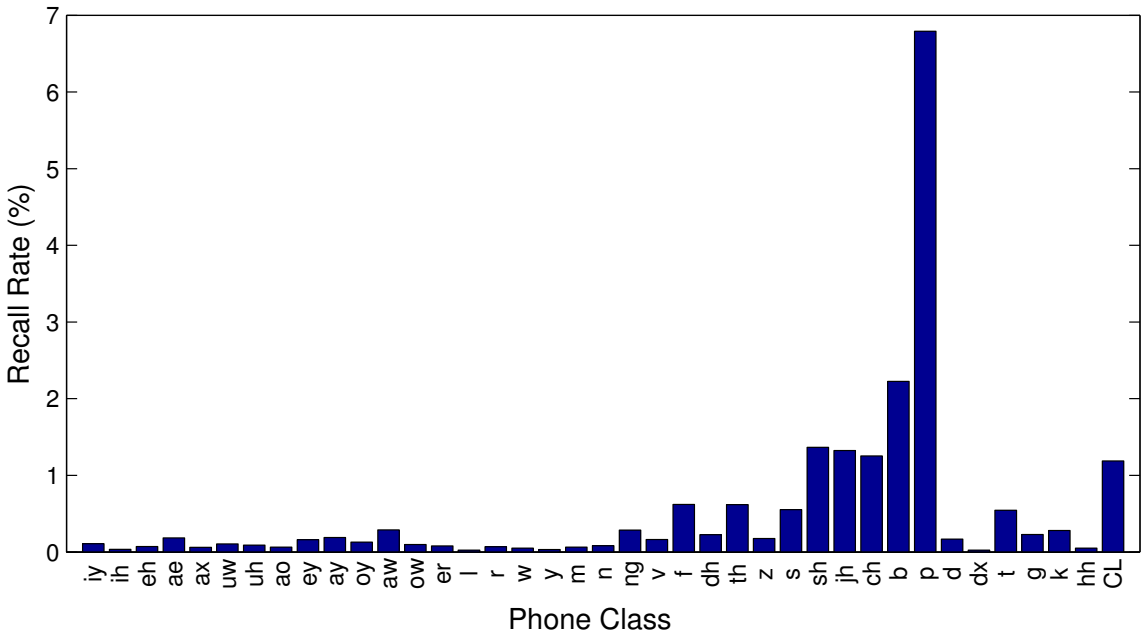


Figure 4-4: Frame level recall rates for each phone class.

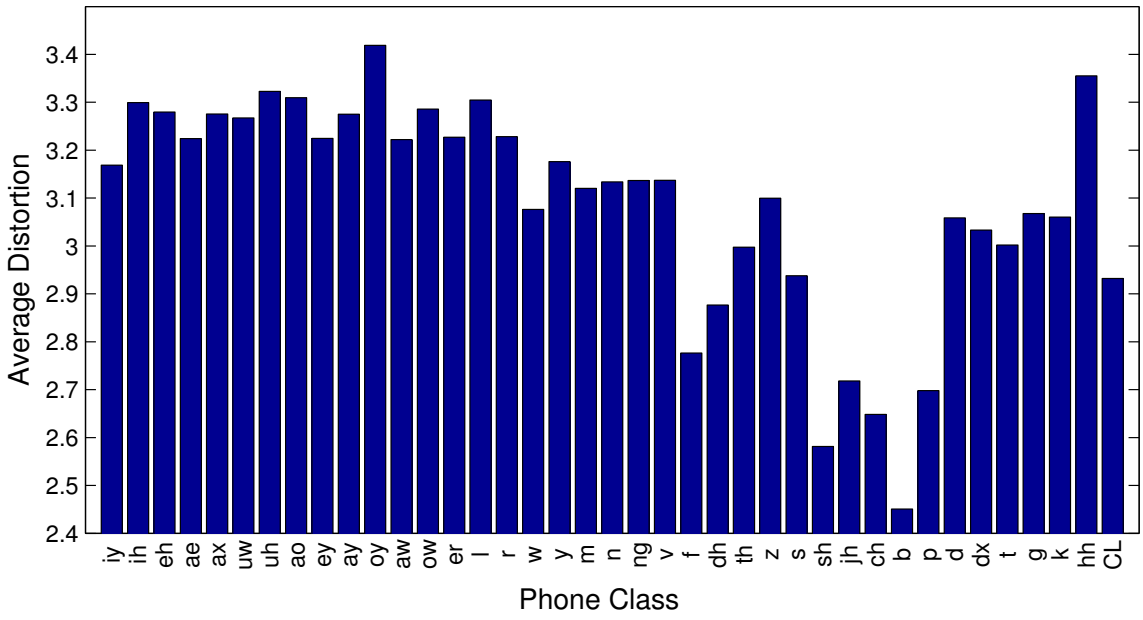


Figure 4-5: Average inter-utterance frame distortion values for each phone class.

Lecture	Length	# of Segments	# of Fragments Found
Physics E& M Lecture	51 mins	1510	255139
Linear Algebra Lecture	47 mins	1260	148446
Friedman Lecture	1 hr 15 mins	2029	238525
ASR Course Lecture 2	1 hr 25 mins	2798	474599
ASR Course Lecture 6	1 hr 18 mins	2042	485837
ASR Course Lecture 19	1 hr 14 mins	1905	351802

Table 4.3: Lecture characteristics including number of path fragments found using a distortion pruning threshold of 3.

4.3 Path Fragments: Word Analysis

In this section, we examine the quality of acoustic matches at the word level by modifying the parameters of the segmental DTW algorithm to search for longer path fragments. The minimum length criterion of $L = 5$ used in the previous section resulted in matches found mostly at the subword level, with few path fragments reaching durations characteristic of words or phrases. In this set of experiments, we expand the length criterion tenfold, using a minimum length of $L = 50$ and search for path fragments that are more representative of word-level matches.

For our word-level experiments, we processed three lectures using the segmental DTW algorithm. The lectures used were the Physics, Linear Algebra, and Thomas Friedman lectures described in Table 2.1. Characteristics of these lectures, together with the number of path fragments found for each are shown in Table 4.3. The number of fragments produced is roughly proportional to the length and number of segments in the lecture. Length and distortion distributions of the paths are shown in Figure 4-6. One interesting observation is that the number of path fragments generated for each of these lectures is a fraction of what was generated in the previous experiment, despite the greater amount of speech in the lectures compared to the Ice Cream corpus and the higher pruning threshold. This illustrates that the minimum length criterion can act as a secondary pruning mechanism in combination with the distortion threshold – as the minimum length increases, there exist fewer path fragments that can simultaneously satisfy the length criterion while having a low average distortion. The distribution of path fragments found for the lectures processed here are similar to those seen for the Ice Cream corpus data. We find that the majority of fragments found tend to be at the upper end of the distortion spectrum and at the lower end of the length spectrum.

In Table 4.4, the 10 path fragments with lowest distortion from the Physics lecture are shown. The table illustrates how each path fragment is associated with two time intervals from the original audio stream and their underlying

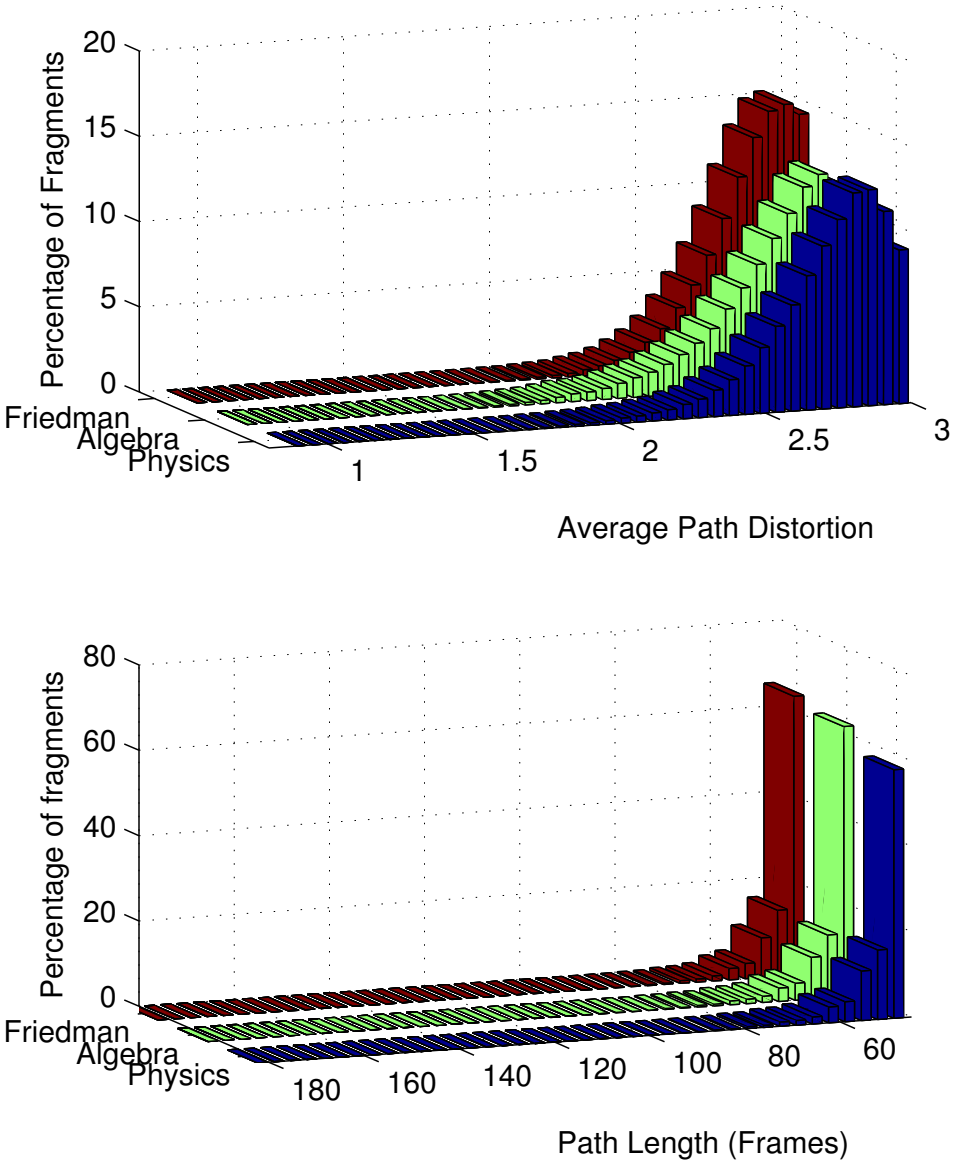


Figure 4-6: Distribution of path fragments according to length and distortion for the first three lectures in Table 4.3.

transcriptions. The matches found by these fragments vary widely, both in the number and types of words matched. In fragments 4 and 9, entire phrases are matched, while fragments 7 and 8 show examples of single word matches. The types of words found populating these paths include content words as well as function words, but with the exception of fragment 8, most shorter function words such as "and", "is", or "the", tend to appear as part of a longer phrase match rather than as single word matches on their own. This phenomenon is likely due to the short duration of these words, since most of these shorter words are significantly reduced in conversational speech.

A more macroscopic perspective of the types of matches found can be seen by considering word occurrence statistics for a large number of fragments. Table 4.5 lists the top 35 words by path fragment occurrence for each lecture. A word's occurrence count is incremented for each path where it occurs as part of *both* path intervals. There are several interesting observations we can make about this table. First, for all the lectures, many function words have high rank, but the average word length of paths including these words is also high compared to other words. This indicates that short function words occur more often as part of a multi-word phrase rather than by themselves. In the Physics and Linear Algebra lectures, the top ranked terms are strongly relevant to the lecture topic and appear in a high number of fragments – many more than their nearest competitors. In contrast, the top ranked word for the Friedman lecture occurs in only 205 path fragments, with topic relevant words like "globalization" appearing in even fewer fragments. The main reason for the disparity in frequencies between the academic lectures and the Friedman lecture becomes clear when one considers the relative vocabulary sizes. While the Physics and Linear Algebra lectures have compact vocabularies of 815 and 651 words, respectively, the Friedman lecture uses 2020 words, indicating greater breadth of topic material and likely less repetition of any single phrase to the extent seen in the academic lectures.

	Distortion	Interval 1		Interval 2		Duration
		Time	Transcription	Time	Transcription	(ms)
1	0.89	1:40	rectangle three	1:59	rectangle and	740
2	1.14	22:33	touch it again with my finger	22:36	touch it again with my finger	1290
3	1.25	4:38	r square	4:51	r square	790
4	1.26	15:47	divided by two epsilon zero	16:00	divided by two epsilon zero	1480
5	1.27	23:45	go inside	23:46	go inside	760
6	1.31	15:45	sigma	15:47	sigma	490
7	1.35	15:42	two epsilon zero	15:45	two epsilon zero	1180
8	1.35	8:23	argument	13:36	argument	500
9	1.36	23:01	ones have it	23:29	left have it	490
10	1.39	4:24	is a sphere	5:32	take a sphere	740
11	1.40	8:14	like this	8:14	like this	580
12	1.41	7:20	what is the electric field	12:26	what is the electric field	1280
13	1.41	14:32	divided by two	16:00	divided by two	900
14	1.42	2:13	case	7:35	this	500
15	1.42	15:41	sigma	16:07	is sigma	600
16	1.42	14:31	sigma	15:47	sigma	500
17	1.42	15:42	two epsilon zero	15:55	by two epsilon zero	1160
18	1.42	2:09	flux	5:54	flux	600
19	1.43	14:32	divided by two	15:55	divided by two epsilon	910
20	1.43	19:46	thirty centimeters away from the	20:23	thirty centimeters away from the	1250

Table 4.4: Top 20 path fragments ranked by average path distortion for the Physics lecture. The leftmost column indicates the fragment index. Distortion refers to the average distortion along the path, and Duration refers to the total duration of the path in milliseconds. The columns labeled Interval 1 and Interval 2 provide information about the speech segments aligned by the path fragment. The time of each interval refers to the global time index of the interval's start, and the transcription is the reference word sequence for that segment

Physics			Linear Algebra			Friedman		
Counts	**Length**	**Word**	**Counts**	**Length**	**Word**	**Counts**	**Length**	**Word**
934	2.74	electric	1618	1.90	matrix	205	3.42	the
930	2.72	field	259	3.03	the	140	1.62	⟨um⟩
496	3.35	the	174	2.44	this	131	3.19	world
343	1.59	surface	124	2.79	that	126	2.47	and
264	2.18	this	96	2.31	three	112	1.65	globalization
169	1.62	inside	91	1.48	matrices	88	3.35	of
164	3.52	is	81	1.88	and	76	2.14	collaboration
133	2.14	plane	76	2.05	two	67	3.63	to
132	2.31	charge	74	2.45	is	64	2.06	people
122	2.98	that	73	2.23	step	53	2.22	this
109	2.55	zero	70	1.69	subtract	44	1.32	platform
89	2.71	epsilon	68	2.16	one	36	3.42	in
89	1.87	sphere	65	1.60	elimination	36	1.82	country
88	2.94	divided	59	3.00	first	34	2.34	eleven
87	2.99	by	59	2.67	what	33	3.39	you
71	3.04	a	56	2.72	so	32	3.45	is
68	3.44	and	54	2.83	i	32	3.11	flat
63	2.27	close	51	1.67	equation	32	2.36	so
54	2.81	r	50	2.37	pivot	31	2.45	nine
52	2.68	square	48	2.19	time	30	3.60	form
50	1.67	symmetry	48	2.04	multiplication	30	2.73	that
43	3.17	of	45	2.98	row	30	1.87	flattener
39	3.76	two	43	2.01	column	29	3.34	i
35	2.73	d	43	1.77	multiply	28	1.32	application
35	2.10	uniformly	33	3.26	right	26	2.75	more
35	1.57	sigma	26	3.63	side	23	3.11	first
34	2.19	distance	26	3.10	a	21	3.74	a
32	3.98	you	24	3.77	hand	21	3.12	just
32	3.95	it	24	3.08	to	21	2.83	work
32	3.42	like	23	2.39	take	21	1.43	outsourcing
32	3.00	here	21	2.29	suppose	21	1.43	convergence
31	3.68	i	21	2.00	zero	19	2.68	said
31	1.71	argument	21	1.76	position	19	2.39	china
30	4.12	if	20	3.30	it's	19	2.32	before
30	3.90	same	20	1.70	operation	19	2.03	imagination

Table 4.5: Top 35 words ranked by their occurrence in the 5000 lowest distortion path fragments for each lecture. The "Count" column for each lecture indicates how many times the word occurred in both intervals of a path fragment. The "Length" column indicates the average word length of the path fragments in which the word appeared.

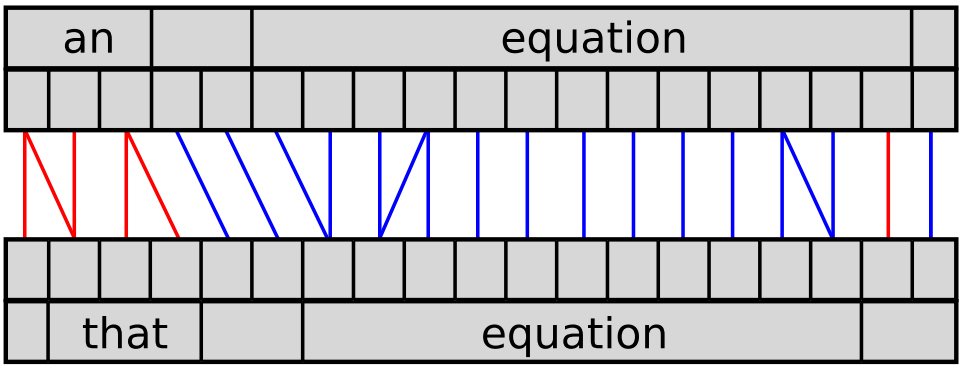


Figure 4-7: Illustration of the frame alignment scoring procedure at the word level. Lines between the two frame sequences indicate the alignment produced by an alignment path fragment. The lines are marked blue or red, corresponding to correct matches and incorrect matches, respectively. Matches between non-word frames are counted as correct.

4.3.1 Path Accuracy

Using time aligned orthographies, we measured the frame level accuracy of the discovered paths at the word level. The scoring procedure was similar to that used for phone level scoring and is shown in Figure 4-7. We used this scoring metric to evaluate the quality of matches at the word level and plotted the average frame accuracy for groups of path fragments at different distortions and lengths in Figure 4-8.

In all three lectures, the frame accuracies followed similar patterns to the phonetic frame accuracies observed in the previous section - lower distortion paths had higher frame accuracies, with accuracies near the pruning threshold approaching 0%. At each distortion level, the longer paths in the group had better frame accuracies, with the optimal group of paths from the Physics, Algebra, and Friedman lectures achieving frame accuracies of 78.7%, 80.7% and 72.6%, respectively. The shorter paths for the same distortion level had frame accuracies of 57.5%, 40.3%, and 34.7%.

The higher word-level frame accuracy of longer path fragments can be partially explained by surveying the types of matching errors seen among the discovered fragments. A list of path fragments with 0% matching accuracy are shown in Table 4.6. Some of these examples, such as fragments 7 and 8, are not actual errors, but are counted as such due to errors in our word stemming procedure prior to evaluation. In general, paths are routinely discovered which match morphological variations of the same word due to the acoustic similarity of their stems. For the majority of the other fragments on this list however, the word lengths are longer than the minimum length criterion, which results in alignment paths being found between common prefixes and suffixes of the

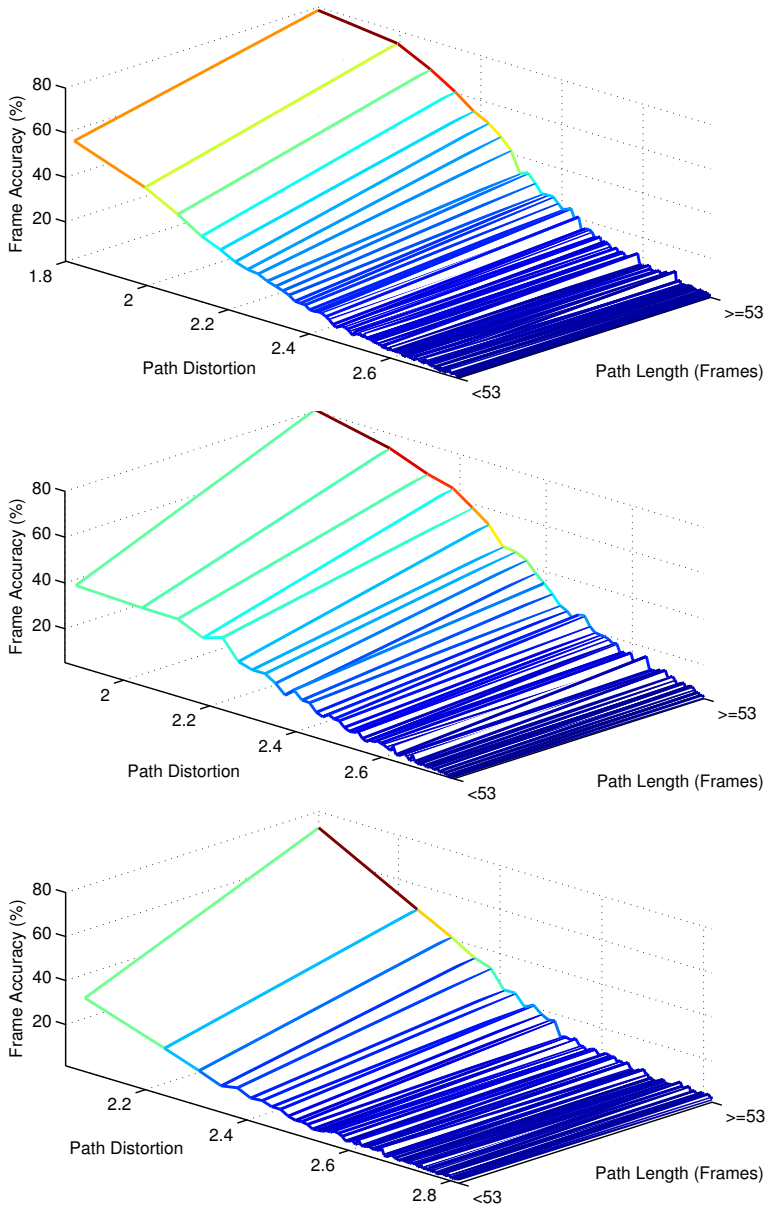


Figure 4-8: Frame level word accuracies plotted for different groupings of path fragments according to path lengths and path distortions. Each distortion level represents a group of 1K fragments which are grouped by length. Top: Physics. Middle: Linear Algebra. Bottom: Friedman.

words, such as "any-" of"anyone" and "anymore", and the "-ation" of words like "globalization", "application", and "collaboration". Although these matches may be marked as errors at the word level, they are very similar acoustically, which implies higher matching accuracy at the phonetic level.

4.4 Summary

In this chapter, we have provided some qualitative and quantitative evaluations of the path fragments generated by the segmental DTW process. One main result that we noted in both the phone-level and word-level experiments was the relationship of path length and path distortion to matching accuracy. We consistently observed that longer path fragments and path fragments with lower distortion had higher matching accuracy. Since the number of paths produced during pairwise comparison of multiple utterances can be very large, pruning high distortion paths can keep the number of total paths to a more manageable level. For the purpose of discovering subword units, the results of our phonetic experiments are initially discouraging, as the distribution of matching path fragments is biased towards certain phone classes. Our word-level experiments yielded more encouraging results. In particular, we found that high matching accuracy can be achieved if using low distortion paths. Among the correct matches generated, the many instances of topic relevant words among the matches found indicates that this technique may prove useful for extracting knowledge from audio streams.

	Distortion	Interval 1		Interval 2		Duration
		Time	Transcription	Time	Transcription	
1	1.72	3:26	conversation	16:41	globalization	48
2	1.74	52:55	riches	1:06:06	niches	57
3	1.74	25:24	collaboration	50:17	variation	47
4	1.77	3:26	conversation	17:25	globalization	51
5	1.77	18:14	globalization	26:26	organization	47
6	1.79	47:14	collaboration	50:17	variation	47
7	1.80	21:38	overinvestment	21:48	investment	64
8	1.83	23:21	suddenly	58:48	sudden and	53
9	1.84	18:14	globalization	20:21	civilization	52
10	1.85	3:26	conversation	15:54	globalization	47
11	1.88	18:02	globalization	20:21	civilization	58
12	1.90	18:02	globalization	26:26	organization	47
13	1.90	59:20	direction	59:47	protection	55
14	1.91	33:21	anyone	33:22	anywhere	47
15	1.92	9:44	this	33:24	device	48

Table 4.6: Fifteen lowest distortion path fragments with matching errors taken from the Friedman lecture.

Chapter 5

Word Acquisition via Clustering

In this chapter, we build on the segmental DTW techniques discussed in previous chapters and present an unsupervised method for automatically discovering words from speech using graph clustering and isolated word recognition.

5.1 From Paths to Clusters

The matching experiments from the previous chapter demonstrated that, with appropriate choice of length constraint, the segmental DTW algorithm will produce a large number of alignment path fragments that are distributed throughout the audio stream. As we saw in the previous chapter, each alignment path consists of two intervals (the regions in time purported to be matching), and the associated distortion along that interval. In Figure 5-1, we show the distribution of path fragments throughout the audio stream. This visualization illustrates how some time intervals in the audio match well to many other intervals, with up to 17 associated path fragments, while some time intervals do not have any matches at all. Since these fragments serve to link regions in time that are acoustically similar to one another, a natural question is to ask whether they can

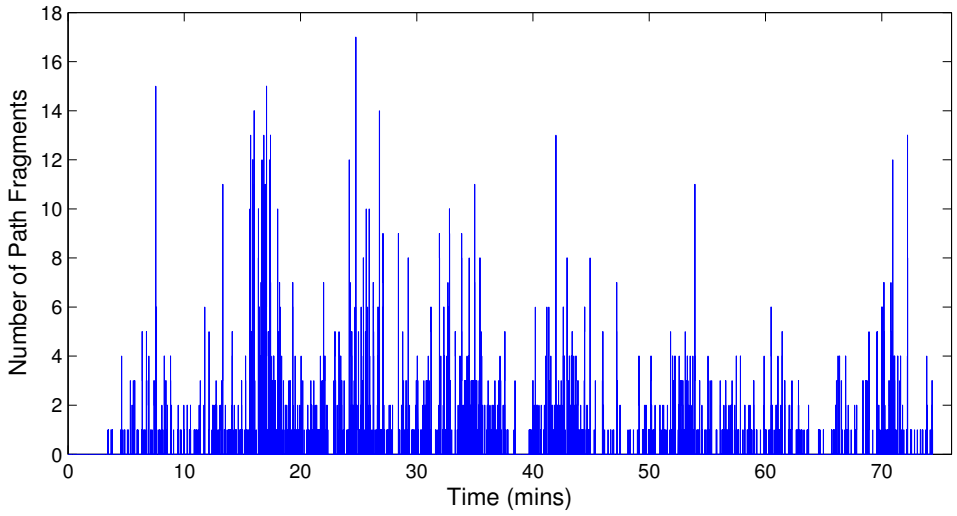


Figure 5-1: Distribution of path fragments through the time line of the Friedman lecture.

be used to build clusters of similar sounding speech segments with a common underlying lexical identity.

Our approach to this problem is cast in a graph theoretical framework, which represents the audio stream as an abstract adjacency graph, G, consisting of a set of nodes, V, and a set of edges, E. In this graph, the nodes correspond to locations in time, and the edges, correspond to measures of similarity between those time indices. Given an appropriate choice of nodes and edges, graph clustering techniques can be applied to this abstract representation to group together the nodes in the graph that are closest to one another. Since graph clustering and partitioning algorithms are an active area of research [105, 31, 72, 81], a wide range of techniques can be applied to this stage of the problem.

An overview of the graph conversion process is shown in Figure 5-2. The time indices indicated in the audio stream are realized as nodes in the adjacency graph, while the alignment paths overlapping the time indices are realized as edges between the nodes. We use these alignment paths to derive edge weights by applying a simple linear transformation of the average path distortions, with the weight between two nodes being given by the following similarity score

$$w(n_i, n_j) = \mathcal{S}(\mathcal{P}(n_i, n_j)) = \frac{\theta - \mathcal{D}(\mathcal{P}(n_i, n_j))}{\theta}. \tag{5.1}$$

In this equation, w is the weight on the edge between nodes n_i and n_j, $\mathcal{P}(n_i, n_j)$ is the alignment path common to both nodes, $\mathcal{D}(\mathcal{P}(n_i, n_j))$ is the average distortion for that path, and θ is a threshold used to normalize the path distortions. The average distortion is used as opposed to the total distortion in order to normalize

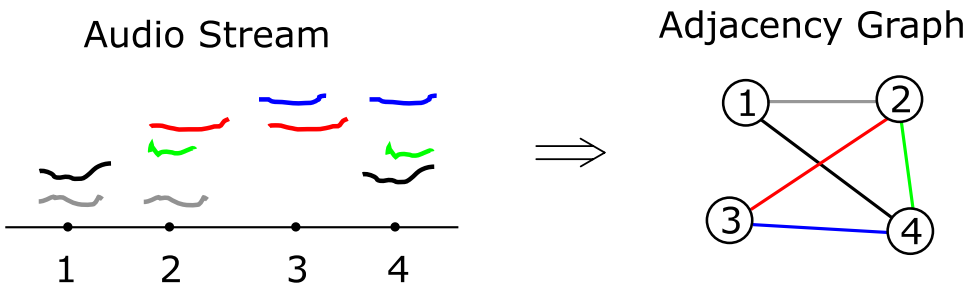


Figure 5-2: Production of an adjacency graph from alignment paths and extracted nodes. The audio stream is shown as a timeline, while the alignment paths are shown as pairs of colored lines at the same height above the timeline. Node relations are captured by the graph on the right, with edge weights given by the path similarities.

for path lengths when comparing paths with different durations. Paths with average distortion greater than θ are not included in the similarity computation. The distortion threshold chosen for all experiments in this chapter was 2.5, which is relatively inclusive considering the matching accuracy experiments from Chapter 4 . The resulting edge weights are closer to 1 between nodes with high similarity, and closer to zero (or non-existant) for nodes with low similarity.

5.1.1 Node Extraction

While it is relatively straightforward to see how alignment path fragments can be converted into graph edges given a set of time index nodes in the audio stream, it is less clear how these nodes can be extracted in the first place. In this section, we describe the node extraction procedure.

Recall that the input to the segmental DTW algorithm is not a single contiguous audio stream, but rather a set of utterances produced by segmenting the audio using silence detection. Our goal in node extraction is to determine a set of discrete time indices within these utterances that are representative of their surrounding time interval. This is accomplished by using information about the alignment paths that populate a particular utterance.

Consider the examples shown in Figures 5-3 and 5-4. In each of these examples, there are a number of alignment paths distributed throughout the utterance with different average path distortions. The distribution of alignment paths is such that some time indices are covered by many more paths than others – and are therefore similar to more time indices in other utterances. These heavily covered time indices are typically located *within* the words and phrases that are matched via multiple alignment paths.

We can use the alignment paths to form a *similarity* profile by summing the

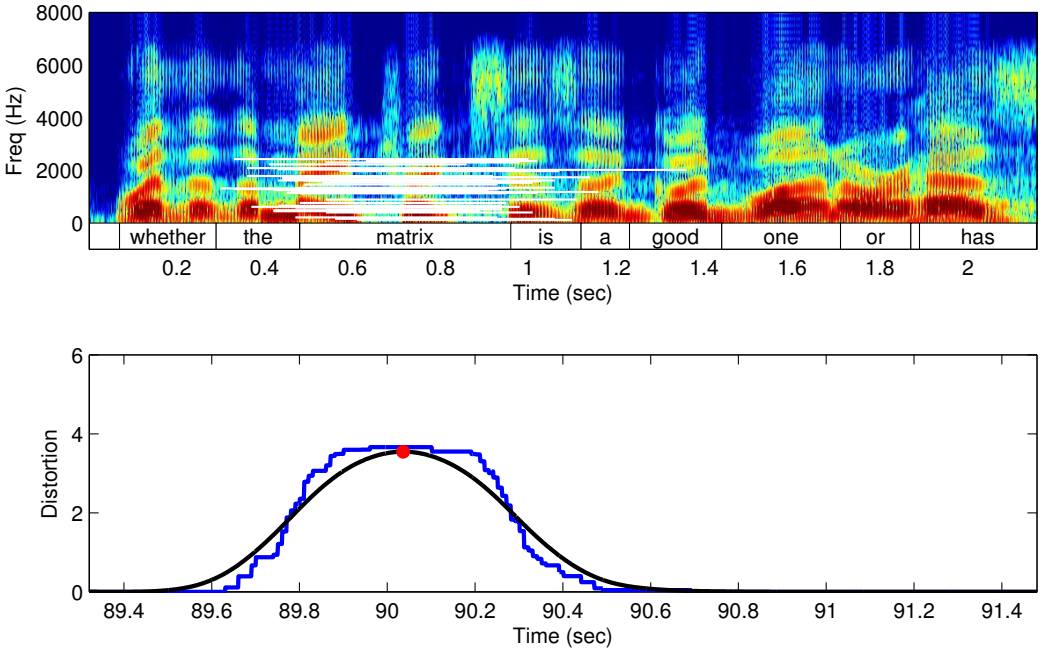


Figure 5-3: Top - An utterance from the Algebra lecture with the time regions from its associated path fragments shown in white. Paths are ordered from bottom to top in increasing order of distortion. Bottom - Similarity profile is shown in blue with the smoothed version shown in black. The extracted time index is shown as a red dot.

similarity scores of Equation 5.1 over time. That is, the similarity score at time t, is given by

$$\mathrm{S(t)} = \sum_{\mathcal{P},\mathrm{t}\in\mathcal{P}} \mathcal{S}(\mathcal{P}) \tag{5.2}$$

In this equation, $\mathcal{P}$ are the paths that overlap t and $\mathcal{S}(\mathcal{P})$ is the similarity value for $\mathcal{P}$ given by equation 5.1.

After smoothing the similarity profile with a triangular averaging window, we take the peaks from the resulting smoothed profile and use those time indices as the nodes in our adjacency graph. Because our extraction procedure finds locations with locally maximized similarity within the utterance, the resulting time indices demarcate locations that are more likely to bear resemblance to other locations in the audio stream.

The reasoning behind this procedure can be understood by noting that only some portions of the audio stream will have high similarity (i.e. low distortion) to other portions. By focusing on the peaks of the aggregated similarity profile, we restrict ourselves to finding those locations that are most similar to other locations. Since every alignment path covers only a portion of an utterance,

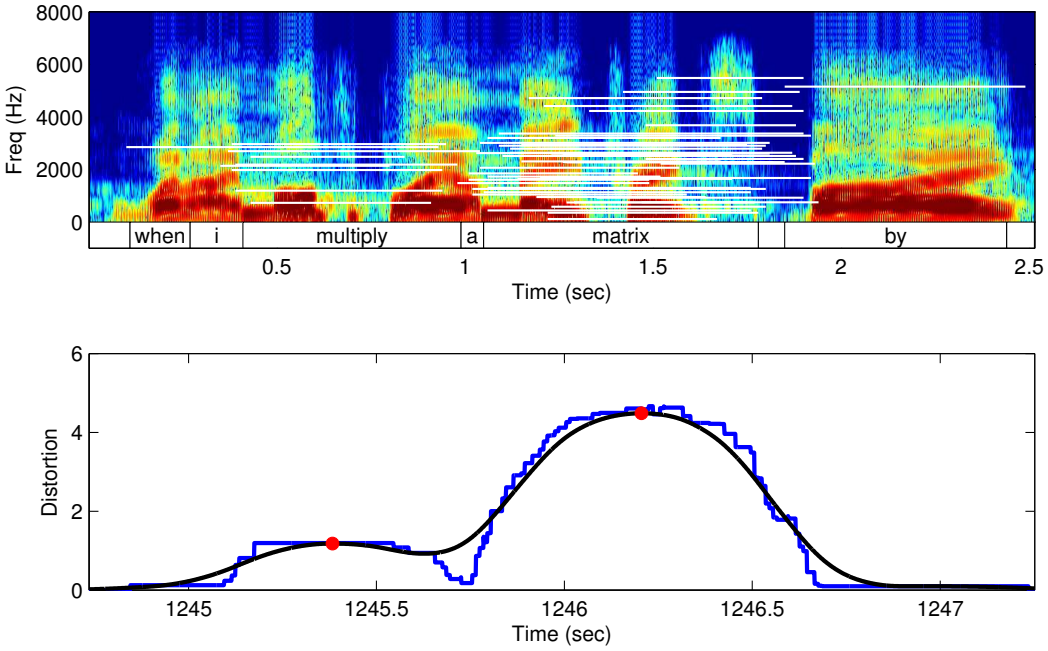


Figure 5-4: Top - An utterance from the Algebra lecture with the time regions from its associated path fragments shown in white. Paths are ordered from bottom to top in increasing order of distortion. Bottom - Similarity profile is shown in blue with the smoothed version shown in black. The extracted time index is shown as a red dot.

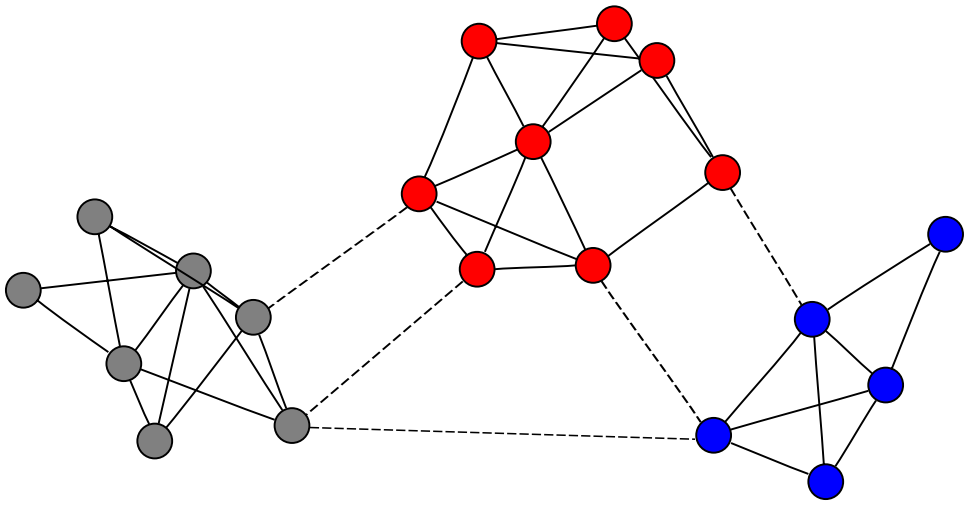


Figure 5-5: Example of graph clustering output. Nodes are colored according to cluster membership. Dashed lines indicate intercluster edges.

the similarity profile will fluctuate over time. This causes each utterance to separate naturally into multiple nodes corresponding to distinct patterns that can be joined together via their common alignment paths.

5.1.2 Graph Clustering

Once an adjacency graph has been generated for the audio stream using the extracted nodes and path fragment edges, the challenge of finding clusters in the graph remains. In an adjacency graph, a good clustering is one where nodes in one cluster are more densely connected to each other than they are to nodes in another cluster. The clustered adjacency graph in Figure 5-5 illustrates this concept. A naive approach to this problem is to simply threshold the edge weights and use the groups of connected components that remain as clusters. Though conceptually simple, this approach is prone to accidental merging if even a single edge with high weight exists between two clusters that should be separated. In contrast to simple edge thresholding, a number of more sophisticated algorithms for automatic graph clustering have been proposed by researchers in other fields [120, 80, 116]. For some applications, such as task scheduling for parallel computing, the clustering problem is cast as a partitioning task, where the number and size of desired clusters is known and the objective is to find the optimal set of clusters with those criteria in mind. For other applications, such as detecting community structure in social and biological networks, the number and size of clusters is typically unknown, and the goal is to discover communities and groups from the relationships between individuals.

In our work, the clustering paradigm aligns more closely with the latter example,

as we are attempting to discover groups of segments corresponding to the same underlying lexical entity, and not partition the audio stream into a set of clusters with uniform size. Since a detailed treatment of the graph clustering problem is outside the scope and intent of this thesis, we focus on an efficient, bottom-up clustering algorithm for finding community structure in networks proposed by Newman [79]. The Newman algorithm begins with all edges removed and each node in its own group, then merges groups together in a greedy fashion by adding edges back to the graph in the order that maximizes a modularity measure, Q, which is given by

$$Q = \sum_i (e_{ii} - a_i^2) \tag{5.3}$$

where e_{ij} is the fraction of edges in the original network that connect vertices in group i to those in group j, and $a_i = \sum_j e_{ij}$. More informally, Q is the fraction of edges that fall within groups, minus the expected value of the same quantity if edges fall at random without regard for the community structure of the graph. The value of Q ranges between 0 and 1, with 0 being the expected modularity of a clustering where intercluster edges occured about as frequently as intracluster edges, and higher scores indicating more favorable clusterings of the graph. The advantages of this particular algorithm are threefold. First, it easily allows us to incorporate edge weight information in the clustering process by considering weights as fractional edges in computing edge counts. Second, it is extremely fast, operating in $O((V + E)V)$ time in the worst case. Finally, the modularity criterion offers a data-driven measure for determining the number of clusters to be detected from a particular graph.

Because our goal is to separate the graph into groups joining nodes sharing the same word(s), multiple groups containing the same word are more desirable than fewer groups containing many different words. We therefore associate a higher cost with the action of mistakenly joining two unlike groups than that of mistakenly leaving two like groups unmerged. This observation leads us to choose a conservative stopping point for the clustering algorithm at 80% of peak modularity.

5.1.3 Nodes to Intervals

Recall from Section 5.1.1 that the nodes in the adjacency graph represent not time intervals in the original audio stream, but time indices. For the purposes of clustering, this time index abstraction may be adequate for representing nodes, but we will, at times, require associating a time interval corresponding to that node. One situation where we need a time interval rather than the time index corresponding to the node is for determining how to transcribe a node. As can be seen from the examples in Figures 5-3 and 5-4, the alignment paths

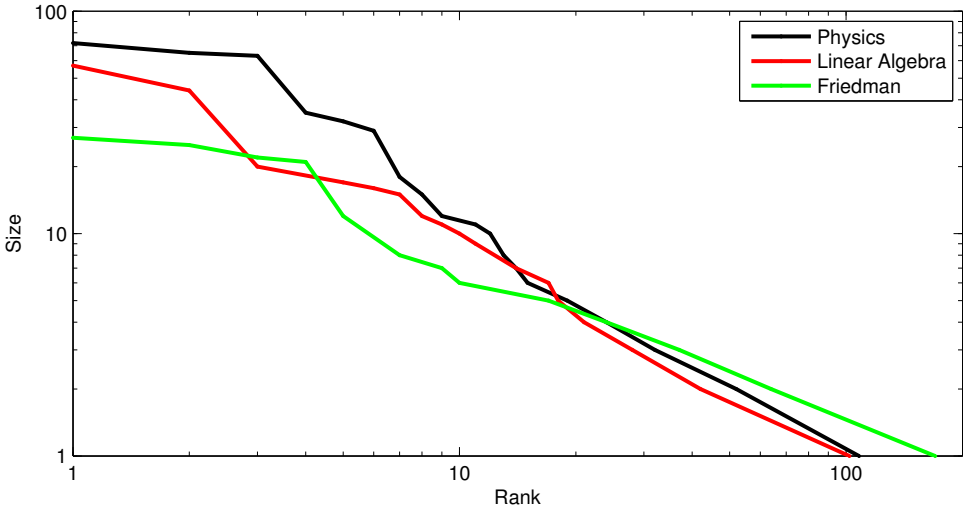


Figure 5-6: Log-log plot of cluster size versus size rank for the Physics, Algebra, and Friedman lectures.

overlapping a particular node rarely agree on starting and ending times for their respective time intervals. We assign a time interval to a node by computing the average start and end times for all the alignment paths for edges occurring within the cluster to which that node belongs.

5.2 Cluster Analysis

We processed the six lectures listed in Table 4.3 using the segmental DTW algorithm and generated clusters for each. The distribution of cluster sizes is shown in Figure 5-6 for the Physics, Algebra, and Friedman lecture. We observe that the cluster sizes exhibit an exponentially decaying trend characteristic of Zipf's Law, with most of the clusters being smaller in size, and larger clusters being few in number. The trend seen in this figure is consistent across all of the other lectures processed in this section.

A more detailed view of the clusters with at least 3 members generated for the Friedman lecture is shown in Table 5.1, with a graphical representation of these clusters is shown in Figure 5-7. In this table, the clusters are listed first in decreasing order of size, denoted by $|\mathcal{C}|$, then by decreasing order of density, $D(\mathcal{C})$, which is a measure of the "interconnectedness" of each cluster. The density of a cluster is given by,

$$D(\mathcal{C}) = \binom{|\mathcal{C}|}{2}^{-1} \sum_{n_1, n_2 \in \mathcal{C}} w(n_1, n_2). \tag{5.4}$$

The quantity in the above equation is the fraction of edges observed in the cluster out of all possible edges that could exist between cluster nodes. Higher densities indicate greater agreement between nodes. Table 5.1 also includes a purity score for each cluster. The purity score is a measure of how accurately the clustering algorithm is able to group together like speech nodes, and is determined by calculating the percentage of nodes that agree with the lexical identity of the cluster. The cluster identity, in turn, is derived by looking at the underlying reference transcription for each node and choosing the the word or phrase that appears most frequently in the nodes of that particular cluster. Clusters with no majority word or phrase (such as those matching subword speech segments), are labeled as '-'. Although Table 5.1 lists the clusters only for the Friedman lecture, similar cluster tables for the other lectures processed for this chapter are included in Appendix A.

Example clusters

Some examples of specific clusters with high and low purity are shown in Figures 5-8 and 5-9, respectively. Cluster 27 in Figure 5-8 is an example of a high density cluster, with each node connecting to each other node, and the underlying transcriptions confirm that each node corresponds to the same recurring phrase. The other two clusters in Figure 5-8, while not displaying the same degree of interconnectedness, nevertheless all consist of nodes with similar transcriptions. One interesting property of these clusters is the high degree of temporal locality displayed by their constituent nodes. With the exception of node 587, most of the other nodes occur within 5 minutes of the other nodes in their respective clusters. This locality may be indicative of transient topics in the lecture which require the usage of terms that are only sporadically used. In the case of cluster 27, these 4 instances of "search engine optimize-" were the only instances where they were spoken in the lecture.

In contrast to these three clusters, each with purities of 100%, cluster 4, shown in Figure 5-9 has a purity of only 38%, with only 8 of the 21 nodes containing the majority word "imagination". The pattern of connections in this cluster is helpful in diagnosing failure modes of the clustering algorithm. Visual inspection reveals that the cluster is actually an accidental merging of several word entities, with connections between the entities being facilitated by suffixes and multi-word phrases. In the lower left portion of the graph, the 8 nodes containing the word "imagination" are joined to the sub-cluster representing the phrase "nine eleven" by the node with the transcription "imagination of nine eleven". In the lower right portion of the graph, several of the "imagination" nodes are connected to nodes with a variety of underlying transcriptions, but the subword units "sh ax n" or "sh iyn" in common. This example cluster illustrates how 'chaining' errors can arise as a result of path fragments with different lengths covering the same node, and why the resulting 'chained' clusters typically have lower density

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**
1	27	0.017	applications	29.6	33	4	0.050	more people	100.0
2	25	0.020	collaboration	88.0	34	4	0.047	-	0.0
3	22	0.043	globalization	100.0	35	4	0.046	supply chain	100.0
4	21	0.023	imagination	38.1	36	4	0.040	-	0.0
5	12	0.032	platform	100.0	37	3	0.203	plug and play	100.0
6	12	0.023	flattener	75.0	38	3	0.201	knowledge and	100.0
7	8	0.053	fiberoptic cable	50.0				work	
8	8	0.030	to ups	37.5	39	3	0.142	china	100.0
9	7	0.033	southwest	100.0	40	3	0.139	the berlin wall	100.0
			airlines		41	3	0.133	multinationals	100.0
10	6	0.102	(n)ever before	100.0	42	3	0.114	huge	100.0
11	6	0.098	informing	100.0	43	3	0.110	system	100.0
12	6	0.077	the history of	83.3	44	3	0.107	fourteen ninety	100.0
13	6	0.073	flat world	100.0				two	
14	6	0.065	outsourcing	100.0	45	3	0.106	solar powered	66.7
15	6	0.036	-	0.0	46	3	0.102	japanese	100.0
16	6	0.030	the beginning	33.3	47	3	0.101	internet	100.0
17	5	0.135	globalizing	100.0	48	3	0.093	-	0.0
18	5	0.111	open source	100.0	49	3	0.090	imported	100.0
19	5	0.106	individuals	100.0	50	3	0.087	um	100.0
20	5	0.080	two thousand	100.0	51	3	0.083	quiet crisis	100.0
21	5	0.076	reservation	100.0	52	3	0.081	discover	100.0
22	5	0.054	horizontal	100.0	53	3	0.075	-	0.0
23	5	0.043	governance	40.0	54	3	0.071	standards	66.7
24	4	0.169	search engine	100.0	55	3	0.069	governance	66.7
			optimizer		56	3	0.066	-	0.0
25	4	0.169	toshiba laptop	100.0	57	3	0.065	business	100.0
26	4	0.146	work together	100.0				processes	
27	4	0.105	inflection point	75.0	58	3	0.064	software	100.0
28	4	0.070	haven connecticut	50.0	59	3	0.064	the night before	100.0
29	4	0.066	flatten the world	100.0	60	3	0.064	the world is flat	100.0
30	4	0.059	economic playing	100.0	61	3	0.059	u p s	66.7
			field		62	3	0.053	productivity	100.0
31	4	0.053	ten percent	50.0				boost	
32	4	0.053	the world	100.0	63	3	0.050	-	100.0

Table 5.1: Information for the 63 clusters with at least 3 members generated for the Friedman lecture. Clusters are ordered first by size, then in decreasing order of density.

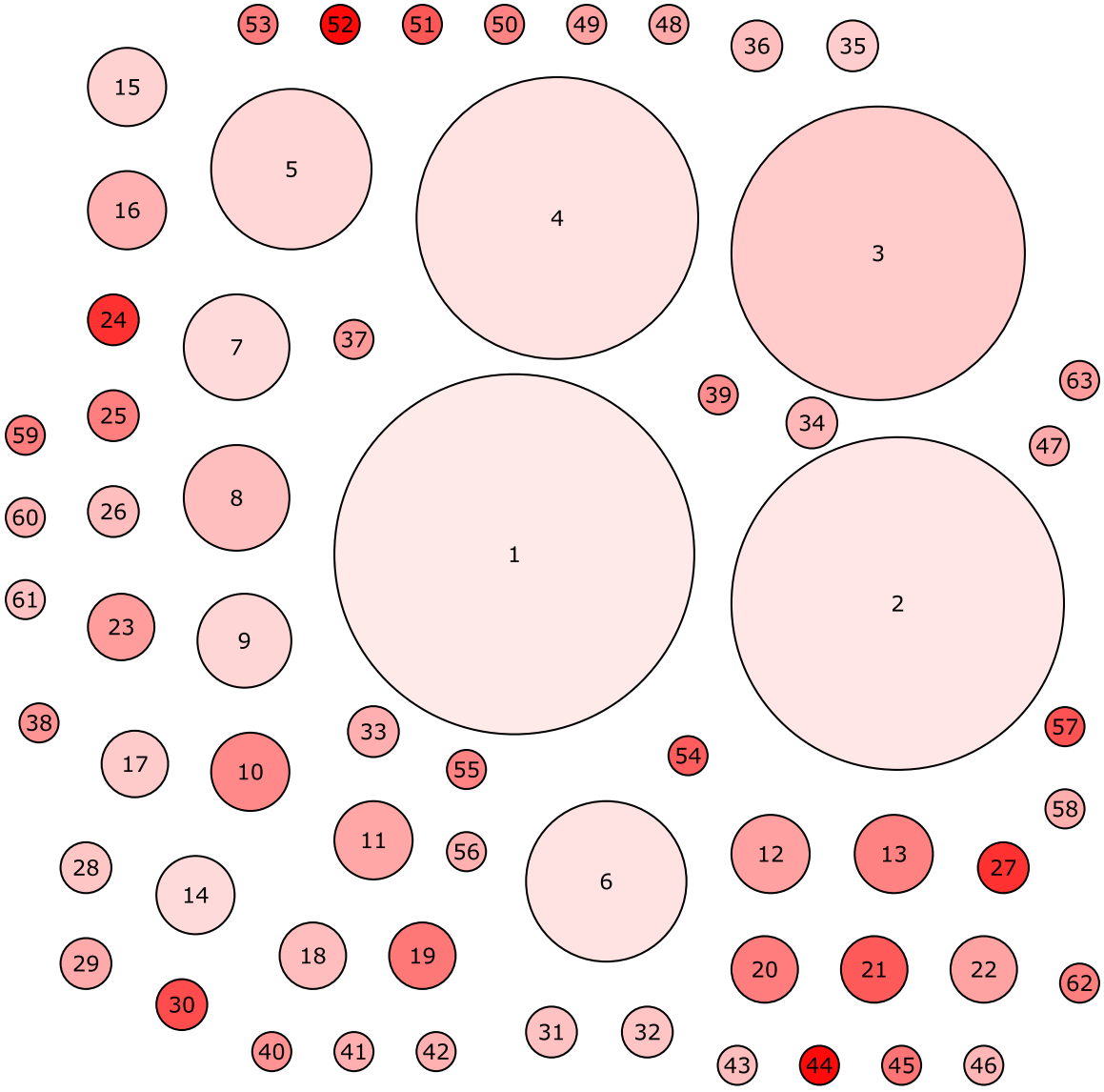


Figure 5-7: Graphical representation of the clusters found in the Thomas Friedman lecture. Only clusters with at least 3 members are shown here, with labels applied in decreasing order of size. The radius of each circle is proportional to the cluster size, and the color intensity is proportional to the edge density.

Lecture	# Clusters	Average Size	Average Purity	# Single Word	#Multi Word
Friedman	63	5.59	79.63	25	31
ASR Lecture 2	92	7.36	86.13	44	44
ASR Lecture 6	87	10.05	93.22	47	40
ASR Lecture 19	63	10.32	91.85	33	29
Physics	51	10.39	89.46	31	20
Algebra	41	8.80	93.98	30	11

Table 5.2: Cluster statistics for all lectures processed in this chapter. Only clusters with at least 3 members are included in this table. The last two columns indicate how many of the generated clusters are associated with a single word identity or a multi-word phrase.

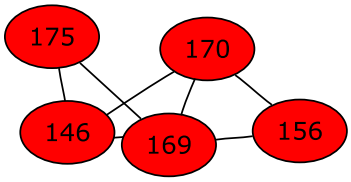
than clusters with higher purity. A more macroscopic view of example clusters extracted from the Physics lecture are shown in Figure 5-10.

Cluster statistics

Several interesting points can be noted regarding the clusters generated from the Friedman lecture. First, most clusters (56 of 63) have a word or phrase that can be considered to be the lexical identity of the cluster. Out of these clusters, over 73% of the clusters have a purity of 100%, which offers encouraging evidence that the segmental DTW measures and subsequent clustering procedure are able to correctly group recurring words together. As might be expected, the cluster density appears to be positively correlated to cluster purity, with an average purity of 87% among clusters with density greater than 0.05, and an average purity of 53% among clusters with density less than or equal to 0.05. We also observe that the clustering algorithm does not appear to discriminate between single words and multi-word phrases that are frequently spoken as a single entity, with more than half of the clusters (31 of 56) mapping to multi-word phrases.

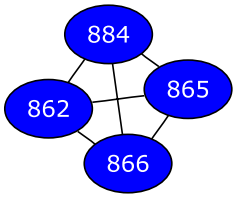
Overall cluster purity statistics for the five other academic lecture processed in this chapter are shown in Table 5.2. We found that across all six lectures, approximately 83% of the generated clusters had density greater than 0.05, and among these higher density clusters, the average purity was 92.2%. In contrast, the average purity across all of the lower density clusters was only 72.6%. These statistics indicate that the observations noted in the previous paragraph appear to transfer to the other lectures. Some notable differences between the Friedman lecture and the academic lectures are the larger average cluster size, and higher overall purity across the clusters in general. The larger size of some clusters can be attributed to the more focused nature of the academic lecture vocabulary, while the higher purity may be a result of differences in speaking style.

Node Index	Time Index	Transcription
146	16:03	countries globalizing
156	16:44	companies globalizing
169	17:27	countries globalizing
170	17:30	companies globalizing
175	17:40	globalizing



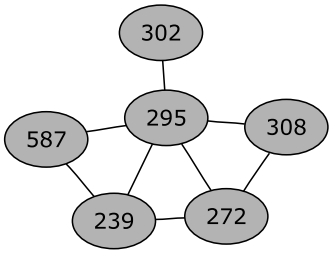
(a) Cluster 21

Node Index	Time Index	Transcription
862	1:06:14	a search engine optimizer
865	1:06:22	be a search engine optimizer
866	1:06:32	search engine optimizing
884	1:08:45	a search engine optimizing



(b) Cluster 27

Node Index	Time Index	Transcription
239	21:08	than ever before
272	23:27	than ever before in
295	24:41	ever before
302	25:08	never before
308	25:15	ever before
587	42:34	than ever before



(c) Cluster 13

Figure 5-8: Detailed view of clusters 17, 24, and 10, including the node indices, transcriptions, and locations in the audio stream.

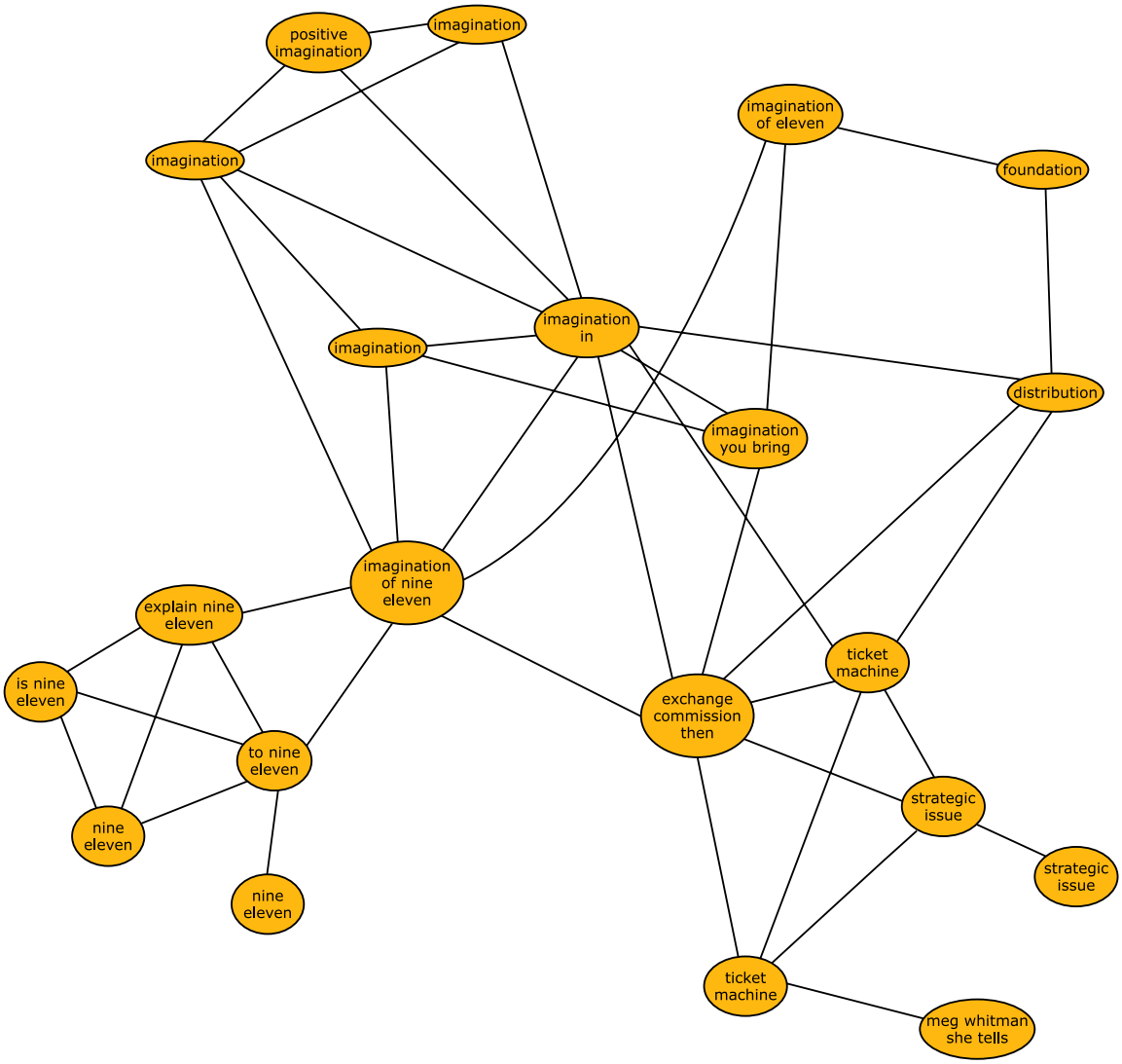


Figure 5-9: Detailed view of cluster 4 from the Friedman lecture illustrating the topology of an impure cluster.

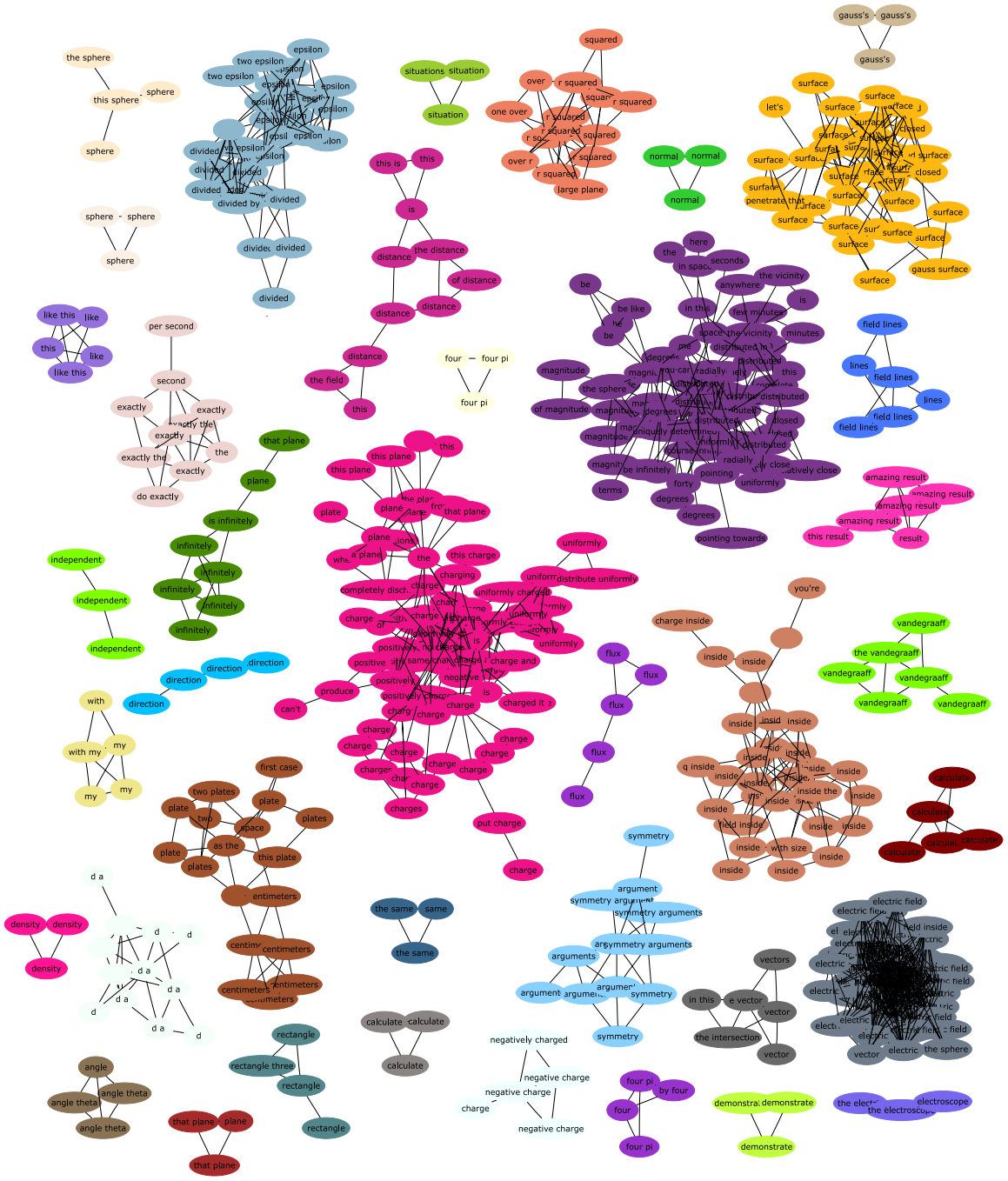


Figure 5-10: Selection of larger clusters generated from Physics lecture. Cluster nodes are labeled with the word(s) spanning the time index associated with the node.

5.2.1 Cluster Relevance

A cursory view of the cluster identities for each lecture indicates that many of the clusters correspond to words or phrases that are highly specific to the subject material of that particular lecture. For example, in the physics lecture, the words "charge", "electric", "surface", and "epsilon", all correspond to some of the larger clusters for the lecture. This phenomenon is somewhat expected, since relevant content words are likely to recur more often, and function words such as "the", "is", and "of", are of short duration and typically exhibit significant pronunciation variation as a result of coarticulation with adjacent words. One way of evaluating how well the clusters capture the subject content of a lecture is to consider the coverage of relevant words by the generated clusters.

Since there is no easy way of measuring word relevancy directly, for the purposes of our work, we use each word's term-frequency, inverse document-frequency (TFIDF) score as a proxy for its degree of relevance [103]. The TFIDF score is the frequency of the word within a document normalized by the frequency of the same word across multiple documents. Our rationale for using this score is that words with high frequency within the lecture, but low frequency in general usage are more likely to be specific to the subject content for that lecture. The word lists in Table 5.3 are the twenty most relevant words for each lecture ranked in decreasing order of their TFIDF score. Each list was generated as follows:

(1) First, words in the reference transcription were stemmed rudimentarily to merge pluralized nouns with their associated root nouns, and various verb tenses with their associated root verbs.

(2) Partial words, filled pauses, single letters and numbers, and contractions such as "you've" or "i'm" were removed from the reference transcription.

(3) Next, the remaining words in the lecture were ranked according to their term-frequency, inverse document frequency, where the document frequency was taken from the 2K most common words in the Brown corpus [34].

Referring back to the lecture descriptions in Table 2.1, the lists of words generated in Table 5.3 appear to be very relevant to the subject matter of each lecture, which qualitatively validates our use of the TFIDF measure. The words for each lecture in Table 5.3 are colored according to their cluster coverage, with words represented by a cluster colored in red. On average, 14.8 of the top 20 most relevant words are covered by a cluster generated by our procedure. This statistic offers encouraging evidence that the recurring acoustic patterns discovered by our approach are not only similar to each other (as shown by the high average purity), but also informative about the lexical content of the audio stream.

Friedman	Algebra	Physics	ASR L2	ASR L6	ASR L19
flat	matrix	electric	frequency	cluster	speaker
globalization	row	zero	vocal	distortion	adaptation
collaboration	zero	sphere	wave	data	model
india	pivot	charge	transform	algorithm	vector
era	equation	plate	fourier	metric	parameter
flattener	elimination	symmetry	vowel	vector	adapt
dollar	column	flux	speech	distance	technique
china	multiply	plane	cavity	speech	utterance
southwest	matrices	vector	signal	split	weight
argue	subtract	uniformly	tract	assign	likelihood
airline	minus	gauss	fold	quantization	estimate
thousand	step	field	sound	dimension	dependent
outsourcing	multiplication	angle	acoustic	train	independent
really	exchange	epsilon	window	iteration	data
platform	inverse	divided	characteristic	plot	recognize
huge	suppose	vandegraaff	function	coefficient	speech
create	plus	surface	source	mean	error
convergence	negative	distribute	velocity	pick	cluster
connect	substitution	inside	tongue	merge	mean
chapter	identity	sigma	noise	criterion	filter

Table 5.3: Twenty most relevant words for each lecture, listed in decreasing order of TFIDF score. Words occuring as part of a cluster for that lecture are colored in red.

5.3 Summary

This chapter has focused on the acquisition of lexical entities from the information produced by the segmental DTW algorithm. We demonstrated how to use alignment paths, which indicate pairwise similarity, to transform the audio stream into an abstract adjacency graph which can then be clustered using standard graph clustering techniques. As part of our evaluation, we showed that the clusters generated by our proposed procedure have both high purity and good coverage of terms that are relevant to the subject of the underlying lecture.

Chapter 6

Cluster Identification

The entire clustering process that we have described until this point has been completely unsupervised, requiring no labeled input data either for model training or for determining the parameters of a discriminative classifier. The clusters that we are able to extract from the audio stream have only two quantities associated with them: the cluster size, which is an indication of the relative frequency of that lexical entity in the lecture, and the cluster edge density, which is an indicator of how well the cluster nodes agree with each other. For some situations, such as direct audio summarization, this information, combined with the original audio, may be enough to improve user interaction with audio on its own. For other situations, such as vocabulary initialization for speech recognition, or speech to text summarization, the ability to automatically identify the word or words associated with a cluster is an additional requirement that is missing from our algorithm as described thus far.

In this chapter, we attempt to perform cluster identification in two ways by employing lightweight speech recognition techniques. The first method is straightforward, using an isolated word recognizer to attempt word identification directly from acoustics. The second method is a more decoupled approach, which represents cluster nodes and lexicon entries as sets of n-phone in order to perform

a baseform search. We describe each method and their accompanying results in the next two sections.

6.1 Isolated Word Recognition

The isolated word recognition (IWR) approach to cluster identification utilizes a unified search strategy by coupling the decoding from acoustics into phones within the lexicon search.

6.1.1 Method

For this recognition-based approach, we use the SUMMIT speech recognizer, as described in Chapter 2, with a recognition grammar designed for isolated word recognition in the context of continuous speech. The topology of this grammar is shown in Figure 6-1. The main component of this word network is the 150K word lexicon, which occupies the body of the network and serves to identify the word in the input time interval. The partial word loops attached to the initial and final states of the network are included to allow for absorption of the extra phones that occur before and after the word of interest. The inclusion of these extra phones results from imprecise placement of the start and end boundaries during time interval estimation. The partial word model itself is an ergodic phone loop with transition probabilities that are trained using a large baseform pronunciation dictionary [9].

In most, if not all, speech recognition systems, utterances are recognized locally, with no enforcement of consistency between multiple realizations of the same word. The recognition approach we describe here attempts to enforce these constraints by incorporating the auxiliary information provided by the clustering algorithm. Under the assumption that multiple speech segments correspond to the same underlying word, recognition can be performed in a way that attempts to choose the word with highest consensus among all input speech segments.

Our cluster identification procedure is as follows:

(1) Peform isolated word recognition on the time intervals for each node of the cluster using the word network shown in Figure 6-1. For each node, retain the N-best list of alternatives produced by the recognizer. In our experiments, we used N=10.

(2) Aggregate the N-best lists for all the nodes together into a single large list. Sort the list by score and choose the top 10 candidates to rescore in the next step.

(3) Rescore each candidate remaining from the previous step by performing

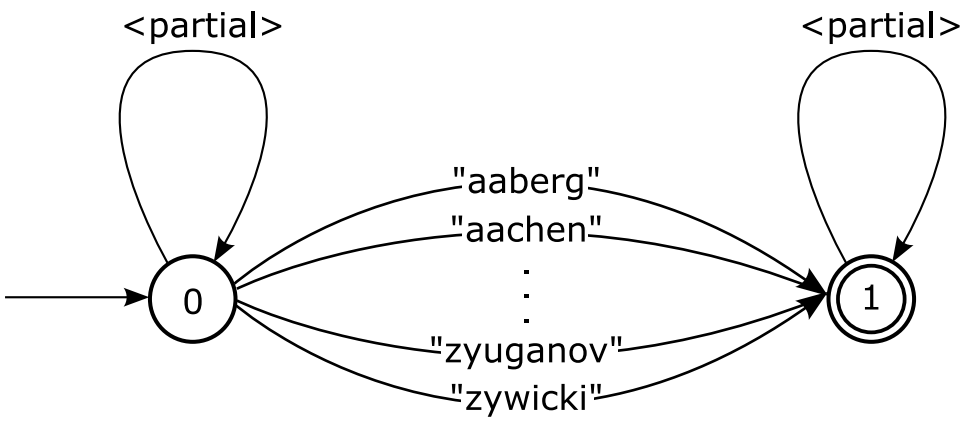


Figure 6-1: Finite state transducer structure for the isolated word recognizer used during cluster identification.

a forced alignment of each node with that candidate, then taking the average of these forced alignment scores.

(4) Rerank the candidates according to the new scores and use the top scoring candidate as the hypothesis for the cluster.

The rationale behind the approach described above comes from the realization that while word recognition for any individual node may be errorful, the multiple instances of the underlying word across the other nodes in the cluster are likely to improve recognition for the entire cluster. The first two steps perform this aggregation process to determine a shortlist of candidate words, and the final two steps generate a word score for each candidate on the nodes that did not hypothesize the candidate on the first pass.

6.1.2 IWR Identification Results

In Table 6.1 we show the cluster identification results using the IWR method for the Friedman lecture, with the top hypothesis shown for each cluster. For the clusters where the correct word occurs in the N-best list, we include the rank of the correct word. Words are considered to be identified correctly based on their roots as long as the morphological change is rather minor. For example, "power" is considered to match to "powered" and "powering", but not to "powerful". Likewise, "govern" matches to "governs", but not to "government" or "governance". Clusters with multi-word phrase references are considered to match any word occuring in the phrase. For example, the phrase "southwest airlines" is considered to match either "southwest" or "airlines" if they are hypothesized in the N-best list for that cluster.

Overall identification accuracy for the single word clusters was high considering

the paucity of lexical information provided to the recognition system. Of the 25 single word clusters (we do not count the filled pause, [um], as a word in these results), 22 contained the correct word in the 10-best list, with all of the correct words having a rank of 7 or better. In 17 of these cases, the correct word was the top hypothesis.

The results in Table 6.1 indicate that the major weakness of this identification algorithm is an inability to robustly identify words occuring in multi-word clusters. Only 14 of the 30 multi-word clusters had reference words appearing in their 10-best list, with only two of the clusters being identified correctly with the top hypothesis. In designing our recognition network, we had conjectured that the partial word model would have been able to identify portions of the multi-word phrases by allowing parts of the utterance to be absorbed by the partial word model. However, a survey of the top hypotheses for these clusters indicates that this is not the case. Although for cluster 7, the identification procedure was correctly able to identify "fiberoptic" while ignoring the second part of the utterance, the majority of cases produced hypotheses that attempted to cover two or more of the underlying reference words. This phenomenon is particularly evident in clusters 13, 24, and 25, where the phrases "flat world", "search engine optimize", and "toshiba laptop" are misidentified as "flatworm", "surcharging", and "shibboleth", respectively.

Cluster identification statistics for the three ASR lectures, which are shown in Table 6.2, indicate that the trends observed above carry over to the other lectures as well (the physics and algebra lectures were not included in the identification experiments because the acoustic model training data included data from these speakers). More detailed cluster identification tables for these other lectures are included in Appendix B. Although the distribution of clusters into single word and multi-word clusters is relatively balanced, there is significant disparity in identification accuracy among the two types of clusters. For single word clusters, approximately 87% of the cluster N-best lists contained the correct word identity for the cluster, with the average rank of the correct word being close to 1.9. In contrast, only 32% of the multi-word cluster N-best lists contained one of the words associated with the cluster identity. The average rank of the correct words for these clusters was also worse than their single word counterparts.

The disparity in performance between the two cluster types is not entirely surprising given the design of our recognition grammar, which is specialized for isolated word recognition. In Chapter 8, we discuss possible ways to improve identifcation performance for multi-word clusters. The high level of accuracy obtained for single word clusters, however, offers encouraging evidence that providing non-local constraints about which speech segments correspond to the same word can generate reliable recognition results even in the absence of local context and language model information.

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	Reference	Hypothesis	Rank	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	Reference	Hypothesis	Rank
1	27	applications	applications	1	33	4	more people	orky	3
2	25	collaboration	collaboration	1	34	4	-	replaced	-
3	22	globalization	mobilization	7	35	4	supply chain	plighting	10
4	21	imagination	adulation	2	36	4	-	doland	-
5	12	platform	platform	1	37	3	plug and play	contemplate	-
6	12	flattener	flattener	1	38	3	knowledge and	witwer	-
7	8	fiber optic cable	fiberoptic	1			work		
8	8	to ups	completes	-	39	3	china	windchime	-
9	7	southwest	swissair	6	40	3	the berlin wall	brolin	-
		airlines			41	3	multinationals	multinationals	1
10	6	(n)ever before	vermiform	-	42	3	huge	huger	2
11	6	informing	informing	1	43	3	system	insistent	-
12	6	the history of	industria	-	44	3	fourteen ninety	fortyniners	7
13	6	flat world	flatworm	-			two		
14	6	outsourcing	outsourcing	1	45	3	solar powered	sorkow	-
15	6	-	contrite	-	46	3	japanese	japanese	1
16	6	the beginning	tibetan	-	47	3	internet	internet	1
17	5	globalizing	globalizing	1	48	3	-	dissipative	-
18	5	open source	insourcing	-	49	3	imported	imported	1
19	5	individuals	individuals	1	50	3	um	on	5
20	5	two thousand	tootles	5	51	3	quiet crisis	winepresses	-
21	5	reservation	reservations	1	52	3	discover	discovered	1
22	5	horizontal	horizontal	1	53	3	-	datas	-
23	5	governance	govs	3	54	3	standards	nonstandard	-
24	4	search engine	surcharging	-	55	3	governance	govs	6
		optimizer			56	3	-	traduced	-
25	4	toshiba laptop	shibboleth	-	57	3	business	misinterprets	5
26	4	work together	workaday	3			processes		
27	4	inflection point	flexion	5	58	3	software	software	1
28	4	haven connecticut	quickset	-	59	3	the night before	pipeful	3
29	4	flatten the world	flatworm	6	60	3	the world is flat	worlders	2
30	4	economic playing	aplenty	-	61	3	u p s	commuted	-
		field			62	3	productivity	productivity	1
31	4	ten percent	cumbersome	-			boost		
32	4	the world	world	1	63	3	-	tactical	-

Table 6.1: Cluster identification results using the IWR approach for the 63 clusters with at least 3 members in the Friedman lecture. The reference is majority lexical identity of the cluster, the hypothesis is the top identification result, and rank is the rank of the correct word in the N-best list, if present.

Lecture	# Single	% in Top 10	Average Rank	# Multi	% in Top 10	Average Rank
Friedman	25	88.00	1.86	31	45.16	4.14
ASR Lecture 2	44	75.00	1.85	44	20.45	3.44
ASR Lecture 6	47	95.74	1.6	40	35.00	3.29
ASR Lecture 19	33	90.91	2.50	29	34.48	3.10
Overall	**149**	**87.2**	**1.93**	**144**	**31.9**	**3.55**

Table 6.2: Cluster identification statistics using the IWR approach for the Friedman lecture and three ASR lectures. Identification statistics are separately computed for clusters corresponding to single word phrases and those corresponding to multi-word phrases.

6.2 Decoupled Baseform Search

The decoupled baseform search (DBS) technique is an alternative approach to the identification strategy proposed in the previous section, and separates the identification process into two stages: phone transcription and baseform search.

6.2.1 Method

The procedure we use to assign words to the clusters generated from the previous section is relatively straightforward. For a given cluster, $\mathcal{C}$, a phonetic recognizer is used to transcribe the interval underlying each node and convert it into a set of n-phones, as shown in Figure 6-2. Likewise, the pronunciations for all words in a large baseform dictionary (150K words), $\mathcal{W}$, are converted into sets of n-phones. Our implementation of this technique uses n-phone sizes with n = 4. This process reduces each node, $\mathbf{n}_i$, and each word, $\mathbf{w}_j$, into sets of n-phone sequences. By comparing the similarity of the words in the dictionary to the nodes in $\mathcal{C}$, the most likely candidate word common to the cluster can be found. The hypothesized cluster identity is given by,

$$\mathbf{w}^* = \arg\max_{\mathbf{w}_j \in \mathcal{W}} \frac{2}{|\mathcal{C}|} \sum_{\mathbf{n}_i \in \mathcal{C}} \frac{|\mathbf{n}_i \cap \mathbf{w}_j|}{|\mathbf{n}_i| + |\mathbf{w}_j|}. \tag{6.1}$$

In this equation, we use the normalized intersection between the sets $\mathbf{n}_i$ and $\mathbf{w}_j$ as a measure of similarity and aggregate this over all nodes in the cluster for each word. We include the $|\mathbf{n}_i| + |\mathbf{w}_i|$ factor in the denominator to normalize for the size of $\mathbf{n}_i$ and $\mathbf{w}_i$, so that longer/shorter words are not favored due to their length. The factor of 2 in the numerator is included so that the overall score ranges between 0 and 1. Using this similarity score, we can generate an N-best list of word candidates for each cluster. As in the IWR approach, the baseform search method combines information for all of the cluster nodes in reaching an

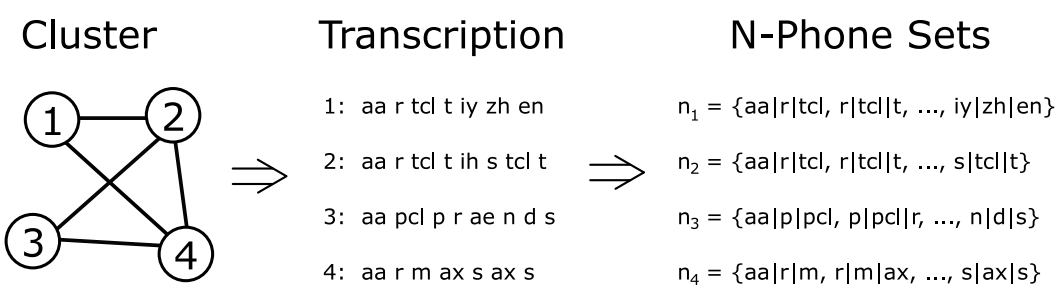


Figure 6-2: Conversion of cluster nodes into groups of n-phones. The intervals under the nodes are phonetically transcribed, then separated into sets of n-phone sequences. In this example, n = 3.

identification decision.

6.2.2 DBS Identification Results

In Table 6.3, we list identification results for the DBS method on the Friedman lecture. Tables of results for the other lectures processed in this chapter are included in Appendix C. We observe some of the same trends from IWR identification results in these results. Single word clusters appear to be identified reliably, with the correct label being found in the top 10 hypotheses of 84% of these clusters. Of the misidentified clusters corresponding to multi-word clusters, many of the problematic hypotheses span multiple words, such as "kemper" for "ten percent", and "halogen" for "knowledge and work".

Unlike the IWR approach, however, the DBS results do not have as much disparity in identification accuracy between multi-word clusters and single word clusters. In Table 6.4, we list identification statistics for the four lectures processed in this chapter. Overall, the percentage of clusters with the correct word occuring in the N-best list is similar for both multi and single word clusters, at 64.4% and 62.5%, respectively. This statistic, along with the average ranks of the correct cluster words, indicates that the DBS approach performs equally well for both types of clusters. Compared to the IWR approach, the N-best identification accuracy of the DBS method on single word clusters is lower (64.4% vs. 87.2%), but the multi-word cluster identification accuracy is almost double (62.5% vs. 31.9%). Moreover, among the multi-word clusters with the correct word occurring in the top 10, the average rank is improved from 3.55 to 1.98.

The reason for the improved performance of the DBS method on multi-word clusters is likely due to the use of n-phone sets as matching elements rather than entire phone sequences. This approach essentially uses short-range similarities between cluster nodes and baseform entries during the baseform search, which means that the DBS method is more likely to find partial matches for the reference labels rather than attempting to find matches that account for as

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	Reference	Hypothesis	Rank	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	Reference	Hypothesis	Rank
1	27	applications	allocations	2	33	4	more people	people	1
2	25	collaboration	collaboration	1	34	4	-	replaced	-
3	22	globalization	mobilization	3	35	4	supply chain	supply	1
4	21	imagination	imagination	1	36	4	-	doing	-
5	12	platform	platform	1	37	3	plug and play	imply	-
6	12	flattener	flatt	-	38	3	knowledge and	halogen	3
7	8	fiber optic cable	optic	1			work		
8	8	to ups	morcom	-	39	3	china	windchime	6
9	7	southwest	selfesteem	3	40	3	the berlin wall	burrill	-
		airlines			41	3	multinationals	multinationals	1
10	6	(n)ever before	before	1	42	3	huge	huger	4
11	6	informing	informing	1	43	3	system	system	1
12	6	the history of	history	1	44	3	fourteen ninety	fourteen	1
13	6	flat world	world	1			two		
14	6	outsourcing	outsourced	1	45	3	solar powered	powered	1
15	6	-	japan	-	46	3	japanese	japanese	1
16	6	the beginning	begin	1	47	3	internet	internet	1
17	5	globalizing	globalized	1	48	3	-	anticipate	-
18	5	open source	insource	3	49	3	imported	imported	1
19	5	individuals	individual's	1	50	3	um	nolen	-
20	5	two thousand	tooth	5	51	3	quiet crisis	quite	-
21	5	reservation	reservation	1	52	3	discover	rediscovery	9
22	5	horizontal	horizontal	1	53	3	-	skuas	-
23	5	governance	governance	1	54	3	standards	standard's	1
24	4	search engine	demise	-	55	3	governance	nongovernment	-
		optimizer			56	3	-	sustain	-
25	4	toshiba laptop	apt	-	57	3	business	proske	-
26	4	work together	worked	1			processes		
27	4	inflection point	flexion	4	58	3	software	-	-
28	4	haven connecticut	which	-	59	3	the night before	perforce	-
29	4	flatten the world	flatten	1	60	3	the world is flat	world	1
30	4	economic playing	play's	1	61	3	u p s	tube	-
		field			62	3	productivity	springtime	-
31	4	ten percent	kemper	2			boost		
32	4	the world	world	1	63	3	-	stecchini	-

Table 6.3: Cluster identification results using the DBS approach for the 63 clusters with at least 3 members in the Friedman lecture. The reference is majority lexical identity of the cluster, the hypothesis is the top identification result, and rank is the rank of the correct word in the N-best list, if present.

Lecture	# Single	% in Top 10	Average Rank	# Multi	% in Top 10	Average Rank
Friedman	25	84.00	1.90	31	64.52	1.70
ASR Lecture 2	44	40.91	1.78	44	54.55	2.00
ASR Lecture 6	47	76.60	1.44	40	67.50	2.33
ASR Lecture 19	33	63.64	1.71	29	65.52	1.79
Overall	**149**	**64.4**	**1.68**	**144**	**62.5**	**1.98**

Table 6.4: Cluster identification statistics for the Friedman lecture and three ASR lectures using the DBS identification aproach. Identification statistics are computed separately for clusters corresponding to single word phrases and those corresponding to multi-word phrases.

much of the acoustic segment as possible. For multi-word phrases, this characteristic of the algorithm may be beneficial, as individual words are more likely to be identified out of the entire phrase. This can be seen, for example, in Clusters 12, 16, and 26, where the references are "the history of", "the beginning", and "work together", respectively. Compared to the IWR approach, the hypothesized identities using the DBS approach are much shorter and also turn out to be correct.

One of the primary reasons for degraded identification performance on single word clusters may be errors introduced at the phonetic recognition stage. By passing only a single phonetic transcription on to the identification phase, we essentially make a hard decision regarding the symbolic representation of a particular node. For some examples, errors made in this stage can not be recovered from if the error is consistently observed in the other nodes. Cluster 58 is an example of such an error. The phonetic transcriptions hypothesized for this segment were

```
s ao f f w eh r er
s ao f _ w eh r
s ao f _ w eh r .
```

In all three of these phone sequences, the t-closure and burst are substituted with either an inter-word pause or another 'f'. This leads to a set of 4-phones with no matching tokens in common with the pronunciation variants of "software" found in the baseform dictionary. Using a phone lattice would relax the single phone sequence constraint, but would result in an approach that is similar in spirit to the IWR strategy of the previous section.

6.3 Summary

In this chapter, we proposed two methods for automatically discerning identities

for the clusters generated in the previous chapter. The first approach, which uses a speech recognizer constrained for isolated word recognition, yields high identification accuracy for single word clusters, but has much lower performance on clusters representing multi-word sequences. The second approach, using baseform searching, had more balanced identification accuracy between the single and multi-word clusters, but had poorer performance on single word clusters when compared to the IWR method. While neither technique exhibits optimal identification performance, the complementary nature of the two approaches indicates that a combined strategy (e.g., performing a linear combination of the N-best list results) may result in improved overall identification accuracy.

Chapter 7

Speaker Segmentation

Until this point, we have been concerned with the use of acoustic pattern discovery for the purpose of *lexical* organization of an audio stream. In this chapter we briefly discuss the speaker-specific nature of the segmental DTW as currently implemented, and illustrate how multi-speaker lectures can be segmented as a consequence of this property [85].

As described in Chapter 2, segmental DTW is performed using the frame-level vector representation for each utterance. During this process, no speaker normalization is applied to the feature representation. The consequence of this approach is that the same word spoken by different speakers will tend to have higher distortion than for the same word spoken by the same speaker. In order for the audio information retrieval and word discovery techniques described in the previous chapters to generalize to multi-speaker settings, the segmental DTW distance metric must be modified to account for speaker variability. However, as we will show in this chapter, the speaker specific nature of the segmental DTW path distortions can also be considered a benefit if applied to the task of speaker segmentation in audio recordings. Our approach, which we will describe in Section 7.2.2, requires no supervision, has relatively low computational complexity, and is suitable for audio streams in which speaker turns are fairly

long. This last characteristic makes our approach more suitable for lecture data than say, broadcast news, where speaker changes occur much more frequently. We will discuss this aspect of the segmentation algorithm later in the chapter. The next section gives a brief overview of previous approaches to the problem of speaker segmentation.

7.1 Background

The problem of speaker segmentation, also known as speaker change detection, can be formally stated as follows: Given an audio stream, find all time indices, t, where a speaker change occurs. Most treatments of the problem make the simplifying assumption that only one speaker is speaking at any particular point in time. A related problem is that of speaker clustering, where the input is a set of M segmented utterances, $U_1 \ldots U_M$, and the output is a set of N groups of utterances where all of the utterances in a particular group were spoken by the same speaker. We will limit the scope of this chapter to the problem of speaker segmentation only.

Although there are many reasons why one might want to segment an audio stream by its constituent speakers, two major reasons stand out. First, for audio documents, speaker changes are often considered natural points around which to structure the document for navigation by listeners. In broadcast news, for example, speaker changes typically coincide with story changes or transitions. Audio recordings of meetings, presentations, and panel discussions are additional examples where organizing audio segments by speaker identity can provide useful navigational cues to listeners.

A second motivation for speaker segmentation relates to automatic transcription of the speech. In many scenarios, the performance of automatic speech recognition can benefit greatly from speaker adaptation, whether supervised or unsupervised. Speaker segmentation, while not a strict pre-requisite for speaker adapation, is important for performing adaptation on multi-speaker data, as it provides the recognizer with data that is homogenous with respect to the speaker.

7.1.1 Related Work

Speaker change detecton has been well examined by many researchers in recent years. Early approaches to change detection viewed it as an extension of the speaker identification problem. If information about the speakers in the test data is known *a priori*, then individual speaker models can be used to identify the most likely speaker for each utterance. Segmentation can then be performed trivially using the hypothesized speaker identities. The main drawback of this supervised

framework is that it does not generalize to scenarios in which unknown speakers are present in the test data.

Unsupervised approaches to the problem have been more commonly seen as of late, and typically consist of two components: a distance metric for comparing two segments of speech, and a method for determining change points in the audio stream using the distance metric.

In [39] and [108], the authors propose comparing adjacent segments of speech to each other using the generalized likelihood ratio as a distance metric. The audio stream is divided into a set of non-overlapping segments on the basis of energy measurements, or by using fixed length windows on the order of several seconds in length. The distance between segments X and Y is then computed by way of a statistical hypothesis test between two competing hypotheses,

$$\begin{aligned} \mathrm{H}_0: &\quad X \text{ and } Y \text{ generated by the same speaker} \\ \mathrm{H}_1: &\quad X \text{ and } Y \text{ generated by different speakers} \end{aligned}$$

Let X and Y be represented by feature vector sequences $\{\mathbf{x}_i\}_{i=1}^{n_x}$ and $\{\mathbf{y}_i\}_{i=1}^{n_y}$, respectively. Under the assumptions that the x_i and y_i are independent and identically distributed and that the underlying speaker models are normally distributed, and be estimated by assuming Gaussian distributions for the underlying speaker models, the log likelihood of the data under the the two hypotheses will be

$$\begin{aligned} \log L(Z; H_0) &= \sum_{i=1}^{n_x} \log N(\mathbf{x}_i; \hat{\mu}_Z, \hat{\Sigma}_Z) + \sum_{i=1}^{n_y} \log N(\mathbf{y}_i; \hat{\mu}_Z, \hat{\Sigma}_Z) \\ \log L(Z; H_1) &= \sum_{i=1}^{n_x} \log N(\mathbf{x}_i; \hat{\mu}_X, \hat{\Sigma}_X) + \sum_{i=1}^{n_y} \log N(\mathbf{y}_i; \hat{\mu}_y, \hat{\Sigma}_y), \end{aligned}$$

where $Z = X \cup Y$, and the hat denotes the maximum likelihood estimate for a particular parameter. The log likelihood ratio (LLR) can then be used as a distance betwen X and Y

$$D_{LLR} = \log \frac{L(Z; H_1)}{L(Z; H_0)} \tag{7.1}$$

This distance will be strictly non-negative, and can be used for change detection either by employing a threshold between adjacent utterances, or for speaker clustering by employing a clustering algorithm over the global distance matrix. Variations on this approach have substituted alternative distance measures such as the symmetric Kullback-Leibler measure [30].

The current dominant approach to speaker change detection uses the Bayesian Information Criterion (BIC) as a metric for comparing two segments [19, 20, 30, 115]. The BIC is a model selection criterion meant for optimally choosing

the model that best represents a given set of data from a set of candidate models. That is, given a set of models, M_i, and n independent data samples $Z = \{\mathbf{z}_j\}_1^n \in \mathbb{R}^d$, the best model is the one that maximizes

$$\mathrm{BIC}_i = \log L(Z; M_i) - \frac{1}{2} k_i \log n, \tag{7.2}$$

where $L(Z; M_i)$ is the likelihood of Z under model M_i and k_i is the number of parameters in model M_i. In the case of speaker change detection, there are only two models to choose from: M_0 if X and Y are modeled by the same underlying process, or M_1 if X and Y are modeled by two different processes. The distance implied by the BIC is then

$$D_{\mathrm{BIC}} = \log \frac{L(Z; M_0)}{L(X; M_1) L(Y; M_1)} - \frac{1}{2} \log n (k_0 - k_1), \tag{7.3}$$

This framework is essentially identical to the hypothesis test for the LLR metric, although the model selection formulation introduces a built-in threshold: a positive value of D_{BIC} indicates a likely change in speaker between X and Y. It is worth noting when the competing models both consist of Gaussians, D_{BIC} and D_{LLR} differ only in the penalty term which accounts for the number of parameters of the two models, i.e.,

$$D_{\mathrm{BIC}} = D_{\mathrm{LLR}} - \frac{1}{2}(d + \frac{d(d+1)}{2}) \log(n_x + n_y). \tag{7.4}$$

A notably different distance measure for comparing two speech segments uses vector quantization [75, 56]. In this approach, a vector codebook $C(\mathcal{X})$ is generated using the speech frames in one segment. Speech frames from the second segment are then compared to the first segment by taking the distance to the nearest vector codeword.

$$d(\mathcal{X}, \mathbf{y}_i) = \min_{\mathbf{x}_c \in C(\mathcal{X})} d(\mathbf{y}_i, \mathbf{x}_c) \tag{7.5}$$

The overall distance between the two segments is then given by

$$d_{\mathrm{VQ}}(\mathcal{X}, \mathcal{Y}) = \frac{1}{N_y} \sum_{i=1}^{N_y} d(\mathcal{X}, \mathbf{y}_i). \tag{7.6}$$

Because this quantity is not symmetric, the choice of which segment to use for the codebook is a factor that may affect performance.

To date, the vast majority of approaches to speaker change detection have used one of the distance measures mentioned above [75, 56, 1, 70, 118]. The differences between many of these approaches has been in the manner in which the distance measures are used to produce a segmentation. Some take the

distance between two halves of a growing window [20, 1, 70, 115]. Others take two halves of a fixed-size analysis window that slides through time [56, 71]. In these scenarios, a change point is hypothesized when the distance exceeds a certain threshold.

7.2 Speaker Segmentation via Segmental DTW

7.2.1 A Segmental DTW based Distance

In this section, we present a novel distance metric for comparing two speech segments that is based on the segmental DTW algorithm introduced in Chapter 2. Our approach is based on the idea of finding word-level speech patterns that are repeated by the same speaker. The segmental DTW distance with which we propose to compare two utterances is based on the best alignment path fragment between the two utterances. More formally, we define the segmental DTW distance between two speech segments $\mathcal{X}$ and $\mathcal{Y}$ to be,

$$\mathcal{D}_{\mathsf{s}}(\mathcal{X}, \mathcal{Y}) \triangleq \min_{\varphi \in \Phi(\mathcal{X}, \mathcal{Y})} \mathsf{d}_{\varphi}(\mathcal{X}, \mathcal{Y}), \tag{7.7}$$

where $\Phi(\mathcal{X}, \mathcal{Y})$ is the set of alignment path fragments discovered during the segmental DTW procedure as described in Chapter 2. That is, the distance is given by the distortion of the minimum distortion alignment path fragment between the two segments.

For two utterances that share both a word and a speaker in common, the alignment path is likely to match the common word and result in a low distortion. On the other hand, if the utterances are spoken by different speakers, the distortion is likely to be much higher, even if the utterances share a word in common, simply because of variation in speaker characteristics in speaking that particular word. Of course, we cannot guarantee that every pair of adjacent utterances with a common speaker will also share a speech pattern on the order of a word. However, by processing blocks of utterances, the likelihood of finding such a repeating pattern is increased. A similar approach was applied to the problem of speaker verification in [38].

Though our segmental DTW distance metric is relatively straightforward to describe, it differs from the proposed solutions reviewed in the previous section in two important ways. First, speech segments are not considered as "bags of frames", where each frame is processed independently of the other frames in the segment. Instead, the alignment path fragments require frames to be considered in the context of other frames - as part of a sequence, rather than a representative token on its own. The second way in which our approach differs from traditional distance measures, is that utterances are compared on the basis

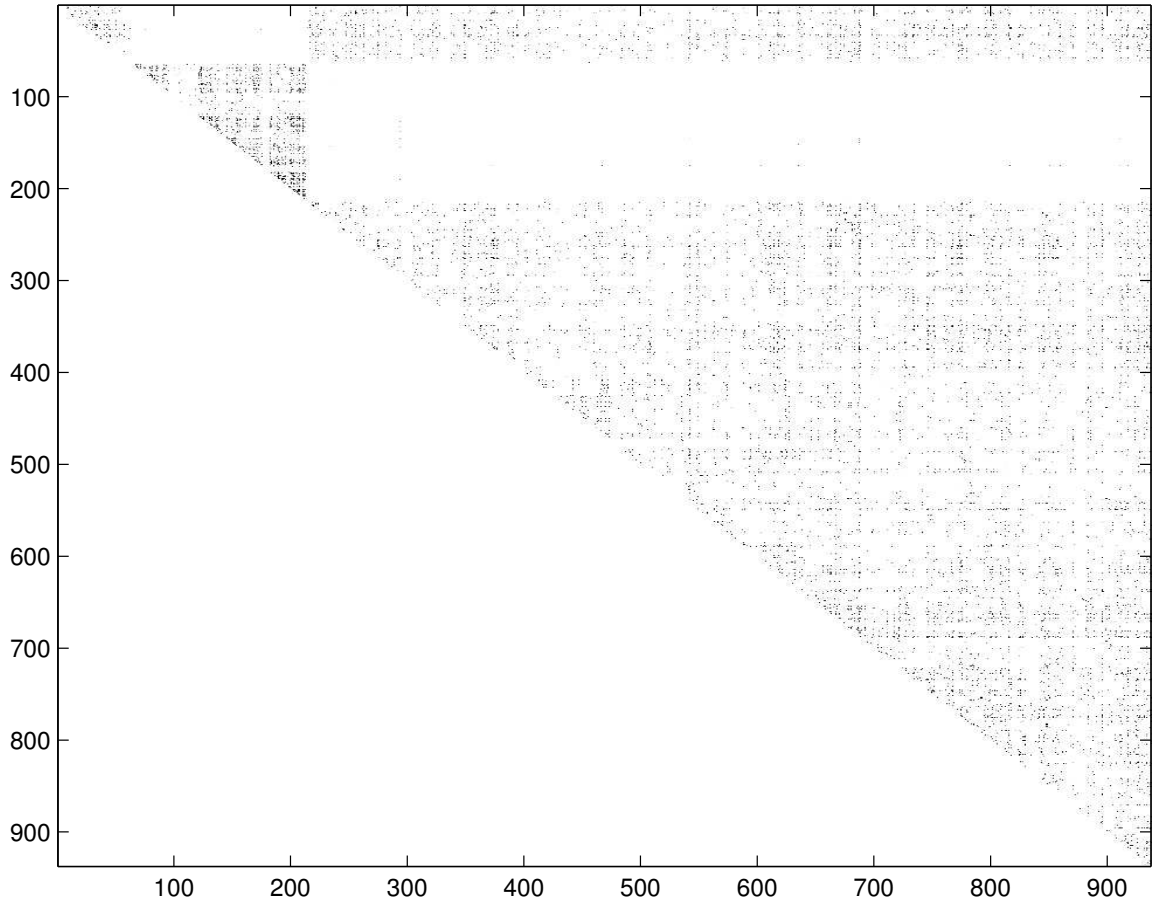


Figure 7-1: Utterance level similarity matrix for a physics lecture consisting of two main speakers and three main segments. The intensity of a cell (i, j) indicates the similarity of utterance i and utterance j using the minimum distortion alignment path fragment computed by the segmental DTW algorithm. Darker cells indicate higher similarity.

of their most similar token (in this case, a sequence of frames), rather than by averaging all tokens from both utterances.

The effectiveness of this approach can be seen qualitatively by considering the utterance level similarity matrix for a physics lecture as shown in Figure 7-1. For this matrix, utterance distances are converted into the similarities using a fixed threshold, θ.

$$\mathrm{S}(\mathcal{X},\mathcal{Y}) = \begin{cases} 1 - \frac{\mathrm{D}(\mathcal{X},\mathcal{Y})}{\theta} & \text{if } \mathrm{D}(\mathcal{X},\mathcal{Y}) < \theta \\ 0 & \text{otherwise} \end{cases} \tag{7.8}$$

Even without access to the true speaker change points for the lecture shown in Figure 7-1, the similarity matrix exhibits a distinct block structure that makes

it relatively trivial to visually identify the speaker change points. What is less clear is how we can move from this visual representation into one that is more amenable to automatic segmentation. We address this question in the next section.

7.2.2 Building a segmentation profile

In this section, we propose a method for producing a *segmentation profile* from the similarity matrix shown in the previous section. A segmentation profile is simply a time varying measure of how likely an utterance is to be a speaker "change" point, i.e. a discontinuity in the similarity matrix, A. Based on this, we propose to track the normalized sum of the cells under a triangular region that slides along the main diagonal of the similarity matrix. This method can be summarized by the diagram in Figure 7-3. The dissimilarity of segment k to its adjacent segments can be expressed as a function of the normalized sum of nearby cells in the similarity matrix, A.

$$V_D(k) = -\log\Big(\frac{1}{F_D(k)} \sum_{i=k-D+1}^{k} \sum_{j=k}^{i+D-1} A(i,j)\Big) \tag{7.9}$$

where $F_D(k)$ is a normalizing term that represents the number of cells being added, and is given by

$$F_D(k) = \begin{cases} f(D) - f(D-(N-k)) & \text{if } N-k < D, \\ f(D) - f(D-k) & \text{if } k < D, \\ f(D) & \text{otherwise.} \end{cases} \tag{7.10}$$

and $f(n)$ is just the number of cells in a triangle with legs of length n,

$$f(n) = \frac{n(n+1)}{2} \tag{7.11}$$

Figure 7-2 illustrates a dissimilarity profile for the physics lecture from Figure 7-1, using a diagonal band, D, of 100 utterances.

As the triangular window slides down the main diagonal of the similarity matrix, the sum of the cells in the window will reach local minima at discontinuities in the matrix. The intuition behind this approach can be readily seen by considering Figure 7-1 again. For points in the lecture that are very similar to adjacent regions, more of the cells in the triangular region centered at that particular point tend to be very dark. Conversely, points in the lecture that are dissimilar to adjacent regions have fewer dark points. From a graph perspective, this method is equivalent to evaluating a hypothesized discontinuity by summing the weights of the arcs interconnecting a block of nodes around that discontinuity.

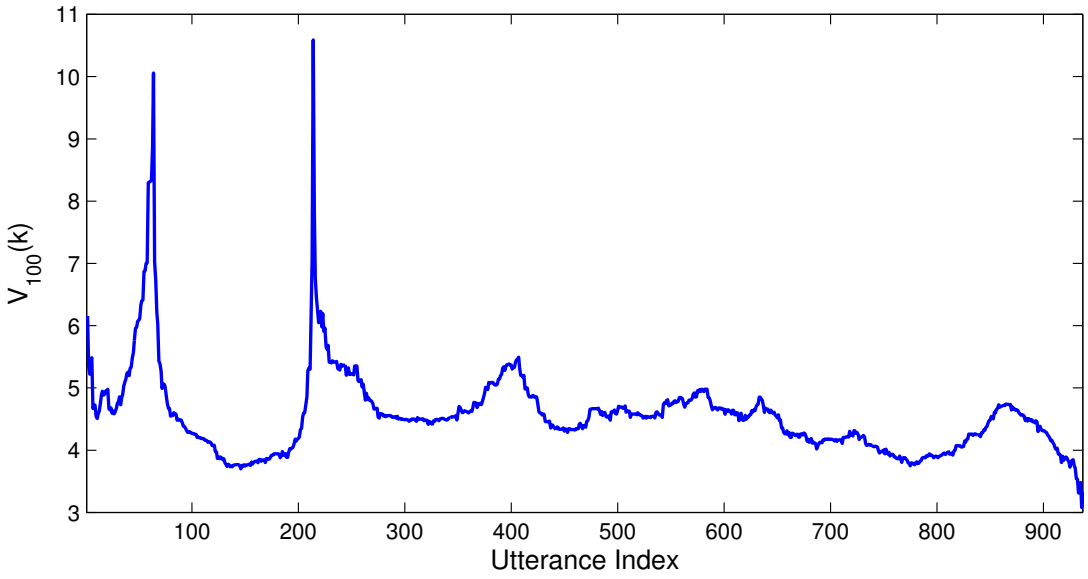


Figure 7-2: The log dissimilarity profile for the physics lecture from Figure 7-1. The width of the diagonal band considered is D = 100 utterances.

This interpretation is shown in Figure 7-4.

Aside from its intuitive appeal, the choice to to consider a small triangular region is motivated by computational reasons as well. With this method, for any particular utterance index, k, it is only necessary to compute similarity scores for pairs of utterances that are within D of k. It follows that the overall running time will be $O(D^2N)$, making the computation linear in the length of the audio stream.

7.3 Data Description

The data used in our segmentation experiments was a subset of the lectures in the MIT World lecture corpus. Unlike the lectures examined in the previous chapters, we specifically selected lectures consisting of speech from multiple presenters speaking for a significant period of time (several minutes or more). At the time of this writing, these lectures were all publicly available on the MIT World website. The subject material contained within the lectures is wide-ranging; topics include hurricane relief response, medical technology, weapons proliferation policy, and quantum mechanics. In all, the data represents more than ten hours of speech contributed by at least 25 different speakers. A summary of the lectures is given in Table 7.2.

One of the useful aspects of this data is the availability of high level reference segmentations provided by organizers of the MIT World site. Each lecture is

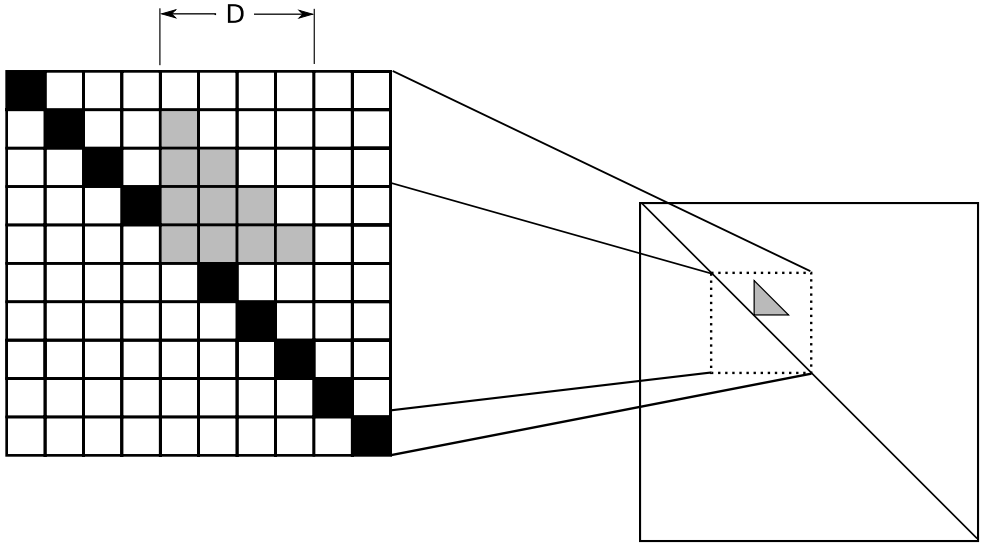


Figure 7-3: The diagonal region used to compute the segmentation profile. In this example, D = 4.

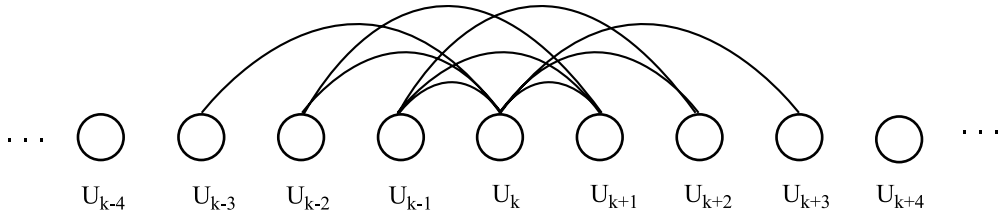


Figure 7-4: A graph perspective of building the similarity profile at utterance U_k. The nodes in the graph represent utterances, while the arcs have weights given by the segmental DTW distance metric. Here, only the arcs being summed to get $V_D(k)$ are shown. In this example, D = 4.

NOTES ON THE VIDEO (Time Index):
Video length is 2:01:35

The Rev. Amy McCreath, Coordinator, Technology and Culture Forum, opens the event.

At 2:00, Daniel Roos, Co-Director, Engineering Systems Division and Associate Dean, School of Engineering, offers some remarks on J. Herbert Hollomon.

At 11:13, Kosta Tsipis, the former director of MIT's Security Studies Program, introduces the panel.

Owen Cote begins at 15:14.

Jeanne Guillemin begins at 34:32.

Steven Miller begins at 57:25.

Philip Morrison begins at 1:26:00.

Q&A begins at 1:38:47.

Table 7.1: Example of a human generated speaker change summary provided on the MIT World web site.

accompanied by a summary page which includes an index of major landmarks in the lecture as judged by a human listener. For the most part, these landmarks typically correspond to speaker changes, but not all speaker changes are included. For instance, minor speaker turns occuring as part of a Q & A session are not individually labeled, but are grouped together as a separate section. An example summary page is shown in Figure 7.1. Since these landmarks are intended to help guide listeners to important times in the lecture, we focus on the task of automatically finding these landmarks and not just arbitrary speaker changes. In some ways, using these landmarks as our target references is less ambitious than traditional speaker change detection tasks as it does not require exhaustively finding all speaker changes. On the other hand, determining how to select the particular change points that constitute important boundaries to human listeners is also a nontrivial task.

Title	# Speakers	Length
1. Transforming Healthcare	**5**	**1:31**
Researchers show how advances in electrical engineering and computer science can be applied to medical science. Among the technologies described are 3D imaging techniques for providing visualizations in computer aided surgery, and knowledge-based systems that can aid medical diagnoses and reduce mistakes in treatment.		
2. Engineering Human Machine Relationships	**5**	**1:32**
Multiple presentations are made on the theme of technology improving the connection between people and machines. Presenters give an overview of new technologies in the fields of spoken language interfaces, cognitive augmentation, and social robotics.		
3. Transforming the Next Century	**3**	**1:07**
In this panel, speakers focus on the shift in electrical engineering and computing toward biology, and toward physics. The first part of the lecture concentrates on biological computing for the purposes of engineering drugs and mapping the regulatory pathways in human cells. The second part of the lecture is concerned with quantum computing.		
4. Weapons of Mass Confusion	**7**	**2:01**
Panelists consider the problem of defining weapons policy in light of the varied types of weapons of mass destruction in existence. The discussion focuses on current policy as well as recommended changes.		
5. Response to Hurricane Katrina	**4**	**1:58**
A panel of speakers discuss the lessons learned from the U.S. response to Hurricane Katrina and consider likely problems faced by government and first responders in the event of future disaster situations.		
6. Future of Flight	**5**	**1:44**
On the 100th anniversary of the first manned air flight, speakers reminisce on the history of flight and speculate about future flight technologies and infrastructure.		

Table 7.2: A description of the 6 lectures examined in this chapter. The number of speakers listed for each lecture is taken from the MIT World web site, and is a lower bound, as most of the discussions also include questions from the audience.

7.4 Segmentation

We begin by considering segmentation profiles for the lectures in our test set. In Figures 7-5 and 7-6, we plot the dissimilarity profiles for the six lectures described in Table 7.2. We observe that these dissimilarity profiles possess the desirable characteristics of having distinct peaks that coincide with the reference boundaries. Unfortunately, intrinsic variation in the profiles result in many spurious peaks, so simply picking the peaks in the profile would vastly overgenerate the potential segment boundaries. We instead utilize an alternative processing technique which focuses on finding *distinct* peaks.

7.4.1 Finding Distinct Peaks

We note two characteristics of distinct dissimilarity peaks that differ from spurious peaks. First, the distinct peaks happen to persist when filtered with smoothing windows of different widths. Second, they they are significantly higher than neighboring values. The former characteristic can be observed by performing so-called scale-space filtering on the dissimilarity profile, as shown in Figure 7-7. Scale-space filtering is a technique widely used in pattern recognition and computer vision which generates multiple resolution representations of a signal by filtering with Gaussian windows, $G_\sigma(k)$, of different variances [121]. For a dissimilarity profile $V(k)$, we denote the smoothed version of the signal as

$$V_\sigma(k) = G_\sigma(k) * V(k). \tag{7.12}$$

where $G_\sigma(k)$ is a Gaussian window with variance σ. Progressively larger values of σ reduce the number of peaks in the smoothed signal. An example of a smoothed profile generated using a σ of $1/30$ is shown in Figure 7-8. We then take the peaks from the smoothed profile and recover the peaks from the original profile by backtracing along the profiles with progressively larger values of σ. This process is shown in Figure 7-9. At this stage, we are left with peaks that "survive" the scale-space filtering process.

We next turn to the second characteristic of distinct peaks listed above. In order to exploit the characteristic that distinct peaks rise higher above local neighboring values than non-distinct peaks, we can use the smoothed profile with the lowest value of σ as a form of gain control by subtracting it from the original profile. We can then threshold this difference

$$V(p) - V_\sigma(p'), \tag{7.13}$$

to produce a set of hypothesized segment boundaries. In Eq. 7.13, p' is a peak found in the smoothed version of the profile, $V_\sigma(k)$, and p is the corresponding peak found by backtracking to the original profile, $V(k)$. Figure 7-9 illustrates

this thresholding procedure. The threshold used in our experiments was set to a fixed value of 0.2 by using a held out lecture as development data. In general, however, the peak picking algorithm is relatively insensitive to the choice of threshold value because of the peakiness of the profile at actual change points.

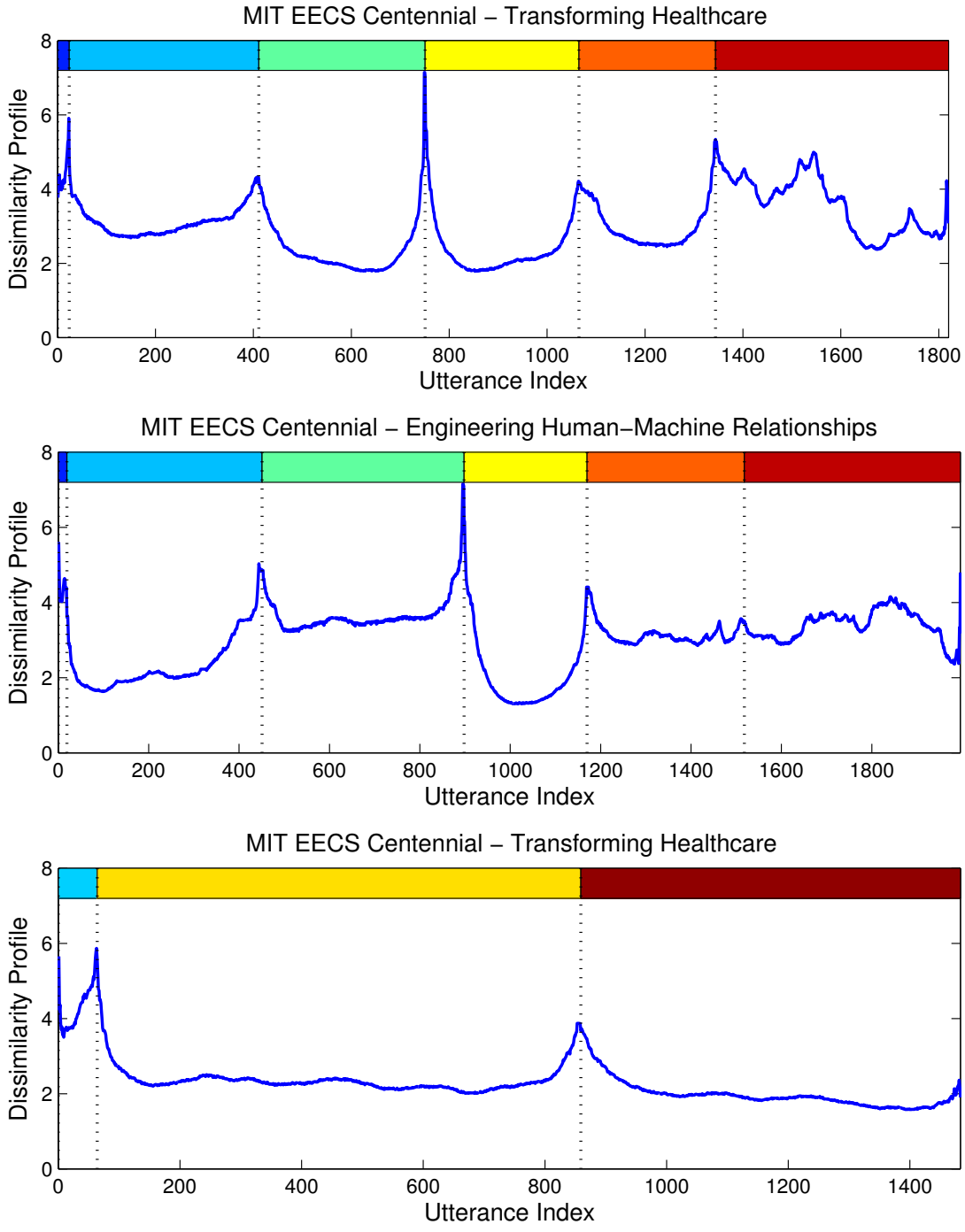


Figure 7-5: Dissimilarity profiles for lectures 1, 2, and 3. The dashed vertical lines indicate where the reference boundaries occur.

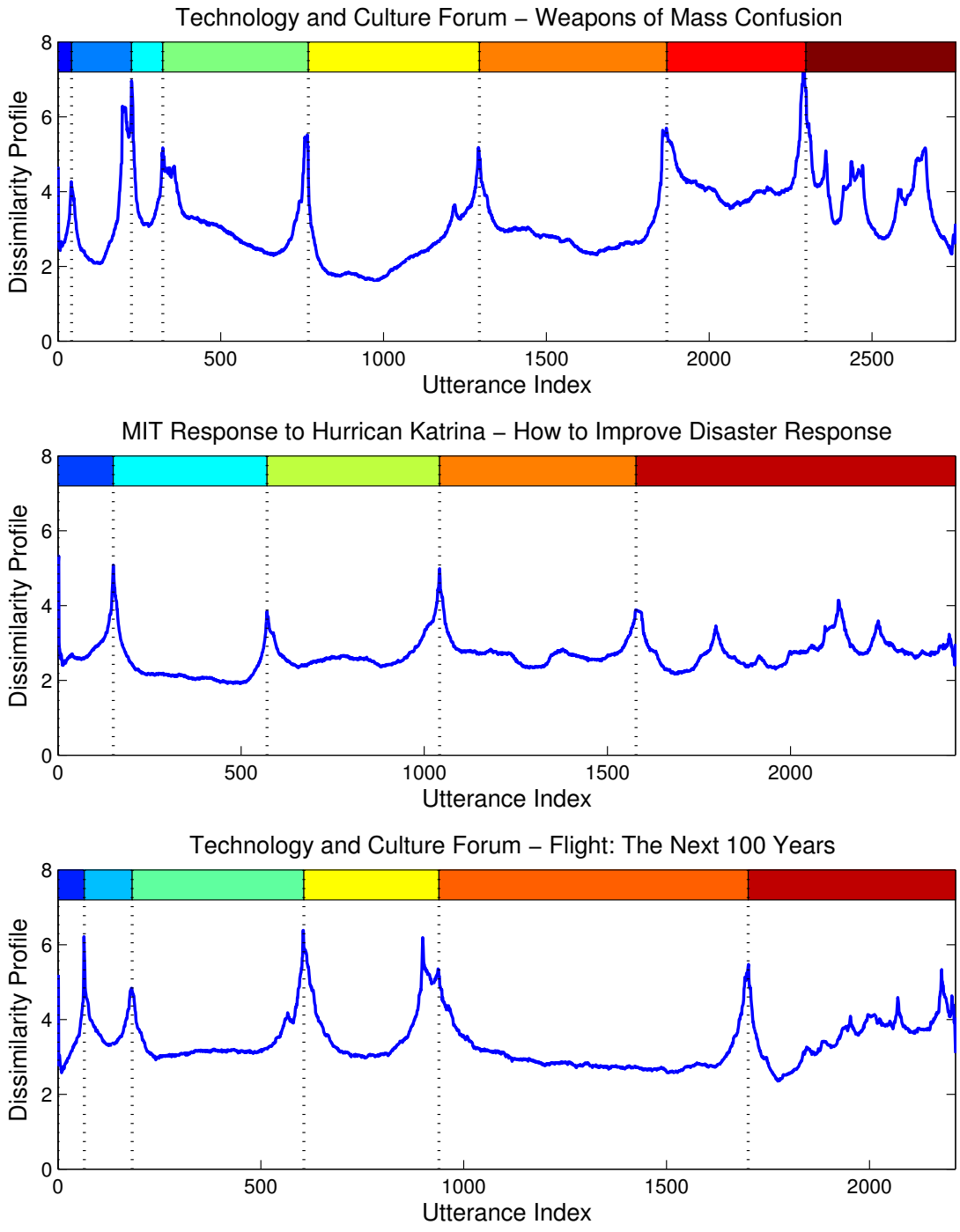


Figure 7-6: Dissimilarity profiles for lectures 4, 5, and 6. The dashed vertical lines indicate where the reference boundaries occur.

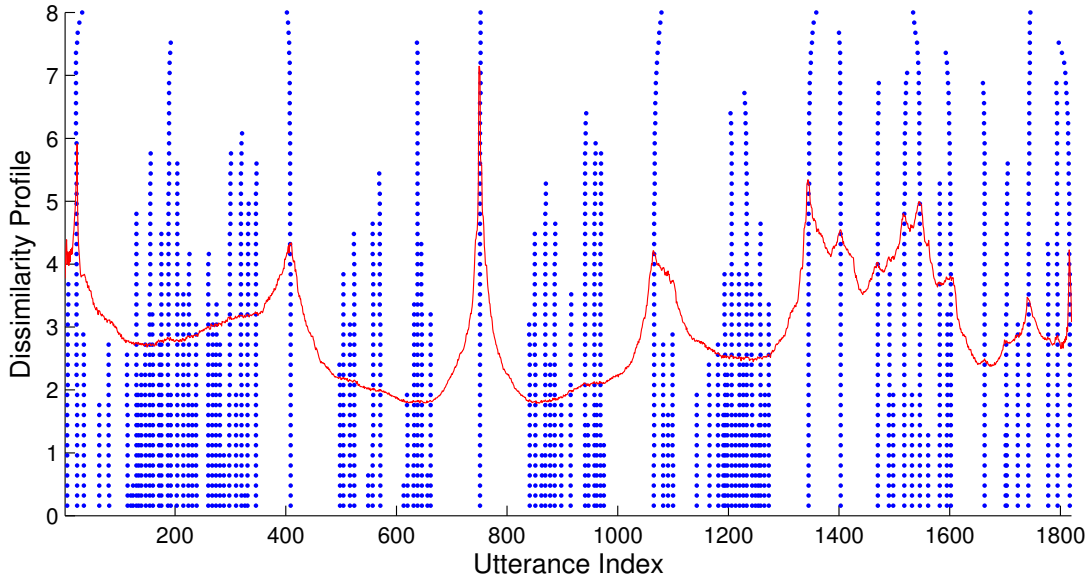


Figure 7-7: Peak locations for scale-space filtered versions of the dissimilarity profile for Lecture 1. The original dissimilarity profile is shown in red. The peak locations are shown overlaying the profile, with peaks for smaller values of sigma shown at higher levels.

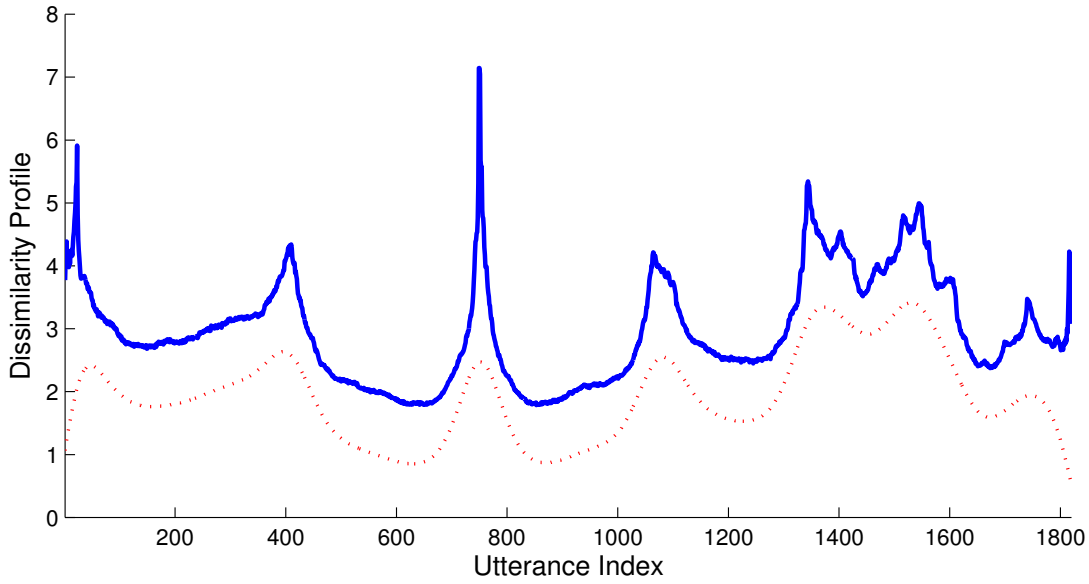


Figure 7-8: Original dissimilarity profile for Lecture 1 (in blue) and the smoothed version of the profile with $\sigma = 30$ (in red). The smoothed profile is vertically offset for visual clarity.

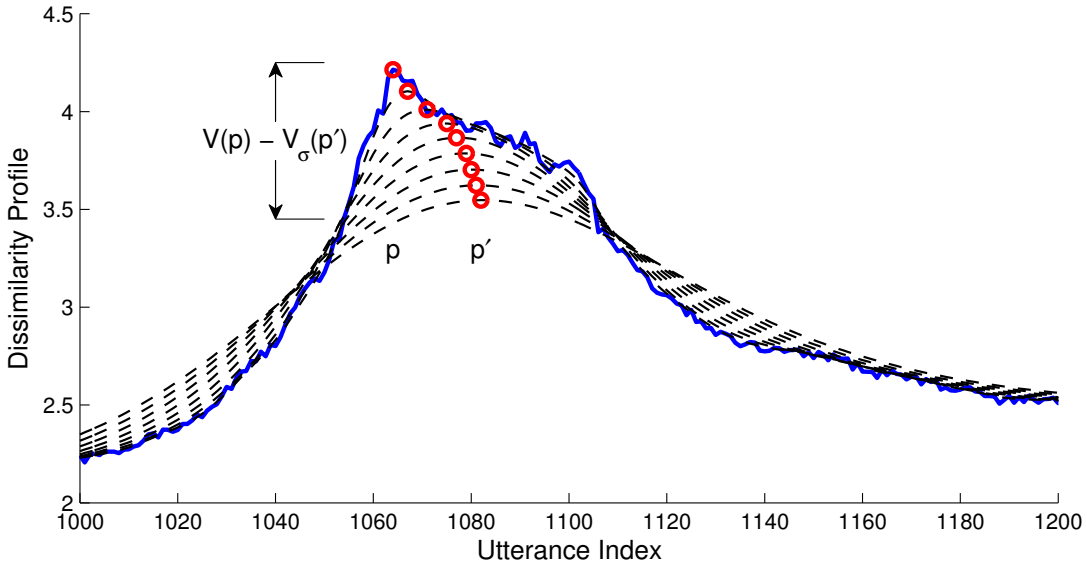


Figure 7-9: Illustration of scale space filtering to backtrace peaks in the dissimilarity profile. The solid blue line is the original profile, and the dashed black lines are smoothed versions of the profile for different values of σ. The red circles represent the peak found at the lowest value of σ, backtraced to find the peak on the original profile. The labeled quantity, $V(p) - V_\sigma(p')$, is the value which is thresholded in order to determine how distinctive a peak is in relation to its neighboring values.

7.5 Analysis and Discussion

The results of our segmentation procedure are summarized in Table 7.3. Rather than use a floating threshold which yields a detection error tradeoff curve, we instead use a fixed threshold and evaluate the precision and recall of the resulting segmentation. Our reason for doing this is because for actual deployment, an actual segmentation is required, meaning the utility of the algorithm is strongly tied to the threshold selected.

In Figure 7-11, a histogram of distances of hypothesized boundaries to closest reference boundaries is shown, illustrating the distribution of proposed boundaries to correct boundaries. For our evaluation, a hypothesized boundary is marked as correct if it falls within seven utterances of a true boundary. In temporal terms, we observed that the average distance between those hypothesized boundaries marked as correct and reference boundaries was 8.5 seconds. Table 7.3 shows that with the development-set selected threshold, the overall recall rate is 100.0%, meaning that all of the human annotated boundaries are found by our segmentation procedure. The overall precision rate is 80.0%, meaning that of the 35 boundaries proposed, seven were not on the list of human-proposed boundaries. It should be noted, however, that all of these "false alarms" actually do correspond to speaker change boundaries that are not annotated in the reference. Therefore, while, these should still be considered errors in the context of our experiment, they may not necessarily detract from the performance of the system in an actual deployment. Indeed, when the automatically generated segmentations are displayed together with the reference segmentations in Figure 7-10, one can observe that all of the false alarms occur in the last segment of each lecture, which corresponds to the Q&A section where multiple, unlabeled speaker changes take place. Aside from these few false alarms, the automatic boundaries correlate well with the reference boundaries, indicating that our proposed procedure may prove useful for providing navigational boundaries for this particular task.

7.6 Summary

The goal of this chapter has been to illustrate how the speaker specific nature of segmental DTW can be exploited to perform speaker segmentation. To that end, we have implemented and evaluated a segmentation algorithm that is able to find "significant" speaker changes as evaluated by human listeners. Although we recognize that there may be more optimal methods for generating segmentations from a set of inter-utterance distances than the one we describe, our procedure is relatively straightforward and computationally efficient. Moreover, our main interest is not in finding an optimal segmentation algorithm *per se*, but rather in exploring the potential of segmental DTW as a novel way of comparing

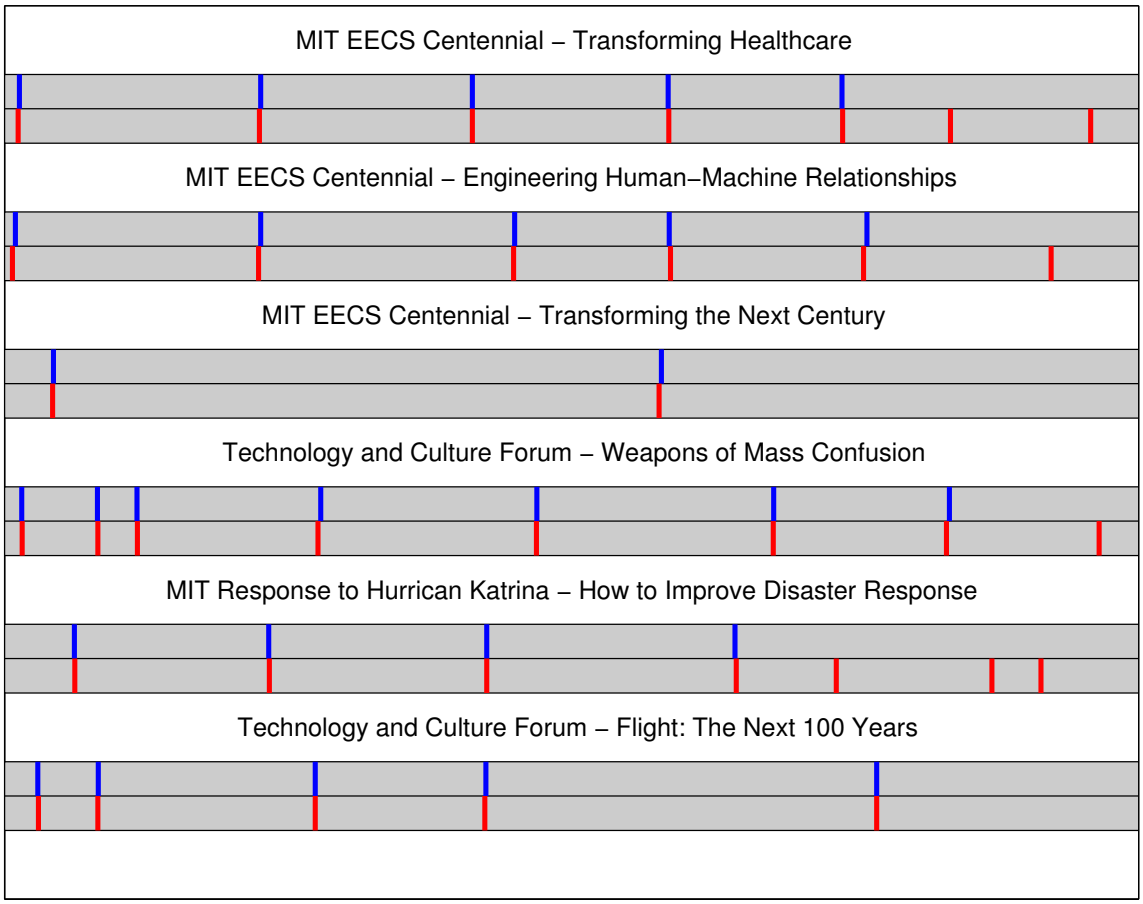


Figure 7-10: Comparison of human generated segmentation with automatic segmentation for the 6 lectures used in this chapter. For each lecture, the reference boundaries are shown as blue lines in the upper panel, and the automatically generated boundares are shown as red lines in the lower panel.

Lecture	# Ref Bounds	# Hyp Bounds	Precision (%)	Recall(%)
1.	5	7	71.4	100.0
2.	5	6	83.3	100.0
3.	2	2	100.0	100.0
4.	7	8	87.5	100.0
5.	4	7	57.1	100.0
6.	5	5	100.0	100.0
Overall	**28**	**35**	**80.0**	**100.0**

Table 7.3: Individual and overall automatic segmentation statistics for the lectures processed in this chapter. # Ref Boundaries refers to the number of non-trivial segmentation boundaries as provided by the human lecture summary. # Hyp Boundaries is the number of segmentation boundaries hypothesized by our segmentation algorithm. Precision is the percentage of hypothesized boundaries that are incorrect, while Recall is the percentage of correct boundaries that are returned in the set of hypothesized boundaries.

utterances. In that context, the results of this chapter are promising, as they demonstrate that segmental DTW can indeed be used as a way to break lengthy audio streams into more manageable segments by their constituent speakers.

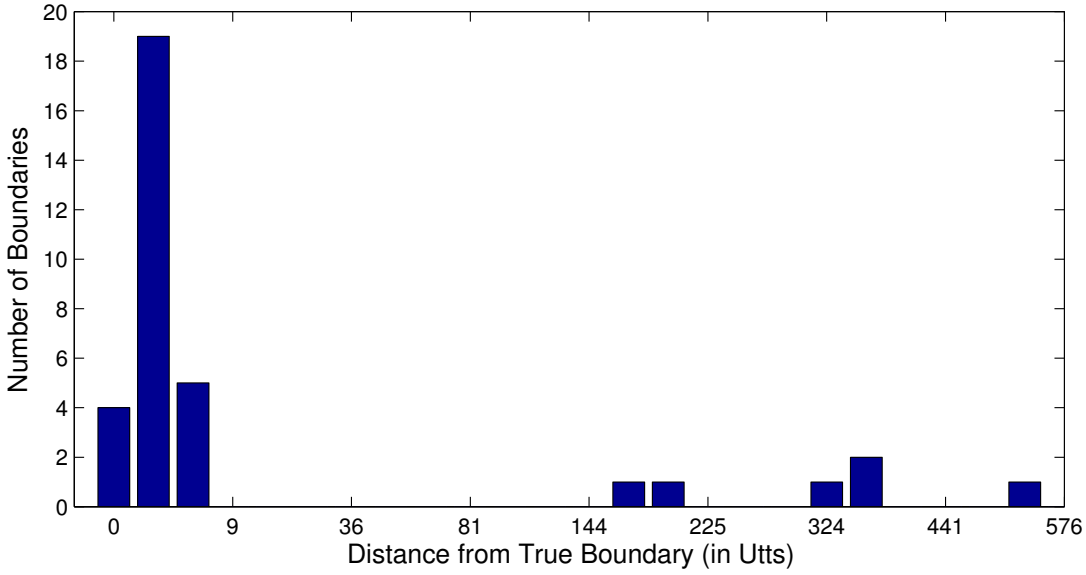


Figure 7-11: Histogram of boundary errors for hypothesized boundaries. The boundary error is the distance of the proposed boundary to the closest reference boundary. The majority of proposed boundaries lie within 7 utterances of a reference boundary. Those with larger errors are considered false alarms.

Chapter 8

Conclusions and Future work

In this chapter, we summarize the contributions of this thesis and outline some directions for future work.

8.1 Summary and Contributions

The work described in this thesis was initially motivated by the problem of acquiring knowledge in an unsupervised manner from large amounts of single speaker audio. Unsupervised knowledge acquisition has been a largely unexplored area in the speech community, partially because of the wide availability of large amounts of labeled training data and the high level of performance currently attained by many state of the art automatic speech recognition systems, which are trained in a supervised, or semi-supervised manner. The techniques we have discussed in this thesis represent a departure from these traditional approaches to speech processing and a step towards understanding how a word lexicon can be automatically acquired from continuous speech. Although researchers have previously attempted to perform unsupervised word acquisition on sequences of phone labels, this work represents one of the first attempts to accomplish the

same objective without first converting the speech signal into an intermediate symbolic representation.

As a first step, we attempted to address the problem of finding reccuring patterns directly from the acoustic signal, which resulted in our development of a novel algorithm for finding matching subsequences between pairs of utterances. Although the segmental DTW approach that we propose is based on a well-known dynamic programming technique that has seen wide use in speech before, our application of the technique is unique in two respects: First, neither of the input utterances represents the template of a known entity. Second, the search process results in fragments that align subsequences of the original utterances rather than the entirety of one to the other.

In order to more fully understand the nature of matches found by the segmental DTW algorithm, we analyzed the characteristics of path fragments with high frame-level matching accuracy at both the phone level and the word level in Chapter 4. Our primary finding was that matching accuracy is closely related to path length and distortion, which can both be used as confidence measures for estimating the quality of a given path fragment.

One of the major consequences of performing segmental DTW on multiple utterances is the generation of multiple alignment path fragments covering the same point in time. Using these alignment paths, we presented a method for representing the audio stream as an abstract adjacency graph, to which standard graph clustering algorithms can be applied. With few exceptions, the clusters produced from the adjacency graph were shown to have a high purity and significant relevance to the underlying lecture topic.

As an extension to the pattern discovery technique of Chapter 5 , we proposed a method for automatically identifying the underlying lexical entity corresponding to the common pattern of each cluster. The identification procedure we described combines information from multiple instances of the same wod to output an hypothesis and was able to achieve relatively high identification accuracy on single word clusters. We discuss possible modifications to this approach for improved identification of multi-word clusters in Section 8.2.2.

In Chapter 7, we showed that the speaker specific nature of the segmental DTW allows us to perform speaker segmentation on multi-speaker lectures. The approach we describe is novel both in its methods and objectives. Unlike most approaches to speaker segmentation which use either the BIC or log-likelihood ratios to compare speech segments, we employ a segmental DTW distance measure. The use of the segmental DTW measure is conceptually different from more standard approaches in that the comparison is based on the best matching sequence of frames between utterances instead of the average difference of all frames evaluated in a non-sequential manner. From the outset, our goal was to attempt recovery of *significant* speaker changes, as evaluated by the human listeners. Our evaluations showed a high level of performance on

this task, finding all significant speaker changes across a variety of lectures and speakers while rarely proposing false boundaries.

8.2 Future Work

Alhough we have presented end-to-end methods for unsupervised word acquisition, from processing of raw audio through to identification of clustered acoustic segments, the work presented in this thesis represents only an initial investigation into the more general problem of knowledge acquisition from speech. Many directions for future work remain, and we expand upon some of them in this section. We divide the specific topics into three categories: those dealing with subsequence matching, those dealing with clustering and identification, and finally applications for acoustic pattern matching and unsupervised word acquisition.

8.2.1 Segmental DTW

The segmental DTW algorithm that we proposed was designed for the purpose of finding matching subsequences between the acoustics of two continuous spoken utterances. Here, we discuss extensions or alternatives to the segmental DTW algorithm.

Image-based path detection

In Chapter 4, we described the segmental DTW algorithm in detail. The fundamental motivation fo this algorithm is to determine whether a pair of utterances contains a matching subsequence that might correspond to a common word. Visual inspection of the distance matrix computed for utterance pairs as shown in Figure 8-1 indicates that matching subsequences are realized as diagonal regions of low distortion. Based upon this observation, it would be interesting to apply image processing techniques to the distance matrix to try to detect the presence and location of these regions. Unlike some standard object recognition tasks such as eye detection and face recognition, the detection of rigid diagonal bands of low distortion is unencumbered by complicating factors such as variation due to rotation, illumination, and scale.

Speaker Independence

The results of our speaker segmentation experiments from Chapter 7 revealed that the within-speaker path distortions tend to be smaller than between-speaker path distortions. This speaker specificity can be partially attributed to the spectral representation used as the input to the segmental DTW algorithm, as mel-scale cepstral coefficients have been shown to be highly effective for speaker

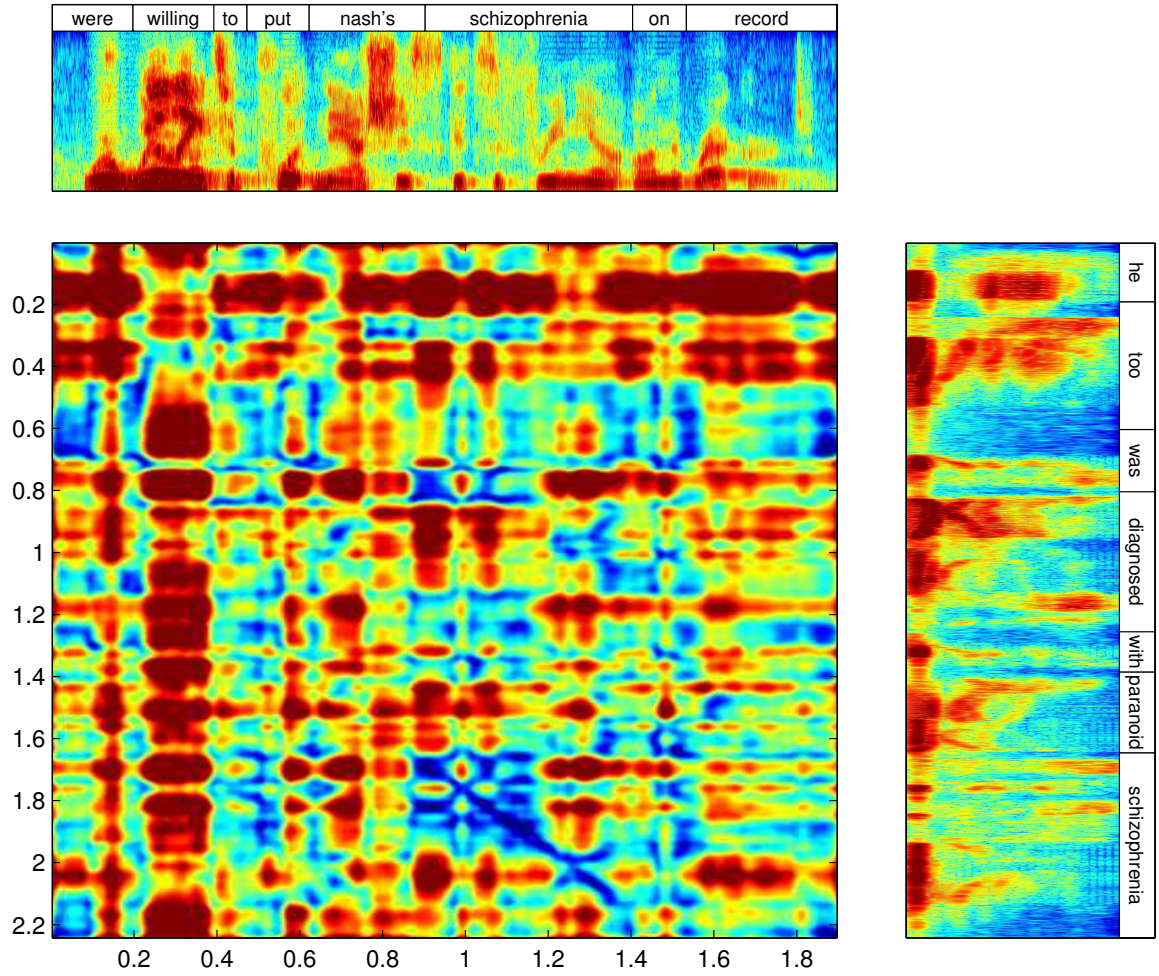


Figure 8-1: Distance matrix and accompanying spectrograms from example shown in Figure 3-6. Parts of the distance matrix matching the same word together appear as a diagonal band of low distortion. These paths may be discoverable using image processing techniques.

recognition applications [95]. When used for speaker segmentation, this property of our representation can be very useful. Additionally, when processing audio data which primarily consist of a single main speaker, the speaker specific nature of our algorithm is not a major concern. However, in situations requiring word discovery across utterances from multiple speakers, we would like the primary source of increased distortion to be due to lexical variation and not speaker differences. As such, it may be necessary to consider alternative signal representations to the whitened MFCCs used in this work.

Some examples of signal representations and techniques that can reduce variability due to speaker physiology include the Mellin transform [54, 23] and vocal tract length normalization [64, 65]. Beyond signal representation issues however, we must also address speaker specific characteristics that exist at a more abstract level, such as pronunciation and idiolectal variation.

8.2.2 Clustering and Identification

The ideas presented in this section concern possible improvements and extensions to the clustering and identification algorithms presented and evaluated in Chapters 5 and 6.

Interval based clustering

In Chapter 5, we showed an example of an impure cluster and explained how accidental merging of lexically different clusters can occur as a result of 'chained' multi-word phrases, or matched subword units such as 'tion'. A potential solution for this problem may be to use time intervals as nodes, rather than time indices. This approach would allow a hierarchical representation of a particular speech segment. Taking the example from Figure 5-9, a word such as "imagination" may consist of three nodes, the first covering the entire word, the second covering just the word root "imagine", and the third covering the suffix "tion". Instead of considering any overlapping path fragment to be an edge for a particular node in the resulting adjacency graph, only path fragments that have significant agreement with the node's time interval would be allowed. In this manner, we might expect to see the creation of separate clusters for each of the constituent parts of the word "imagination", and a decrease in the number of accidental merges such as the example "imagination" and "nine eleven" being joined through the node containing "imagination of nine eleven".

Iterative refinement of node endpoints

Another possible approach to improving cluster purity and finding more precise word boundaries is to adopt an iterative approach to cluster formation, as shown

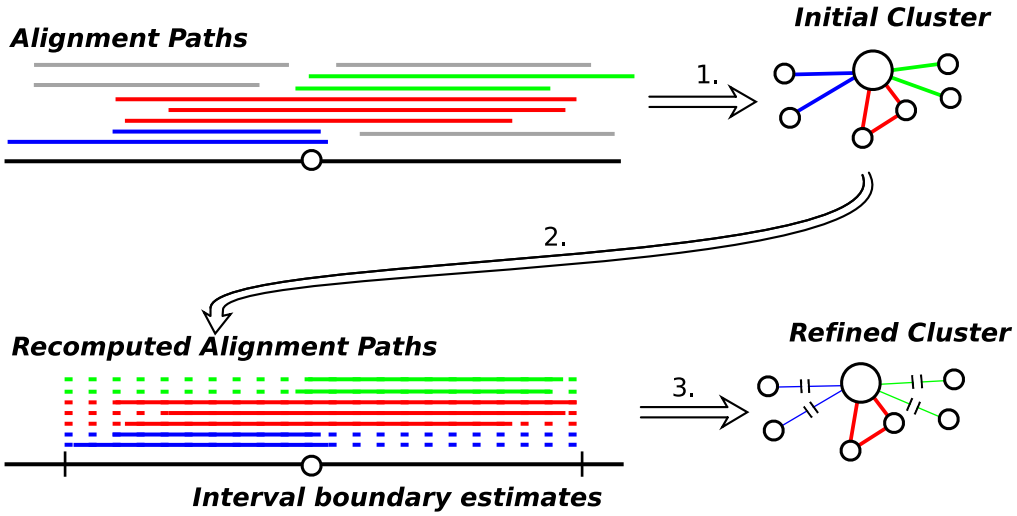


Figure 8-2: Cluster refinement using path re-estimation. 1. Alignment paths are used to generate an initial clustering as described in the original word acquisition algorithm. 2. Interval end-points for each node are estimated based upon the alignment paths present in the current cluster. These new end-points are used to re-calculate alignment path distortions. 3. The new distortions are used to re-cluster the original graph.

in Figure 8-2. After clusters have been formed and the time intervals for each node have been estimated, edge weights between cluster nodes can be recomputed using the start and end times of the node intervals as constraints. Based on these new edge weights, nodes can be rejected from the cluster and the time intervals can be re-estimated, with the process continuing until convergence to a final set of nodes. The idea behind this approach is to eliminate chaining and partial match errors by forcing clusters to be generated based on distortions that are computed over a consistent set of speech intervals.

Identification of Multi-word Phrases

One of the limitations of the cluster identification technique that we described in Chapter 5 was the inability of the designed recognition network to accurately identify multi-word phrases. Since roughly half of the clusters found in any particular lecture were observed to correspond to these multi-word phrases, this represents a significant area for improvement.

There are two primary methods that we propose to improve identification performance for this type of scenario. The first approach is to attempt to convert the multi-word clusters into single word clusters by using the path fragments that are *not* edges in the cluster to find word boundaries within the interval computed for a particular node. This strategy relies on the premise that the

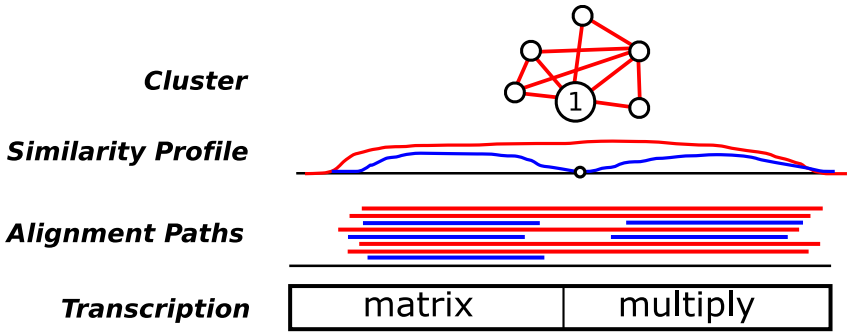


Figure 8-3: Separation of multi-word clusters using non-cluster alignment paths. The red alignment paths represent paths which are included in the clustering. Blue alignment paths represent paths which are excluded from the clustering. By removing the cluster paths, the similarity profile for node interval will change, potentially allowing us to find additional segmentations of the original interval.

similarity profile for a particular time index will be significantly different with the removal of a set of path fragments, and that the multi-word phrase contains words that occur separately elsewhere in the lecture. Once the node time intervals can be separated into their constituent words, then single word clusters can be extracted from the original clusters, after which we can apply original cluster identification algorithm.

Another approach to cluster identification is to retain the structure of the cluster itself, but change the recognition strategy. Instead of using an isolated word recognizer, we can use a continuous recognizer with an unconstrained grammar to produce a word lattice for each node's time interval. The word hypothesis for the cluster can then be recovered by enumerating and scoring the multiword paths in a similar manner to Steps 2-4 of the isolated word approach.

Global N-Best Reranking

The N-best lists generated during the cluster identification procedure, for many of the single-word clusters, contained the correct word. However, we still evaluate each cluster independently of the others generated for the lecture. One unexplored possibility for re-ranking the N-best lists is to exploit the global semantic relationship between the different words that occur within the lecture. Since the clusters are all drawn from the same lecture, it is likely that the underlying words have a high degree of co-occurrence in similar documents. For example, the words "matrix", "vector", and "eigenvalue", occur as clusters in the algebra lecture, but also in linear algebra textbooks and other related documents. As such, a better global set of cluster identities may be determined by finding the set of identities that jointly maximize co-occurrence in auxiliary data.

Multilingual experiments

Throughout our discussion of word acquisition using segmental DTW, our word identification strategies have been mostly language independent. An interesting area of study would be to examine the applicability of our word discovery and clustering methods on non-English speech. For languages where the word morphologies are considerably more complex than English, such as Arabic, acoustic determination of lexical entities may be a suitable approach for finding word roots. Likewise, in languages such as German, where word compounding occurs frequently, our pattern matching approach may be able to find the constituent words in these compound entities.

8.2.3 Applications

In this section, we propose several ways in which the segmental DTW algorithm and subsequent clustering procedure can be incorporated into existing speech recognition systems, or be used for novel applications on its own.

ASR Integration

One of the major possible applications for our unsupervised clustering algorithm is as a complementary source of information for traditional automatic speech recognition systems. Since most speech recognizers process each utterance independently of one another, they typically do not take advantage of the consistency of realization of the same word across multiple utterances. Alignment paths generated by segmental DTW can be used to point out locations where an automatic transcription is not consistent by indicating where two acoustically similar segments produced different transcriptions. In Table 8.1, we show several situations where this approach may be able to correct such errors. Several nodes are shown from some clusters generated for the second ASR lecture processed in Chapter 5, with the automatic speech recognition hypothesis generated when using unadapted speaker independent models with a general purpose vocabulary and language model. Each of the nodes for a particular cluster covers the same reference word.

In the first set of nodes, taken from cluster 3, the correct transcription is "fourier transform". Since "fourier" is out of vocabulary for this recognizer, we can not expect to achieve the correct hypothesis for the first part of this term. However, the automatically generated hypotheses exhibit considerable variation even for the second term, with word sequences such as "it has on", "chance are", "each has one", and "chancellor", being subsituted for the actual word, which is "transform". Likewise, in the second set of nodes, phrases like "care for steak" are substituted for "characteristic". When evaluated independently, these hypotheses may be the most reasonable interpretation of the available acoustics,

Reference	Node	Time	Hypothesis
Cluster 3 "fourier transform"	38	2:11	if we transform
	579	30:10	for you transform
	580	30:12	for each and so
	609	31:59	for a transformed
	611	32:06	free transform
	615	32:43	for each has one
	645	33:56	afford it has on
	652	34:08	if we chance are
	787	40:46	free chancellor
	801	41:36	free transform
Cluster 4 "characteristic"	673	35:27	character state
	800	41:33	characteristic
	1272	1:08:00	care for steak
	1352	1:11:57	characteristic
Cluster 35 "equations"	1329	1:10:42	equations
	1330	1:10:47	depression
	1331	1:10:49	equations

Table 8.1: Examples of automatic speech recognition hypotheses for some cluster nodes in ASR Lecture 2. All of the nodes within each category have the same underlying reference, which is shown in the lefthand column.

but by providing the information that each of the nodes refer to the same word, we can hope to recover from these errors by using the joint hypotheses from the other nodes in the cluster.

Vocabulary Initialization

A potential application for the cluster identification procedure disussed in Chapter 6 is as a tool for vocabulary initialization for speech recognition. For lecture transcription tasks, appropriate selection of the recognizer vocabulary and language model training data is important for obtaining accurate speech recognition outputs [87]. Without prior knowledge of the lecture topic, a general purpose, large vocabulary speech recognizer must be used in order to ensure coverage of the possible words used in the lecture. For similar reasons, the language model will typically not be adapted towards any particular topic or word usage patterns. By generating and identifying clusters prior to transcription, the cluster identities can be used to form a rough idea of the lecture topic or to provide a keyword-based summary of the lecture. More importantly, these cluster identities can be used as queries in a web search that would also allow us to find supplementary documents that can be used initialize a vocabulary and obtain topic specific language model data.

To illustrate the power of such auxiliary knowledge, we performed a web search with the keywords corresponding to the 3 most dense clusters for the Friedman lecture. These clusters corresponded to the queries: "search engine optimizer", "knowledge and work", and "toshiba laptop". The resulting two matches were: a transcript of the lecture itself, independently provided by a Chinese educational foundation, and the full text of an electronic copy of the book upon which the lecture is based, *The World is Flat*. Although this is obviously an extreme example, it demonstrates the utility of the web as a resource for finding textual material relevant to the audio stream being processed.

Audio navigation interface

In Chapters 4 and 5, we illustrated how path fragments generated across the lecture serve to link speech segments from one temporal location to another. These fragments can be discovered in an unsupervised fashion and represent a unique way to navigate through a lecture or audio stream. A user listening to the audio stream can use the paths to jump to other locations in the audio where the same word or phrase occurs. For example, if the user hears the phrase "gauss-jordan elimination" during a linear algebra lecture and wants to hear about its application to a problem, or a more comprehensive definition of the term, then they can choose to navigate to places in the audio stream that are referenced by the path fragments covering that particular phrase. If a graphical interface is available, more specific matches could be obtained by allowing the user to explicitly specify a desired time interval to match, in which case only path fragments with a good fit to that time interval would be returned.

Appendix A

Cluster Tables

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**
1	72	0.009	charge	70.8	27	4	0.125	times a	75.0
2	65	0.010	distribute	16.9	28	4	0.110	direction	100.0
3	63	0.074	electric	96.8	29	4	0.110	flow out	100.0
4	35	0.037	surface	97.1	30	4	0.108	sphere	100.0
5	32	0.021	inside	90.6	31	4	0.089	plane	50.0
6	29	0.065	epsilon	75.9	32	3	0.308	sphere	100.0
7	18	0.032	plate	50.0	33	3	0.258	four pi r squared	100.0
8	15	0.063	r squared	86.7	34	3	0.191	square	100.0
9	12	0.060	argument	100.0	35	3	0.176	indpendent	100.0
10	12	0.046	d a	100.0	36	3	0.174	i want to know	100.0
11	11	0.044	distance	54.5	37	3	0.174	to	100.0
12	10	0.060	exactly	70.0	38	3	0.160	situation	100.0
13	8	0.057	infinitely	87.5	39	3	0.158	the normal	100.0
14	7	0.054	outside	57.1	40	3	0.155	that plane	100.0
15	6	0.093	result	100.0	41	3	0.153	density	100.0
16	6	0.092	vandegraaff	83.3	42	3	0.147	concentric	66.7
17	6	0.089	e vector	66.7				spheres	
18	6	0.086	field lines	100.0	43	3	0.140	gauss law	100.0
19	5	0.262	with my finger	100.0	44	3	0.140	means there	100.0
20	5	0.107	like this	100.0	45	3	0.139	and so now	100.0
21	5	0.097	calculate	100.0	46	3	0.124	is sigma	100.0
22	5	0.086	charge	100.0	47	3	0.118	electro	100.0
23	5	0.085	flux	100.0	48	3	0.117	be the same	100.0
24	4	0.201	rectangle	100.0	49	3	0.115	symmetry	100.0
25	4	0.191	four pi	100.0	50	3	0.108	calculate	100.0
26	4	0.167	angle theta	100.0	51	3	0.105	radius	66.7

Table A.1: Clusters generated for Physics lecture.

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**
1	57	0.051	matrix	96.5	22	4	0.132	inverse	100.0
2	44	0.038	matrix	88.6	23	4	0.126	parentheses	75.0
3	20	0.031	three	75.0	24	4	0.122	inverse	100.0
4	20	0.030	elimination	95.0	25	4	0.121	position	75.0
5	17	0.065	matrices	100.0	26	4	0.098	one of	100.0
6	16	0.049	subtract	93.8	27	4	0.078	the multiplier	100.0
7	15	0.049	equations	93.3	28	3	0.226	plus	100.0
8	12	0.088	matrix	100.0	29	3	0.177	zero	100.0
9	11	0.058	first	90.9	30	3	0.162	extra column	100.0
10	10	0.079	right hand side	100.0	31	3	0.153	suppose	100.0
11	9	0.110	substitution	77.8	32	3	0.138	the same	100.0
12	9	0.066	exchange	77.8	33	3	0.129	this	100.0
13	9	0.047	multiply	100.0	34	3	0.125	two minus	100.0
14	7	0.073	times	100.0	35	3	0.123	the next step	100.0
15	7	0.055	six	42.9	36	3	0.114	operations	100.0
16	7	0.054	pivot	71.4	37	3	0.108	pivot zero	100.0
17	6	0.062	row two	100.0	38	3	0.105	purpose	100.0
18	5	0.099	determinant	100.0	39	3	0.097	what	100.0
19	5	0.085	and	100.0	40	3	0.088	and	100.0
20	5	0.063	minus	100.0	41	3	0.087	is this	100.0
21	4	0.164	in this case	100.0					

Table A.2: Clusters generated for Linear Algebra lecture.

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**
1	55	0.005	that	18.2	33	5	0.066	-	0.0
2	52	0.013	frequency	84.6	34	5	0.064	time	100.0
3	35	0.033	fourier transform	91.4	35	5	0.060	equations	80.0
4	30	0.026	characteristic	80.0	36	5	0.058	infinite	60.0
5	30	0.017	function	70.0	37	5	0.055	boundary	60.0
6	29	0.058	vocal tract	96.6				condition	
7	17	0.024	vocal	94.1	38	5	0.055	oftentimes	80.0
8	17	0.019	for example	76.5	39	5	0.049	sound in	100.0
9	16	0.034	cavity	75.0	40	5	0.049	like this	100.0
10	16	0.031	spectrogram	62.5	41	5	0.047	multiply	60.0
11	14	0.030	american english	100.0	42	5	0.040	what wavelength	60.0
12	13	0.040	speech production	76.9				exactly	
13	11	0.043	basically	100.0	43	5	0.039	talk about	60.0
14	11	0.027	sound	100.0	44	4	0.177	followed by	100.0
15	8	0.064	what	100.0	45	4	0.104	understand im	100.0
16	7	0.069	propagation	100.0				saying	
17	7	0.051	waveform	85.7	46	4	0.095	the window	100.0
18	7	0.045	acoustic	85.7	47	4	0.093	vocal folds	100.0
19	7	0.041	-	71.4	48	4	0.092	expression	100.0
20	7	0.037	related to	57.1	49	4	0.091	acoustic	75.0
21	7	0.034	velocity	71.4	50	4	0.090	opening	75.0
22	6	0.088	waveform	100.0	51	4	0.082	perfectly	75.0
23	6	0.067	generate	100.0				periodic	
24	6	0.066	constriction	100.0	52	4	0.072	does that mean	100.0
25	6	0.059	source	100.0	53	4	0.068	proportional to	75.0
26	6	0.059	the source	50.0	54	4	0.059	the second	50.0
27	6	0.057	characterize	100.0				formant	
28	6	0.047	over here	100.0	55	4	0.057	-	0.0
29	6	0.031	it turns out	66.7	56	4	0.054	to introduce	100.0
30	5	0.180	seven	100.0	57	4	0.054	between	100.0
31	5	0.095	stop consonants	100.0	58	4	0.051	solve the problem	100.0
32	5	0.077	pieces of	100.0	59	4	0.046	-	0.0

Continued on Next Page...

Table A.3 – Continued

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**
60	3	0.208	electrical engineer	100.0	76	3	0.102	hand side	100.0
					77	3	0.101	speech	100.0
61	3	0.171	try to	100.0	78	3	0.101	the length of the tube	100.0
62	3	0.163	this guy	100.0					
63	3	0.147	rib cage	100.0	79	3	0.095	you see that	100.0
64	3	0.143	equation	100.0	80	3	0.093	nasal	100.0
65	3	0.136	and then	100.0	81	3	0.091	rectangular window	100.0
66	3	0.129	consonant	66.7					
67	3	0.123	you come	100.0	82	3	0.090	configuration	100.0
68	3	0.121	that were the case	100.0	83	3	0.089	very important	100.0
					84	3	0.087	anatomical	100.0
69	3	0.120	the third formant	100.0	85	3	0.082	something we call	100.0
70	3	0.115	sinusoids	100.0	86	3	0.079	the	100.0
71	3	0.113	waveform	100.0	87	3	0.077	eventually	100.0
72	3	0.112	people	66.7	88	3	0.068	going to	100.0
73	3	0.107	speech recognition	100.0	89	3	0.063	sound	66.7
					90	3	0.062	one dimension	100.0
74	3	0.105	intuitions	100.0	91	3	0.060	tongue body	100.0
75	3	0.105	phoneme	100.0	92	3	0.059	zeros	100.0

Table A.3: Clusters generated for ASR Lecture 2

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**
1	128	0.013	cluster	96.9	31	6	0.077	example	83.3
2	78	0.018	data points	75.6	32	5	0.149	k equals three	80.0
3	64	0.023	cluster	90.6	33	5	0.139	interesting	100.0
4	54	0.035	distance	90.7	34	5	0.128	update	100.0
5	43	0.064	distortion	100.0	35	5	0.114	code word	100.0
6	27	0.035	coefficients	77.8	36	5	0.104	sub i	60.0
7	26	0.026	speech	34.6	37	5	0.104	dependent	80.0
			recognition		38	5	0.085	first	100.0
8	23	0.103	vector	91.3	39	4	0.189	robustness	100.0
			quantization		40	4	0.189	any question	100.0
9	19	0.044	probability	68.4				about	
10	18	0.051	samples	61.1	41	4	0.176	merge together	100.0
11	17	0.044	result	88.2	42	4	0.175	squared error	100.0
12	15	0.046	k means	86.7	43	4	0.152	unseen data	100.0
13	14	0.060	test data	100.0	44	4	0.148	transition	100.0
14	14	0.056	dimensions	50.0	45	4	0.141	got stuck	100.0
15	13	0.051	one of the things	69.2	46	4	0.134	new data	100.0
16	12	0.070	iteration	100.0	47	4	0.132	speech	100.0
17	10	0.040	talk about	100.0	48	4	0.123	based on	100.0
18	9	0.078	dendrogram	100.0	49	4	0.116	a good thing	100.0
19	9	0.073	together	77.8	50	4	0.112	example	100.0
20	8	0.125	criterion	87.5	51	4	0.110	useful	100.0
21	8	0.115	the speech signal	100.0	52	4	0.102	talking about	100.0
22	8	0.085	distance	100.0	53	4	0.094	nineteen eighties	100.0
23	8	0.063	assign	75.0	54	3	0.282	transcription	100.0
24	7	0.127	quantize	100.0	55	3	0.268	the very first	100.0
25	7	0.124	represent	85.7	56	3	0.266	percent	100.0
26	7	0.122	distribution	100.0	57	3	0.248	it turns out	100.0
27	7	0.072	assignment	100.0	58	3	0.246	codebook	100.0
28	6	0.176	average value	100.0	59	3	0.241	you can see	100.0
29	6	0.096	probabilistic	66.7	60	3	0.227	guarantee	100.0
30	6	0.093	prototype	100.0	61	3	0.223	in the literature	100.0

Continued on Next Page...

Table A.4 – Continued

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**
62	3	0.214	very popular	100.0	75	3	0.143	ten twenty four	100.0
63	3	0.182	biggest variance	66.7	76	3	0.143	different	100.0
64	3	0.177	process	100.0	77	3	0.139	k means	100.0
65	3	0.172	robust	100.0	78	3	0.139	the size	66.7
66	3	0.171	acoustic model	100.0	79	3	0.135	depending on	100.0
67	3	0.169	local optimum	100.0	80	3	0.133	initialize	100.0
68	3	0.168	the overall	100.0	81	3	0.130	value	100.0
69	3	0.168	you merge	100.0	82	3	0.128	criterion	100.0
70	3	0.163	structure	100.0	83	3	0.126	between	100.0
71	3	0.161	people	100.0	84	3	0.123	another example	100.0
72	3	0.158	that you get	100.0	85	3	0.122	result	100.0
73	3	0.149	more compact	100.0	86	3	0.120	that you have	100.0
74	3	0.146	plot	100.0	87	3	0.119	so	100.0

Table A.4: Clusters generated for ASR Lecture 6

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**
1	137	0.013	speaker	70.8	31	4	0.297	this term goes here	100.0
2	109	0.020	adaptation	93.6					
3	43	0.019	vector	88.4	32	4	0.107	-	0.0
4	25	0.043	likelihood	96.0	33	4	0.284	density	100.0
5	25	0.022	parameters	100.0	34	4	0.167	gaussian mixture model	100.0
6	19	0.032	observation	42.1					
7	15	0.068	recognize	93.3	35	4	0.194	resource management	100.0
8	13	0.077	characteristic	69.2					
9	11	0.047	turns out that	81.8	36	4	0.089	data	100.0
10	10	0.056	principal component	90.0	37	4	0.165	mth frame	100.0
					38	4	0.144	second formant	100.0
11	10	0.077	recognition	80.0	39	4	0.153	vowel	100.0
12	9	0.096	reference speaker	88.9	40	4	0.130	variance	100.0
13	9	0.077	the speech	77.8	41	4	0.091	different channel	75.0
14	8	0.075	specific	100.0	42	3	0.167	things that you got	100.0
15	8	0.062	models	100.0					
16	8	0.058	unsupervised	87.5	43	3	0.122	this in particular	100.0
17	7	0.074	speaker cluster	100.0					
18	7	0.110	like this	85.7	44	3	0.095	you could	100.0
19	7	0.083	speakers	100.0	45	3	0.122	combination	100.0
20	7	0.078	vocal tract	100.0	46	3	0.133	this technique	66.7
21	6	0.074	classes	100.0	47	3	0.132	because	100.0
22	6	0.070	point four	83.3	48	3	0.103	techniques	100.0
23	6	0.059	this is	83.3	49	3	0.129	data	100.0
24	6	0.106	variation	100.0	50	3	0.151	have enough data	100.0
25	6	0.059	typically	66.7	51	3	0.141	the nice thing about	100.0
26	5	0.094	reduction	100.0					
27	5	0.097	transformation	100.0	52	3	0.170	an example	100.0
28	5	0.090	warping	100.0	53	3	0.170	estimate	100.0
29	4	0.125	interpolation	100.0	54	3	0.184	units	100.0
30	4	0.110	you see	100.0	55	3	0.147	training data	100.0

Continued on Next Page...

Table A.5 – Continued

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	**Transcription**	**Purity**
56	3	0.145	utterances	100.0	61	3	0.180	likelihood	100.0
57	3	0.265	speech signal	100.0				contour	
58	3	0.171	best describes	66.7	62	3	0.140	normalization	100.0
59	3	0.254	as possible	100.0	63	3	0.119	this is typically	100.0
60	3	0.116	these	100.0					

Table A.5: Clusters generated for ASR Lecture 19

Appendix B

IWR Cluster Identification Results

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	Reference	Hypothesis	Rank	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	Reference	Hypothesis	Rank
1	55	that	reticulates	-	32	5	pieces of	mystification	-
2	52	frequency	infrequency	2	33	5	-	inset	-
3	35	fourier transform	fleetness	8	34	5	time	fierstein	-
4	30	characteristic	characteristic	1	35	5	equations	syndication	-
5	30	function	suppression	4	36	5	infinite	disintermediate	-
6	29	vocal tract	protract	-	37	5	boundary condition	fricative	-
7	17	vocal	roccaforte	10					
8	17	for example	resample	4	38	5	oftentimes	oftentimes	1
9	16	cavity	cavity	1	39	5	sound in	antibeach	-
10	16	spectrogram	spectrogram	1	40	5	like this	fleitas	-
11	14	american english	americanization	4	41	5	multiply	mortified	2
12	13	speech production	retraction	-	42	5	what wavelength exactly	rectify	-
13	11	basically	basically	1					
14	11	sound	nightstand	6	43	5	talk about	manful	-
15	8	what	what	1	44	4	followed by	phillippi	-
16	7	propagation	propagation	1	45	4	understand im saying	inseminating	-
17	7	waveform	waveform	1					
18	7	acoustic	eucharist	-	46	4	the window	window	1
19	7	-	pinkest	-	47	4	vocal folds	kofler	-
20	7	related to	reflated	2	48	4	expression	suppression	-
21	7	velocity	velocity	1	49	4	acoustic	acoustical	2
22	6	waveform	waveform	1	50	4	opening	inane	4
23	6	generate	generate	1	51	4	perfectly periodic	interjected	-
24	6	constriction	constriction	1					
25	6	source	resources'	-	52	4	does that mean	lisanti	-
26	6	the source	issuers'	-	53	4	proportional to	proportionate	4
27	6	characterize	characterizing	1	54	4	the second formant	effectively	-
28	6	over here	overseer	-					
29	6	it turns out	curacao	-	55	4	-	mckevitt	-
30	5	seven	sevens	1	56	4	to introduce	reintroduced	-
31	5	stop consonants	tomkinson	-	57	4	between	between	1

Continued on Next Page...

Table B.1 – Continued

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**
58	4	solve the problem	suffolk	-	75	3	phoneme	phoneme	1
59	4	-	ellestad	-	76	3	hand side	ironside	-
60	3	electrical	retrograding	-	77	3	speech	eurospeech	5
		engineer			78	3	the length of the	leibowitz	-
61	3	try to	sweetheart	-			tube		
62	3	this guy	despite	-	79	3	you see that	yasuda	-
63	3	rib cage	birdcage	-	80	3	nasal	nasals	1
64	3	equation	recreation	-	81	3	rectangular	rekindling	2
65	3	and then	anderton	-			window		
66	3	consonant	consonant	1	82	3	configuration	configuration	1
67	3	you come	jacobowitz	-	83	3	very important	reimport	-
68	3	that were the	fabricates	-	84	3	anatomical	anatomical	1
		case			85	3	something we call	suchanek	4
69	3	the third formant	interpharm	-	86	3	the	inhibiting	2
70	3	sinusoids	sinusoid	1	87	3	eventually	evangelize	-
71	3	waveform	waveform	1	88	3	going to	schenectady	-
72	3	people	politically	-	89	3	sound	hisao	-
73	3	speech	interconnection	2	90	3	one dimension	undulation	-
		recognition			91	3	tongue body	somebody	-
74	3	intuitions	intuitions	1	92	3	zeros	zeros	1

Table B.1: IWR cluster identification results for ASR Lecture 2.

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**
1	128	cluster	clusters	1	30	6	prototype	prototype	1
2	78	data points	datapoint	1	31	6	example	zempel	5
3	64	cluster	clustering	1	32	5	k equals three	upholstering	-
4	54	distance	disincentive	7	33	5	interesting	introspecting	2
5	43	distortion	distortion	1	34	5	update	update	1
6	27	coefficients	coefficients	1	35	5	code word	codewords	1
7	26	speech recognition	interpolation	-	36	5	sub i	baie	-
					37	5	dependent	content	3
8	23	vector quantization	radicalization	-	38	5	first	first	1
					39	4	robustness	robustness	1
9	19	probability	probability	1	40	4	any question about	questionable	3
10	18	samples	samples	1					
11	17	result	results	1	41	4	merge together	marcinek	-
12	15	k means	atamian	-	42	4	squared error	mosquera	2
13	14	test data	testator	-	43	4	unseen data	unseeded	7
14	14	dimensions	dimensioned	1	44	4	transition	transitioned	1
15	13	one of the things	wanting	-	45	4	got stuck	benstock	-
16	12	iteration	generation	5	46	4	new data	noodle	-
17	10	talk about	talkable	-	47	4	speech	speech	1
18	9	dendrogram	dendrogram	1	48	4	based on	beton	5
19	9	together	appointing	3	49	4	a good thing	goetting	-
20	8	criterion	criteria	1	50	4	example	unexampled	3
21	8	the speech signal	spitznagel	-	51	4	useful	useful	1
22	8	distance	distances	3	52	4	talking about	contibel	-
23	8	assign	assigned	1	53	4	nineteen eighties	nightingale	8
24	7	quantize	quantize	1	54	3	transcription	transcription	1
25	7	represent	representable	2	55	3	the very first	sublimater	-
26	7	distribution	distribution	1	56	3	percent	percent	1
27	7	assignment	symons	3	57	3	it turns out	editors'	-
28	6	average value	averaged	1	58	3	codebook	codebooks	1
29	6	probabilistic	probabilist	2	59	3	you can see	eudosia	-

Continued on Next Page...

Table B.2 – Continued

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**
60	3	guarantee	guaranteed	1	74	3	plot	applaud	-
61	3	in the literature	literature	1	75	3	ten twenty four	centanni	-
62	3	very popular	depopulate	2	76	3	different	settled	-
63	3	biggest variance	antedated	-	77	3	k means	kamins	-
64	3	process	processed	1	78	3	the size	decidable	-
65	3	robust	robust	1	79	3	depending on	dependable	2
66	3	acoustic model	cristobal	7	80	3	initialize	initialize	1
67	3	local optimum	clopton	-	81	3	value	value	1
68	3	the overall	stayover	-	82	3	criterion	criterion	1
69	3	you merge	feuerstein	-	83	3	between	between	1
70	3	structure	structured	1	84	3	another example	another's	1
71	3	people	people	1	85	3	result	resulted	1
72	3	that you get	thudded	-	86	3	that you have	yeuhi	5
73	3	more compact	noncontact	-	87	3	so	so	1

Table B.2: IWR cluster identification results for ASR Lecture 6.

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**
1	137	speaker	spieker	1	31	4	this term goes here	sternberger	-
2	109	adaptation	annotation	2					
3	43	vector	antimatter	9	32	4	-	dissuade	-
4	25	likelihood	likely	6	33	4	density	density	1
5	25	parameters	parameters	1	34	4	gaussian mixture model	internationale	-
6	19	observation	estimations	2					
7	15	recognize	recognizer	1	35	4	resource management	reassessment	4
8	13	characteristic	characteristic	1					
9	11	turns out that	resounded	-	36	4	data	dative	2
10	10	principal component	slipons	-	37	4	mth frame	entranced	-
					38	4	second formant	secondquarter	-
11	10	recognition	recognition	1	39	4	vowel	nowels	7
12	9	reference speaker	speculating	5	40	4	variance	variance	1
13	9	the speech	inspeech	2	41	4	different channel	indifferent	-
14	8	specific	specific	1	42	3	things that you got	exited	-
15	8	models	supermodels	2					
16	8	unsupervised	uncivilized	4	43	3	this in particular	mistaking	-
17	7	speaker cluster	cederquist	5					
18	7	like this	midas	-	44	3	you could	seconded	-
19	7	speakers	speakers	1	45	3	combination	commination	6
20	7	vocal tract	olczak	-	46	3	this technique	mistaking	-
21	6	classes	classes'	1	47	3	because	speakers	8
22	6	point four	repoints	-	48	3	techniques	techniques	1
23	6	this is	lisenby	-	49	3	data	beta	4
24	6	variation	variation	1	50	3	have enough data	annotated	-
25	6	typically	intuitively	-	51	3	the nice thing about	aycinena	-
26	5	reduction	reductionist	2					
27	5	transformation	transformational	2	52	3	an example	example	1
28	5	warping	artifact	3	53	3	estimate	estimates	1
29	4	interpolation	interpolation	1	54	3	units	units'	1
30	4	you see	idiocy	-	55	3	training data	trained	-

Continued on Next Page...

Table B.3 – Continued

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**
56	3	utterances	renounces	-	61	3	likelihood	cyclotrons	4
57	3	speech signal	spitznagel	2			contour		
58	3	best describes	estis	6	62	3	normalization	normalization	1
59	3	as possible	possible	1	63	3	this is typically	systemic	-
60	3	these	eastep	-					

Table B.3: IWR cluster identification results for ASR Lecture 19.

Appendix C

DBS Cluster Identification Tables

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**
1	55	that	affrication	-	32	5	pieces of	tissue	-
2	52	frequency	africans	-	33	5	-	used	-
3	35	fourier transform	transform	1	34	5	time	endtimer	7
4	30	characteristic	interesting	-	35	5	equations	abrasions	-
5	30	function	anshan	-	36	5	infinite	ubicom	-
6	29	vocal tract	retract	2	37	5	boundary condition	conditions	1
7	17	vocal	shackelford	-					
8	17	for example	example	1	38	5	oftentimes	lafontaine	2
9	16	cavity	calve	-	39	5	sound in	jaeyeon	-
10	16	spectrogram	spectrogram	1	40	5	like this	biked	-
11	14	american english	americanism	10	41	5	multiply	octopi	5
12	13	speech production	production	1	42	5	what wavelength	exact	1
13	11	basically	basically	1			exactly		
14	11	sound	leicht	-	43	5	talk about	about	1
15	8	what	happen	-	44	4	followed by	follow	1
16	7	propagation	propagation	1	45	4	understand im	yunis	-
17	7	waveform	reform	-			saying		
18	7	acoustic	kristy	-	46	4	the window	window	1
19	7	-	speakings	-	47	4	vocal folds	fold	1
20	7	related to	tooth	-	48	4	expression	expression	1
21	7	velocity	velocities	1	49	4	acoustic	stick	-
22	6	waveform	sway	1	50	4	opening	apennine	2
23	6	generate	generate	1	51	4	perfectly	periodically	2
24	6	constriction	constrained	-			periodic		
25	6	source	sourcing	1	52	4	does that mean	butthead	-
26	6	the source	sourced	1	53	4	proportional to	proportion	2
27	6	characterize	characterize	1	54	4	the second	kalla	-
28	6	over here	which	-			formant		
29	6	it turns out	turns	1	55	4	-	wanton	-
30	5	seven	seven	1	56	4	to introduce	studer	-
31	5	stop consonants	consonant	1	57	4	between	beat	-

Continued on Next Page...

Table C.1 – Continued

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**
58	4	solve the problem	prab	3	75	3	phoneme	whinnying	-
59	4	-	plosive	-	76	3	hand side	incite	-
60	3	electrical	granges	2	77	3	speech	beech	-
		engineer			78	3	the length of the	ditch	-
61	3	try to	triton	-			tube		
62	3	this guy	fact	-	79	3	you see that	heed	-
63	3	rib cage	ripka	-	80	3	nasal	solves	-
64	3	equation	ukraine	-	81	3	rectangular	erecting	2
65	3	and then	pandan	-			window		
66	3	consonant	const	2	82	3	configuration	configuration	1
67	3	you come	cucumber	6	83	3	very important	import	2
68	3	that were the	flatware	2	84	3	anatomical	structural	-
		case			85	3	something we call	something	1
69	3	the third formant	actin	-	86	3	the	beginning	-
70	3	sinusoids	lasorda	-	87	3	eventually	invention	-
71	3	waveform	form	-	88	3	going to	ellington	-
72	3	people	peep	2	89	3	sound	scriver	-
73	3	speech	reckoning	-	90	3	one dimension	dumais	-
		recognition			91	3	tongue body	widebody	2
74	3	intuitions	intuitions	1	92	3	zeros	fierro	-

Table C.1: DBS cluster identification results for ASR Lecture 2.

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**
1	128	cluster	clusters	1	30	6	prototype	prototype	1
2	78	data points	datapoint	1	31	6	example	example	1
3	64	cluster	string	-	32	5	k equals three	equals	1
4	54	distance	distance	1	33	5	interesting	interesting	1
5	43	distortion	distortion	1	34	5	update	update	1
6	27	coefficients	efficiency	-	35	5	code word	coaled	10
7	26	speech	expectation	-	36	5	sub i	baie	-
		recognition			37	5	dependent	independence	-
8	23	vector	quantization	1	38	5	first	first	1
		quantization			39	4	robustness	robots	-
9	19	probability	probability	1	40	4	any question	about	1
10	18	samples	samples	1			about		
11	17	result	results	1	41	4	merge together	together	1
12	15	k means	mccamy	9	42	4	squared error	squared	1
13	14	test data	testator	-	43	4	unseen data	indata	2
14	14	dimensions	mentions	4	44	4	transition	transitions	1
15	13	one of the things	wanna	6	45	4	got stuck	staack	-
16	12	iteration	iteration	1	46	4	new data	uday	4
17	10	talk about	walkabout	4	47	4	speech	bait	-
18	9	dendrogram	dendrogram	1	48	4	based on	based	1
19	9	together	together	1	49	4	a good thing	haygood	3
20	8	criterion	criteria	2	50	4	example	unexampled	2
21	8	the speech signal	speech	1	51	4	useful	deduced	-
22	8	distance	distances	1	52	4	talking about	stocking	-
23	8	assign	assign	1	53	4	nineteen eighties	nineteen	1
24	7	quantize	unties	2	54	3	transcription	transcription	1
25	7	represent	represented	1	55	3	the very first	firstier	2
26	7	distribution	distribution	1	56	3	percent	percent	1
27	7	assignment	synan	-	57	3	it turns out	turns	1
28	6	average value	average	1	58	3	codebook	books	-
29	6	probabilistic	probabilistic	1	59	3	you can see	conceal	-

Continued on Next Page...

Table C.2 – Continued

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**
60	3	guarantee	arendt	4	74	3	plot	plot	1
61	3	in the literature	badertscher	-	75	3	ten twenty four	twentyfold	2
62	3	very popular	popular	1	76	3	different	different	1
63	3	biggest variance	antedating	-	77	3	k means	canings	-
64	3	process	process	1	78	3	the size	sizes	1
65	3	robust	robots	-	79	3	depending on	deep	-
66	3	acoustic model	lustig	2	80	3	initialize	ruination	-
67	3	local optimum	optima	3	81	3	value	cabal	2
68	3	the overall	overall	1	82	3	criterion	mcwright	8
69	3	you merge	firstly	-	83	3	between	between	1
70	3	structure	struck	-	84	3	another example	example	1
71	3	people	people	1	85	3	result	result	1
72	3	that you get	fracture	-	86	3	that you have	yu-han	-
73	3	more compact	compact	1	87	3	so	-	

Table C.2: DBS cluster identification results for ASR Lecture 6.

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**
1	137	speaker	speaker	1	31	4	this term goes here	sturm	-
2	109	adaptation	meditation	-					
3	43	vector	vectors	1	32	4	-	zita	-
4	25	likelihood	wycliffe	-	33	4	density	density	1
5	25	parameters	pram	8	34	4	gaussian mixture model	gaussian	1
6	19	observation	observations	1					
7	15	recognize	wrecking	-	35	4	resource management	mismanagement	3
8	13	characteristic	characteristic	1					
9	11	turns out that	misters	-	36	4	data	eatonton	-
10	10	principal component	campanis	-	37	4	mth frame	franze	4
					38	4	second formant	conformance	-
11	10	recognition	wrecking	4	39	4	vowel	fouls	3
12	9	reference speaker	speaker	1	40	4	variance	rans	-
13	9	the speech	speech	1	41	4	different channel	different	1
14	8	specific	specific	1	42	3	things that you got	things	1
15	8	models	models	1					
16	8	unsupervised	revise	-	43	3	this in particular	mystical	-
17	7	speaker cluster	speaker	1					
18	7	like this	bats	-	44	3	you could	busman	-
19	7	speakers	speakers	1	45	3	combination	kamens	-
20	7	vocal tract	tract	1	46	3	this technique	stake	-
21	6	classes	classes	1	47	3	because	bakhash	2
22	6	point four	repoints	5	48	3	techniques	blakeney's	-
23	6	this is	this's	1	49	3	data	outta	-
24	6	variation	variation	1	50	3	have enough data	havin'	2
25	6	typically	typically	1	51	3	the nice thing about	singable	3
26	5	reduction	percent	-					
27	5	transformation	transformation	1	52	3	an example	example	1
28	5	warping	warping	1	53	3	estimate	housemate's	2
29	4	interpolation	interplay	2	54	3	units	unit	1
30	4	you see	jusino	-	55	3	training data	straying	3

Continued on Next Page...

Table C.3 – Continued

$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Reference**	**Hypothesis**	**Rank**
56	3	utterances	sincere	-	61	3	likelihood	contour	1
57	3	speech signal	spizzirri	2			contour		
58	3	best describes	pesticide	-	62	3	normalization	normalization	1
59	3	as possible	possible	1	63	3	this is typically	typically	1
60	3	these	articular	-					

Table C.3: DBS cluster identification results for ASR Lecture 19.

Bibliography

[1] J. Ajmera, I. McCowan, and H.Bourlard. Robust speaker change detection. *IEEE Signal Processing Letter*, 11(8):649–651, August 2004. 112, 113

[2] R. K. Ando and L. Lee. Mostly-unsupervised statistical segmentation of Japanese. In *First Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 241–248, 2000. 27

[3] K. Aono, K. Yasuda, T. Takezawa, S. Yamamoto, and M. Yanaida. Analysis and effect of speaking style for dialogue speech recognition. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 339–344, St. Thomas, U.S. Virgin Islands, December 2003. 22

[4] M. Bacchiani. *Speech Recognition System Design Based on Automatically Derived Units*. PhD thesis, Boston University, 1999. 26

[5] M. Bacchiani and M. Ostendorf. Using automatically-derived acoustic subword units in large vocabulary speech recognition. In *Proc. ICSLP*, volume 5, pages 1843–1846, Sydney, Australia, December 1998. 26

[6] L. Bahl, P. Brown, P. de Souza, R. Mercer, and M. Picheny. Acoustic Markov models used in the Tangora speech recognition system. In *Proc. ICASSP*, pages 497–500, New York, NY, April 1988. 26

[7] L.R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer. A new algorithm for the estimation of hidden Markov model parameters. In *Proc. ICASSP*, pages 493–496, New York, NY, April 1988. 42

[8] I. Bazzi. *Modelling Out-of-vocabulary words for robust speech recognition*. PhD thesis, Massachusetts Institute of Technology, 2002. 22

[9] I. Bazzi and J. Glass. Learning units for domain-independent out-of-vocabulary word modelling. In *Proc. Eurospeech*, pages 61–64, Aalborg, 2001. 100

[10] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995. 38

[11] L. Bloomfield. *Language*. Holt, New York, NY, 1933. 21

[12] A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert. Approaches to the automatic discovery of patterns in biosequences. *J. Computational Biology*, 5(2):279–305, 1998. 23

[13] M. Brent and T. Cartwright. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61:93–125, 1996. 27

[14] M. R. Brent. An efficient probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34(1-3):71–105, February 1999. 27

[15] E. Brill. *A Corpus-based approach to language learning*. PhD thesis, University of Pennsylvania, Philadelphia, PA, December 1993. 26

[16] C.J. Burges, D. Plastina, J.C. Platt, E. Renshaw, and H. Malvar. Using audio fingerprinting for duplicate detection and thumbnail generation. In *Proc. ICASSP*, volume 3, pages 9–12, Philadelphia, PA, March 2005. 25

[17] G. Carroll and E. Charniak. Two experiments on learning probabilistic dependency grammars from corpora. In C. Weir, S. Abney, R. Grishman, and R. Weischedel, editors, *Working Notes of the Workshop on Statistically-based NLP techniques*, pages 1–13. AAAI Press, Menlo Park, CA, 1992. 26

[18] Wei Chai and Barry Vercoe. Structural analysis of musical signals for indexing and thumbnailing. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 27–34, Washington, DC, USA, 2003. 25

[19] S.S. Chen and P.S. Gopalakrishnan. Clustering via the bayesian information criterion with applications in speech recognition. In *Proc. ICASSP*, pages 645–648, Seattle, WA, May 1998. 111

[20] S.S. Chen and P.S. Gopalakrishnan. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *Proc. DARPA Broadcast News Transcription Workshop*, pages 127–132, Lansdowne, VA, February 1998. 111, 113

[21] N.A. Chomsky. *Knowledge of Language: Its Nature, Origin, and Use*. Praeger, New York, NY, 1986. 21

[22] A. Clark. *Unsupervised Language Acquisition: Theory and Practice*. PhD thesis, University of Sussex, Brighton, UK, December 2001. 26

[23] L. Cohen. The scale representation. *IEEE Trans. on Signal Processing*, 41(12):3275–3291, December 1993. 135

[24] T. Colthurst, O. Kimball, F. Richardson, H. Shu, C. Wooters, R. Iyer, and H. Gish. The 2000 BBN Byblos LVCSR system. In *Proceedings of the DARPA Speech Transcription Workshop*, College Park, MD, May 2000. 22

[25] S. Crain. Language acquisition in the absence of experience. *Behavioral and Brain Sciences*, 14(4):601–699, December 1991. 21

[26] R. Dannenberg and N. Hu. Pattern discovery techniques for music audio. In *Proceedings of Int'l Conference on Music Information Retrieval*, Paris, France, October 2002. 25

[27] S.B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, August 1980. 38

[28] C.G. de Marcken. The unsupervised acquisition of a lexicon from continuous speech. Technical Report 1558, Massachusetts Institute of Technology Artificial Intelligence Laboratory, Cambridge, MA, November 1996. 27

[29] C.G. de Marcken. *Unsupervised Language Acquisition*. PhD thesis, Massachusetts Institute of Technology, 1996. 27

[30] P. Delacourt and C.J. Wellekens. DISTBIC: A speaker-based segmentation for audio data indexing. *Speech Communications*, 32(2):111–126, September 2000. 111

[31] C. Ding, X. He, H. Zh, M. Gu, and H.D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proc. International Conference on Data Mining*, pages 107–114, San Jose, CA, November 2001. 82

[32] R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press, Cambridge, UK, 1998. 25

[33] J.P. Egan. Articulation testing methods. *Laryngoscope*, 58(9):955–991, 1948. 31

[34] W.N. Francis and H. Kucera. *Frequency Analysis of English usage: Lexicon and Grammar*. Houghton-Mifflin, Boston, MA, 1982. 96

[35] T. Fukada, M. Bacchiani, K. Paliwal, and Y. Sagisaka. Speech recognition based on acoustically derived segment units. In *Proc. ICSLP*, volume 2, pages 1077–1080, Philadelphia, PA, October 1996. 26

[36] J.-L. Gauvain, L. Lamel, and G. Adda. Transcribing broadcast news for audio and video indexing. *Commun. ACM*, 43(2):64–70, 2000. 33

[37] J.L. Gauvain, L. Lamel, and G. Adda. The LIMSI broadcast news transcription system. *Speech Communications*, 37(1-2):89–108, May 2002. 22

[38] D. Gillick, S. Stafford, and B. Peskin. Speaker detection without models. In *Proc. ICASSP*, volume I, pages 757–760, Philadelphia, 2005. 113

[39] H. Gish, M.-H Siu, and R. Rohlicek. Segregation of speakers for speech recognition and speaker identification. In *Proc. ICASSP*, volume 2, pages 873–876, Toronto, CA, April 1991. 111

[40] J. Glass. *Finding Acoustic Regularities in Speech: Application to Phonetic Recognition*. PhD thesis, Massachusetts Institute of Technology, 1988. 26

[41] J. Glass. A probabilistic framework for segment-based speech recognition. *Computer, Speech, and Language*, 17(2-3):137–152, 2003. 38

[42] J. Glass, T. J. Hazen, L. Hetherington, and C. Wang. Analysis and processing of lecture audio data: Preliminary investigations. In *Proc. HLT-NAACL 2004 Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval*, pages 9–12, Boston, May 2004. 33

[43] J.J. Godfrey, E.C. Holliman, and J. McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *Proc. ICASSP*, pages 517–520, San Francisco, CA, March 1992. 33

[44] Y. Gong. Speech recognition in noisy environments: A survey. *Speech Communication*, 16(3):261–291, April 1995. 22

[45] H. Goodluck. *Language Acquisition*. Blackwell Publishers, Cambridge, MA, 1991. 21

[46] M. Goto. A chorus-section detecting method for musical audio signals. In *Proc. ICASSP*, volume 5, pages 437–440, April 2003. 25

[47] O. Gotoh. An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, 162:705–708, 1982. 25

[48] T. Hain, P.C. Woodland, G Evermann, and D. Povey. The CU-HTK March 2000 Hub5e transcription system. In *Proceedings of the DARPA Speech Transcription Workshop*, College Park, MD, May 2000. 22

[49] T.J. Hazen. *The Use of Speaker Correlation Information for Automatic Speech Recognition*. PhD thesis, Massachusetts Institute of Technology, 1998. 22

[50] T.J. Hazen, I.L. Hetherington, H. Shu, and K. Livescu. Pronunciation modeling using a finite-state transducer representation. In *Proceedings of the ISCA Tutorial and Research Workshop on Pronunciation Modeling*

and Lexicon Adaptation for Spoken Language, pages 99–104, Estes Park, CO, September 2002. ISCA. 39

[51] T.J. Hazen, L. Hetherington, H. Shu, and K. Livescu. Pronunciation modeling using a finite-state transducer representation. *Speech Communication*, 46(2):189–203, June 2005. 39

[52] I.L. Hetherington. An efficient implementation of phonological rules using finite-state transducers. In *Proc. Eurospeech*, pages 1599–1602, Aalborg, Denmark, September 2001. 39

[53] T. Holter and T. Svendsen. Combined optimisation of baseforms and model parameters in speech recognition based on acoustic subword units. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 199–206, Santa Barbara, CA, December 1997. 26

[54] T. Irino and R. Patterson. Segregating information about the size and the shape of the vocal tract us- ing a time-domain auditory model: The stabilised wavelet-mellin transform. *Speech Communication*, 36(3):181–203, March 2002. 135

[55] F. Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64:532–536, April 1976. 42

[56] K. W. Jørgensen, L. L. Mølgaard, and L. K. Hansen. Unsupervised speaker change detection for broadcast news segmentation. In *Proceedings of European Signal Processing Conference*, Florence, Italy, September 2006. 112, 113

[57] P.W. Jusczyk. *The discovery of spoken language*. MIT Press/Bradford Books, Cambridge MA, 1997. 21

[58] P.W. Jusczyk and R.N. Aslin. Infants' detection of sound patterns of words in fluent speech. *Cognitive Psychology*, 29(1):1–23, August 1995. 21

[59] E. J. Keogh and M. J. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 285–289. ACM Press, New York, NY, 2000. 42

[60] K.F.Lee and H.W.Hon. Speaker-independent phone recognition using hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 37(11):1641–1648, November 1989. 62

[61] D. Klein. *The Unsupervised Learning of Natural Language Structure*. PhD thesis, Stanford University, Palo Alto, CA, March 2005. 26

[62] C.-H. Lee, F. Soong, and B.-H. Juang. A segment model based approach o speech recognition. In *Proc. ICASSP*, pages 501–504, New York, NY, April 1988. 26

[63] K.F. Lee and H.W. Hon. Large-vocabulary speaker-independent continuous speech recognition. In *Proc. ICASSP*, pages 123–126, New York, NY, April 1988. 42

[64] L. Lee and R. Rose. Speaker normalization using frequency warping procedures. In *Proc. ICASSP*, pages 353–356, Atlanta, GA, May 1996. 135

[65] L. Lee and R. Rose. A frequency warping approach to speaker normalization. *IEEE Transactions on Speech and Audio Processing*, 6(1):49–60, January 1998. 135

[66] D. Lewis. Languages and language. In K. Gunderson, editor, *Language, Mind, and Knowledge*. University of Minnesota Press, Minneapolis, MN, 1975. 21

[67] Y-L. Lin, T. Jiang, and K-M. Chao. Efficient algorithms for locating the length-constrained heaviest segments with applications to biomolecular sequence analysis. *J. Computer and System Sciences*, 65(3):570–586, January 2002. 51

[68] A. Ljolje, D.M. Hindle, M.D. Riley, and R. W. Sproat. The AT&T LVCSR-2000 system. In *Proceedings of the DARPA Speech Transcription Workshop*, College Park, MD, May 2000. 22

[69] B. Logan and S. Chu. Music summarization using key phrases. In *Proc. ICASSP*, pages 749–752, Istanbul, Turkey, June 2000. 25

[70] J.F. Lopez and D.P.W. Ellis. Using acoustic condition clustering to improve acoustic change detection on broadcast news. In *Proc. ICSLP*, Beijing, China, October 2000. 112, 113

[71] A. Malegaonkar, A. Ariyaeeinia, P. Sivakumaran, and J. Fortuna. Unsupervised speaker change detection using probabilistic pattern matchin. *IEEE Signal Processing Letter*, TO Appear, 2006. 113

[72] M. Meila and J. Shi. Learning segmentation by random walks. In T.K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, volume 13, pages 873–879. MIT Press, Cambridge, MA, 2001. 82

[73] MIT. MIT Open Courseware Website: http://ocw.mit.edu. 33

[74] MIT. MIT World Website: http://mitworld.mit.edu. 33

[75] K. Mori and S. Nakagawa. Speaker change detection and speaker clustering using VQ distortion for broadcast news speech recognition. In *Proc. ICASSP*, pages 413–416, Salt Lake City, UT, May 2001. 112

[76] M.E. Munich and P. Perona. Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 108–115, Korfu, Greece, September 1999. 42

[77] C.S. Myers and L.R. Rabiner. A level building dynamic time warping algorithm for connected word recognition. *IEEE Transactions on Acoustics Speech and Signal Processing*, 29(2):284–297, April 1981. 42, 46

[78] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970. 25

[79] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133, 2004. 87

[80] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, 2004. 86

[81] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856, Cambridge, MA, 2002. MIT Press. 82

[82] K. Paliwal. Lexicon building methods for an acoustic sub-word based speech recognizer. In *Proc. ICASSP*, volume 2, pages 729–732, Alburquerque, NM, April 1990. 26

[83] D. Pallett, J. Fiscus, J. Garofolo, A. Martin, and M. Przybocki. Broadcast news benchmark test results: English and non-english word error rate performance measures, 1998. 33

[84] A. Park and J. Glass. Towards unsupervised pattern discovery in speech. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, San Juan, Puerto Rico, 2005. 41

[85] A. Park and J.R. Glass. A novel DTW-based distance measure for speaker segmentation. In *IEEE/ACL Workshop on Spoken Language Technology*, Palm Beach, Aruba, December 2006. (To be presented). 109

[86] A. Park and J.R. Glass. Unsupervised word acquisition from speech using pattern discovery. In *Proc. ICASSP*, Toulouse, France, April 2006. 41

[87] A. Park, T.J. Hazen, and J.R. Glass. Automatic processing of audio lectures for information retrieval: Vocabulary selection and language modeling. In *Proc. ICASSP*, Philadelphia, PA, March 2005. 22, 139

[88] G. Peeters, A.L. Burthe, and X. Rodet. Toward automatic music audio summary generation from signal analysis. In *Proceedings of Int'l Conference on Music Information Retrieval*, Paris, France, October 2002. 25

[89] F. Pereira and M. Riley. Speech recognition by composition of weighted finite automata. In E. Roche and Y. Schabes, editors, *Finite-State Language Processin*, pages 431–453. MIT Press, Cambridge, MA, 1997. 40

[90] F. Pereira and Y. Schabes. Inside-outside reestimation from partially bracketed corpora. In *30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Newark, Delaware, 1992. Association for Computational Linguistics. 26

[91] S. Pinker. *The Language Instinct*. William Morrow and Company, New York, NY, 1994. 21

[92] L. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993. 37

[93] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989. 42

[94] L.R. Rabiner, A. Rosenberg, and S. Levinson. Considerations in dynamic time warping algorithms for discrete word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(6):575–582, December 1978. 42

[95] D.A. Reynolds. Speaker identification and verification using gaussian mixture models. *Speech Communications*, 17(1-2):91–108, August 1995. 135

[96] I. Rigoutsos and A. Floratos. Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. *Bioinformatics*, 14(1):55–67, February 1998. 23

[97] D. Roy. *Learning from Sights and Sounds: A Computational Model*. PhD thesis, Massachusetts Institute of Technology, 1999. 27

[98] D. Roy and A. Pentland. Learning words from sights and sounds: A computational model. *Cognitive Science*, 26(1):113–146, January 2002. 27

[99] J.R. Saffran. Constraints on statistical language learning. *Journal of Memory and Language*, 47(1):172–196, July 2002. 23

[100] J.R. Saffran, R.N. Aslin, and E.L. Newport. Statistical learning by 8-month old infants. *Science*, 274:1926–1928, December 1996. 23

[101] H. Sakoe and S. Chiba. A dynamic programming approach to continuous speech recognition. In *Proceedings of International Congress on Acoustics*, Budapest, Hungary, 1971. Paper 20C13. 42

[102] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, February 1978. 42

[103] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical Report TR87-881, Cornell University, Ithaca, NY, 1987. 96

[104] G. K. Sandve and F. Drabløs. A survey of motif discovery methods in an integrated framework. *Biology Direct*, 1:1–11, April 2006. 25

[105] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000. 82

[106] Y. Shiraki and M. Honda. LPC speech coding based on variable-length segment quantization. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(9):1437–1444, 1988. 26

[107] H. Shu and I.L. Hetherington. EM traingig of finite-state transducers and its application to pronunciation modeling. In *Proc. ICSLP*, pages 1293–1296, Denver, CO, September 2002. 40

[108] M.-H Siu, G. Yu, and H. Gish. An unsupervised, sequential learning algorithm for the segmentation of speech waveforms with multiple speakers. In *Proc. ICASSP*, volume 2, pages 189–192, San Francisco, CA, March 1992. 111

[109] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981. 25

[110] Z. Solan, D. Horn, E. Ruppin, and S. Edelman. Unsupervised context sensitive language acquisition from a large corpus. In L. Saul, editor, *Advances in Neural Information Processing Systems*, volume 16, Cambridge, MA, 2004. MIT Press. 26

[111] Z. Solan, E. Ruppin, D. Horn, and S. Edelman. Automatic acquisition and efficient representation of syntactic structures. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, Cambridge, MA, 2003. 26

[112] N. Ström, I.L. Hetherington, T.J. Hazen, and J. Glass. Acoustic modeling improvements in a segment-based speech recognizer. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 139–142, Piscataway, NJ, December 1999. 39

[113] T. Svendsen, K. Paliwal, E. Harborg, and P.O. Husoy. An improved sub-word based speech recognizer. In *Proc. ICASSP*, volume 1, pages 108–111, Glasgow, Scotland, May 1989. 26

[114] T. Svendsen and F. Soong. On the automatic segmentation of speech signals. In *Proc. ICASSP*, pages 77–80, Dallas, TX, April 1987. 26

[115] A. Tritschler and R.A. Gopinath. Improved speaker segmentation and segments clustering using the bayesian information criterion. In *Proc. Eurospeech*, pages 679–682, Budapest, Hungary, September 1999. 111, 113

[116] S. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000. 86

[117] A. Venkataraman. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):352–372, September 2001. 28

[118] M. Vescovi, M. Cettolo, and R. Rizzi. A DP algorithm for speaker change detection. In *Proc. Eurospeech*, pages 2997–3000, Geneva, Switzerland, September 2003. 112

[119] M.S. Waterman and M. Eggert. A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. *Journal of Molecular Biology*, 197:723–725, 1987. 25

[120] S. White and P. Smyth. A spectral clustering approach to finding communities in graphs. In *SIAM International Conference on Data Mining*, Newport Beach, CA, 2005. 86

[121] A. Witkin. Scale-space filtering: A new approach to multi-scle description. In *Proc. ICASSP*, volume IX, pages 150–153, San Diego, CA, 1984. 120

[122] L. Xie. *Unsupervised pattern discovery for multimedia sequences*. PhD thesis, Columbia University, 2005. 25

[123] D. Yuret. *Discovery of linguistic relations using lexical attraction*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1998. 26

[124] V. Zue, S. Seneff, and J. Glass. Speech database development at MIT: TIMIT and beyond. *Speech Communications*, 9(4):351–356, August 1990. 35