Conversational Scene Analysis

by

Sumit Basu

S.B., Electrical Science and Engineering,
Massachusetts Institute of Technology (1995)
M.Eng., Electrical Engineering and Computer Science,
Massachusetts Institute of Technology (1997)

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 2002

Author .....
Department of Electrical Engineering and Computer Science
August 30, 2002

Certified by .....
Alex P. Pentland
Toshiba Professor of Media Arts and Sciences
Thesis Supervisor

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

Conversational Scene Analysis

by

Sumit Basu

Submitted to the Department of Electrical Engineering and Computer Science
on August 30, 2002, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

In this thesis, we develop computational tools for analyzing conversations based on nonverbal auditory cues. We develop a notion of conversations as being made up of a variety of scenes: in each scene, either one speaker is holding the floor or both are speaking at equal levels. Our goal is to find conversations, find the scenes within them, determine what is happening inside the scenes, and then use the scene structure to characterize entire conversations.

We begin by developing a series of mid-level feature detectors, including a joint voicing and speech detection method that is extremely robust to noise and microphone distance. Leveraging the results of this powerful mechanism, we develop a probabilistic pitch tracking mechanism, methods for estimating speaking rate and energy, and means to segment the stream into multiple speakers, all in significant noise conditions. These features gives us the ability to sense the interactions and characterize the style of each speaker's behavior.

We then turn to the domain of conversations. We first show how we can very accurately detect conversations from independent or dependent auditory streams with measures derived from our mid-level features. We then move to developing methods to accurately classify and segment a conversation into scenes. We also show preliminary results on characterizing the varying nature of the speakers' behavior during these regions. Finally, we design features to describe entire conversations from the scene structure, and show how we can describe and browse through conversation types in this way.

Thesis Supervisor: Alex P. Pentland
Title: Toshiba Professor of Media Arts and Sciences

Acknowledgments

There are so many people who have contributed to this endeavor in so many ways, it would be impossible to list them all. I apologize for any omissions, and I hope you know that you are in my heart and mind even if not on this page.

Of those I do mention, I would first and foremost like to thank Sandy for all of his guidance, support, and inspiration over all of these years. Sandy, you've taught me the importance of imagination and creativity in the pursuit of science, and about the courage it takes to try something really different. That spirit is something I will always carry with me.

The rest of my committee comes next, for all of the effort and thought they have put in to help with this process. First, Jim Rehg, of the College of Computing at Georgia Tech, who has been discussing this topic with me for quite some time. You've been a great friend and mentor, Jim, and I appreciate everything you've done for me. Leslie Pack Kaelbling, of the MIT AI Lab – you put in more time and effort than I ever would have expected, giving me detailed and insightful comments at all levels of the work. And finally, Trevor Darrell, also of the AI Lab. As always, Trevor, you shot straight from the hip and gave me some really solid feedback on the work – precisely why I wanted you on board. To my entire committee: all of your help, insight, and feedback are so very much appreciated.

Someone who could never fit into just one category is Irfan Essa, a research scientist in the group when I came in and now a tenured professor at Georgia Tech. Irfan, you have been mentor, collaborator, and friend to me: you showed me how to read technical papers, how to write technical papers, and helped me get my first major publication. More recently, you've been an incredible help getting me ready for the "real" world. Where would I be without you?

As Sandy has often said, "there's only one group of people you go to grad school with." I think I've been exceptionally lucky to have been here with the smartest and kindest set of people I could have hoped for. First among the Pentlanditos, of course, are Brian and Tanzeem, my officemates and best friends, who have been an unparalleled source of learning and joy. We've spent so many years together learning about pattern recognition and human behavior, both in a technical and a philosophical sense, that I wonder how I will navigate the world without you. Vikram, a more recent addition, has been a wonderful friend and colleague as well. Jacob, though he spent only a year here, is the closest friend I've ever had, and I hope we can continue telecommunicating for many years to come. Judy, Karen, and Liz have been wonderful to have around, both as friends and as skilled office managers. Then there's Deb, Ali, Chris, Push, Barbara, Nathan, Drew, Bernt, Nuria, Yuri, Tony, Tom, Peter, Kris, Ben, Paul, Francois, and so many others. You know who you are, and thank you for being there.

A special thank you to Marilyn Pierce in the EECS Graduate Office, whose help and reassurances over the years have been a welcome source of solace. There are an endless number of deadlines in graduate school, and with your help I've (finally) managed to get through them all.

Then there are the many wonderful people who have kept the wheels of this great institute turning, the unsung heroes of MIT: Will, Chi, and all of the others at NecSys; Kevin, Matt, Julie, Henry, and the rest of the Facilities crew; and Deb, Pat, Carol, and others in our custodial staff whose friendly hellos have brought smiles to so many dreary winter days.

Outside the academic walls, there are many more friends who have helped in one way or another: Mark, the best of my many roommates and a great friend in difficult times; Aaron and Mike, from back in the Infinite Monkey days, who introduced me to the concept of a social life; and more recently Kumalisa, who sometimes managed to pull me from my work-addicted shell to hit a Somerville party or two. Without a doubt, I must include The Band: Stepha, Anindita, Aditi, and Tracie – listening to our jam session tapes I realize that those Sunday afternoons were among the best times I've ever had in Cambridge.

My family – Baba, Ma, Didi Bhi, and more recently Zach – there is so much you have done for me. Without you, I could never have come this far, and I always thank you for everything you've done and continue to do. You may be far away, but I feel your support when I need it most.

Almost last but certainly not least, to Kailin, Stepha, and Jo: each in your own way, you have taught me something very important about love, and I will always remember you for that.

Finally, to everyone who has been with me in this long process, thank you for believing in me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The process of human communication involves so much more than words. Even in the voice alone, we use a variety of factors to shape the meaning of what we say – pitch, energy, speaking rate, and more. Other information is conveyed with our gaze direction, the motion of our lips, and more. The traditional view has been to consider these as "icing" on the linguistic cake – channels of additional information that augment the core meaning contained in the words. Consider the possibility, though, that things may work in the other direction. Indeed, it may be these other channels which form the substrate for language, giving us an audio-visual context to help interpret the words that come along with it. For instance, we can watch people speaking to each other in a different language and understand much of what is going on without understanding any of the words. Presumably, then, by using these other signals alone we can gain some understanding of what is going on in a conversation.

This is the idea behind what we call "conversational scene analysis" as a parallel to the well-established field of Auditory Scene Analysis (ASA) [5]. In the ASA problem, the goal is to take an arbitrary audio signal and break it down into the various auditory events that make up the scene – for instance, the honking of a car horn, wind rustling by, a church bell. In Conversational Scene Analysis (CSA), we use a variety of input signals but restrict ourselves to the domain of conversations, in the hopes that we can make a much deeper analysis.

Let us consider the meaning of a conversational scene in this context. Like a

movie, a given conversation is made up of a number of scenes. In each scene, there are actors, each of whom is performing a variety of actions (typically utterances for the purposes of our analysis). Furthermore, the roles that these actors play with respect to each other vary from scene to scene: in a given scene, person A may be leading the conversation, while in another, A and B may be rapidly exchanging quips. In fact, it is these changes in role that will determine the scene boundaries.

The goal of this work is to develop computational means for conversational scene analysis, which breaks down into a number of tasks. First, we want to find where scenes exists, and whether a given pair of people are having a conversation. Next, we would like to find the scene boundaries, and even predict a change of scene. We would also like to characterize the roles the actors are playing, and find out who, if anyone, is holding the floor. We would like to find out the characteristics of the individual speakers in their scene, i.e., how they're saying what they're saying. Finally, we wish to characterize the overall conversation type based on its composition of scenes.

Why is this worth doing? There are a number of compelling reasons for this work that imply a broad range of future applications. First, there is the spectre of the vast tracts of conversational audio-visual data already collected around the world, with more being added every day: meetings, classrooms, interviews, home movies, and more. There is an ever-growing need of effective ways to analyze, summarize, browse, and search through these massive stores of data. This requires something far more powerful than a fast-forward button: it is important to be able to examine and search this data at multiple scales, as our proposed ontology of actors, scenes, and conversation types would allow us to do.

Another motivation is the huge number of surveillance systems installed in stores, banks, playgrounds, etc. For the most part, the security people on the other end of these cameras are switching amongst a large number of video feeds. In most cases, the audio is not even a part of these systems for two reasons. First, it is assumed that the only way to use it would be to listen to the content, which could be a huge privacy risk, and second, because it would just be too much information – you can watch 10 monitors at once, but you can't listen to 10 audio streams and make sense of them.

With a mechanism to analyze conversational scenes, these security guards could get summary information about the conversations in each feed: on a playground, is an unknown individual trying to start conversations with several children? In an airport, are two or more individuals talking to each other repeatedly with their cellphones? In a store, is a customer giving a lecture to a salesperson? All of these situations can be easily resolved with human intervention – if they can be caught in time.

Surveillance is not always a matter of being watched by someone else – sometimes we want to have a record of our own interactions. The higher level analyses developed here will result in a powerful feedback tool for individuals to reflect on their conversations. There may be aspects of our style that we never notice – perhaps we always dominate the conversation, never letting the other person get a word in edgewise, or perhaps we are curt with certain people and chatty with others. While these differences in style may be obvious to third party observers, they are often difficult for us to notice, and sometimes socially unacceptable for others to tell us. Thinking further ahead, a real-time mechanism for analyzing the scene could give us immediate feedback about our ongoing conversations.

The personal monitoring vein extends to our fourth area, applications in health and wellness. Clinicians have long noted that depression, mania, fatigue, and stress are reflected in patients' speaking styles: their pitch, energy, and speaking rate all change under different psychological conditions. With the analysis techniques we will develop here, we can quantify these style parameters. This is quite different from attempting to recognize emotions – we are merely characterizing how a person's characteristics are changing with respect to his norm. Though this will not result in a litmus test for depression, it could be a very useful tool for the patient and the doctor to see how they vary from day to day and how certain behaviors/drugs are affecting their state of mind.

A final motivation for this work is in the development of socially aware conversational agents and user interfaces. If we finally do achieve the vision of the robot butler, for instance, it would be nice if it could understand enough about our interactions to interrupt only at appropriate times. In a more current scenario, your

car could be constantly analyzing the conversational scenes you are involved in on your cellphone and in the car. Coupled with the wealth of current research in automatic traffic analysis, this could become an important tool for accident prevention. If you are holding the floor in a conversation or firing back and forth while entering a complicated intersection, the car could forcibly interrupt the conversation with a warning message, reconnecting you once the difficult driving scenario had passed. To be even more concrete, any interface which needs to ask for your attention – your instant messenger application, your cellphone, your PDA – could all benefit from being conversationally aware.

This is still but a sampling of the possible reasons and applications for this work: furthermore, the more refined our techniques become, the more varied and interesting the application areas will be.

1.1 Our Approach

The area of conversational scene analysis is broad, and we will certainly not exhaust its possibilities in this work. However, we will make significant gains in each of the tasks we have described above. We now present the reader with a brief roadmap of how we will approach these tasks and the technologies involved. Note that for this study, we will be focusing exclusively on the auditory domain. Though we have spent significant efforts on obtaining conversation-oriented features from the visual domainin our past work, particularly in terms of head [3] and lip [4] tracking, we have yet to integrate this work into our analysis of conversations.

We will begin our work with the critical step of mid-level feature extraction. We first develop a robust, energy-independent method for extracting the voiced segments of speech and identifying groupings of these segments into speech regions using a multi-layer HMM architeture. These speech regions are the individual utterances or pieces thereof. The novelty of this method is that it exploits the changing dynamics of the voicing transitions between speech and non-speech regions. As we will show, this approach gives us excellent generalization with respect to noise, distance from micro-

phone, and indoor/outdoor environments. We then use the results of this mechanism to develop several other features. Speaking rate and normalized voicing energy are simple extensions. We then go on to develop a probabilistic pitch tracking method that uses the voicing decisions, resulting in a robust tracking method that can be completely trained from data.

At this point, we will already have developed the methods necessary for finding and describing the utterances of the individual actors in the conversation scene. We then begin our work on the conversational setting by briefly examining the problem of speaker separation, in which we attempt to segment the speech streams that are coming from different speakers. We look at this in two cases: in the first version, there are two microphones and two speakers; in the second, there is only one microphone. The features available to us are energy, energy ratios, and the direction of arrival estimate. The challenge here is to use the right dynamics in integrating these noisy features, and we show how our voicing segmentation provides a natural and effective solution.

The next task is to find and segment the conversational scenes. We start this with an investigation of how to determine that two people are involved in an interaction. We construct a simple measure of the dynamics of the interaction pattern, and find a powerful result that lets us very reliably determine the answer with only two-minute samples of speech. We then develop a set of block-based features for this higher level of analysis that characterize the occurences in the scene over a several second window. We use some of these features in an HMM that can identify and segment three kinds of states: speaker one holds the floor, speaker two holds the floor, or both are parlaying on equal footing. While the features themselves are noisy, the dynamics of the exchanges are strong and allow us to find the scenes and their boundaries reliably. We also show how we can predict these scene boundaries just as they are beginning – though we see many false alarms as well, the latter give us interesting information about possible changeover times. Once we have found the scene boundaries, we show how we can integrate features over the course of the scene to describe additional characteristics of it.

Finally, we develop two summary statistics for an entire conversation, i.e., a collection of scenes, and show how we can use the resulting histograms to describe conversation types. This highest level of description gives us an interesting bird's eye view of the interaction, and could prove to be a powerful feature for browsing very long-scale interactions.

This course of work will not cover all possible aspect of CSA, but it begins the path to an important new area. We will thus conclude with some ideas about specific directions that we hope to take this work in the years to come.

1.2 Data Sources

For the mid-level features through the conversation finding work, we will use data from a variety of sources: some of it from desktop microphones, some from condenser microphones, and some from body-mounted electret microphones – we will describe the details of these situations as they occur in our experiments. For the remainder of those experiments and for the entirety of the scene finding and characteriziation work, our primary source of data will be the LDC Callhome English database.

This database, collected by the LDC (Linguistic Data Consoritium) at the University of Pennsylvania, consists of 63 conversations over international telephone lines. It is freely available to member institutions of the LDC and on a fee-basis to non-members. Native English speakers were recruited to call friends or family members overseas who were also native English speakers. The subjects agreed to completely release the contents of their conversation for research purposes: in return, they were able to have the international phone call free of charge and were compensated $10 for their time. The data is split into two channels, one for each speaker, each sampled at 8 kHZ with 8-bit mulaw encoding. There is a varying degree of noise due to the variance in quality of international telephone lines, sometimes appearing as constant static and sometimes as bursty segments of periodic noise, as we will describe later. Since the calls are all between friends and family members, the interactions are quite natural and represent a wide variety of interaction styles. These are precisely the

kinds of variations we are trying to analyze, and thus this is an ideal data source for our experiments.

1.3 Previous Work

As this project covers a wide array of technical areas, the list of related work is large. However, there has been little coordination of these areas into our manner of analysis, as we will see below. This section will give an overview of the work done in the related areas, while details of the respective techniques will be illuminated as necessary in the technical descriptions of later chapters.

In terms of the low-level auditory features, there has been a variety of work in doing related tasks. A majority of it has dealt only with the case of close-talking microphones, as this has been the primary case of interest to the speech community. For our purposes, though, we require robustness both to noise and microphone distance. There has been prior work in finding voicing segments and tracking pitch in noisy conditions, and we will show how our new method significantly outperforms these results in terms of robustness and smoothness. Our application of the linked HMM to this problem is novel, and allows us to take advantage of the dynamics of speech production. It also gives us simultaneous decoding of the voicing and speech segmentation, with each helping the other's performance. Furthermore, as our features are independent from signal energy levels, our technique works in a wide variety of conditions without any retuning. We also present algorithms for probabilistic pitch tracking, speaking rate estimation, and speaking energy, all based on our segmentation and robust to significant environmental noise.

Speaker segmentation has also received some attention from the speech community, but in ways that are not ideal for our problem. The speaker identification methods (see [9] for a review) are effective only when there are 15-30 second contiguous segments of speech to work over. The direction-of-arrival based methods using multiple microphones are similarly effective when there are relatively stationary sources, but have difficulty in distinguishing short segments from noise in the

cross-correlation and use artificial dynamic constraints to smooth over these changes. Since we are interested in capturing the sudden changes that occur in conversation, it is necessary for us to attempt something different. We show how we can leverage the results of our voicing-speech model to integrate over the noisy features of energy and phase (direction of arrival), resulting in performance that far exceeds that of the raw sigals.

As for the conversation detection and alignment results, we know of little other work in this area. While there has been some work on detecting conversations for a PC user based on head pose and direction of arrival [23], we are working on a quite different task: detecting whether two streams are part of the same conversation. To our knowledge, this is the first attempt to work on such a task, and our results are surprisingly strong. We show how we can pick out a conversational pair from amongst thousands of possibilities with very high accuracy and very low false alarm rates using only two minutes of speech. This has obvious applications in security, and we feel it is a major contribution to the speech processing community.

This brings us to our work on conversations and conversational scenes. While we are not the first to look at conversations, we are the first to look at their higher level structure in these terms. There has been a long history of work in linguistics and more recently in speech processing to determine discourse events using intonational cues (see the work of Heeman et al. [13] and of Hirschberg and Nakatani [14]). This work is at a much lower level of detail than we are considering here – their goal is to determine the role of particular utterances in the context of a dialogue; e.g., is new information being introduced? Is this answering a previous question? Our interest, on the other hand, is in finding the structure of the interaction: who is running the show, how are the individual actors behaving, and what overall type of interaction are they having?

The main other work in this final area is on multimedia and meeting analysis. The BBN "Rough'n'Ready" system, for instance, attempts to segment news streams into stories based on the transcription results of a speech recognition system [22]. While their results are very impressive, they are for quite a different task than ours.

Their goals involve tracking changes in the *content* of the dialogue, while we are more interested in the nature of the interaction. They are not able and not (yet) interested in analyzing conversational scenes. Hirschberg and Nakatani have also attacked this problem, seeking to use intonational cues to signal shifts in topic [15] with modest results. While this an interesting avenue, it differs from our notion of conversational scenes, which has more to do with the flow of interaction. We expect that the shifts in speaker dominance we seek will signify changes in topic, but we do not pursue this hypothesis in this work.

The other significant work on conversations comes from groups working on meeting analysis, among the most successful of which has been Alex Waibel's "Meeting Browser" system [32]. While their overall goal is to help in browsing multimedia data of conversations, their path has been quite different from ours. Their primary focus has been on enhancing speech recognition in this difficult scenario and using the results for automatic summarization. More recently, they have begun some preliminary work on speech act classification (e.g., was this a question, statement, etc?). They do not seem interested as of yet in the overall structure of the interaction, though to us it seems this could be a very powerful mechanism to aid in browsing conversations.

With this background, we are ready to embark on our study of conversational scene analysis. What follows covers a broad range of material, and we will make every attempt to refer to the appropriate literature when explaining our models and methods. We do expect a basic familiarity with signal processing and graphical models. If the reader feels they would like more background in these, we would heartily recommend the following: for graphical models, Jordan and Bishop's new book *An Introduction to Graphical Models* [17] is an excellent guide for novices and experts alike; for speech processing, Oppenheim and Schafer's classic text *Discrete-Time Signal Processing* [24] is a signal processor's lifetime companion.

Chapter 2

Auditory Features

There are many features in the auditory domain that are potentially useful for conversational scene analysis, but we choose to focus on a few: when there is speech information, what energy and pitch it is spoken with, and how fast or slowly it is being spoken. Our choice comes from a long history of results championing these features in psycholinguistics, for instance the work of Scherer et al. in 1971 [30]. Scherer and his colleagues showed that pitch, amplitude, and rate of articulation were sufficient for listeners to be able to judge the emotional content of speech. While we are not focusing on emotions, we strongly believe that they are a parallel dimension to the information we seek – i.e., if there is enough information to judge the emotional content, there should be enough to judge the conversational interactions. Beyond this, the literature tells us little about which computational features to use, as our task of conversational scene analysis is still new.

By the end of this chapter, we will deal with each of these features in turn. We must begin, though, at the beginning, by finding when there is even speech to be processed.

2.1 Speech and Voicing Detection

To understand the problem of speech and voicing detection, we must first examine the process of speech production. Figure 2-1 shows a simplified model of what occurs

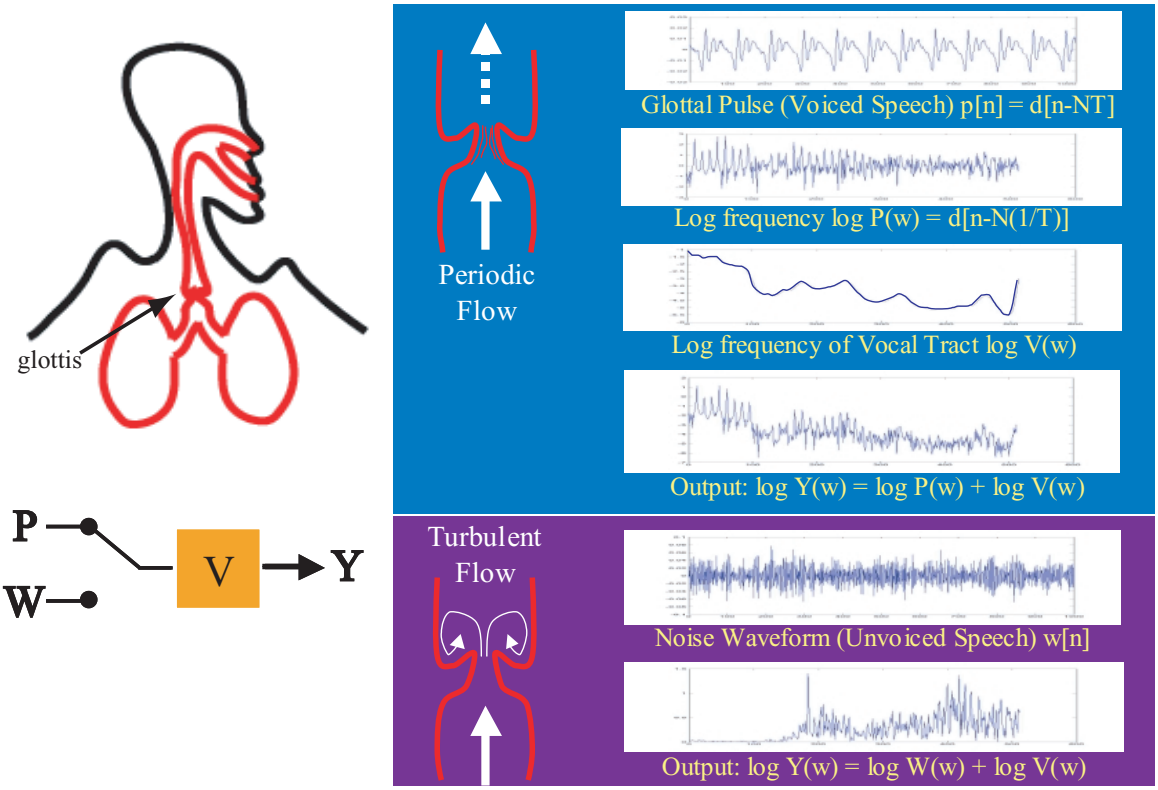

Figure 2-1: The human speech production system. The lungs push air through the glottis to create either a periodic pulse, forcing it to flap open and closed, or just enough to hold it open in a turbulent flow. The resulting periodic or flat spectrum is then shaped by the vocal tract transfer function $V(w)$.

during speech production, adapted from [25]. Speech can be broken up into two kinds of sounds: voiced and unvoiced. The voiced sounds are those that have a pitch, which we can think of loosely as the vowel sounds. The unvoiced sounds are everything else – bursts from the lips like /p/, fricatives like /s/ or /sh/, and so on. During the voiced segments, the lungs build up air pressure against the glottis, which at a certain point pops open to let out a pulse of air and then flaps shut again. This happens at a fixed period and results in an (almost) impulse train $p[n]$, whose Fourier transform $P[w]$ is thus also an (almost) impulse train with a period that is the pitch of the signal. This impulse train then travels through the vocal tract, which filters the sound with $V[w]$ in the frequency domain, resulting in the combined output $Y[w]$, which is the vocal tract filter's envelope multiplied by an impulse train, i.e.,

$$Y[w] = V[w] * P[w]. \tag{2.1}$$

This is where the expressive power of our vocal instrument comes in: humans have a great deal of flexibility in how they can manipulate the vocal tract to produce a variety of different resonances, referred to as formants. The result is the full range of vowels and then some. In the unvoiced case, the lungs put out just enough pressure to push the glottis open and keep it open, as shown in the panel to the lower right. Once again, the sound is shaped by the configuration of the vocal tract, including the position of the tongue and teeth. This results in sounds like /s/ and /sh/. The remaining cases of plosives, such as /p/ and /t/, result from pressure buildup and release at other place in the vocal tract – the lips for /p/ and the tongue and palate for /t/. The common element of all of these cases is that the resulting sound is not periodic.

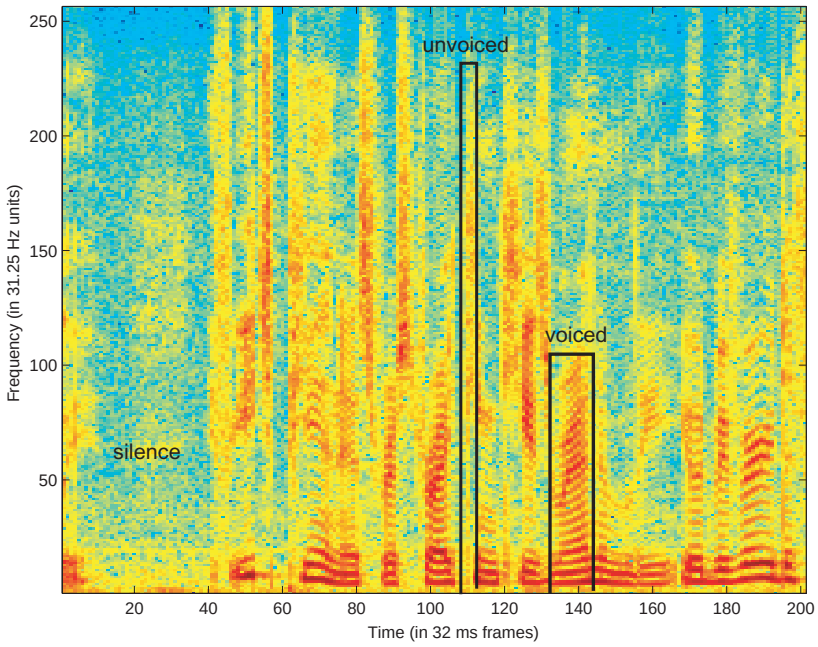

Figure 2-2: Spectrogram of a speech signal sampled at 16kHz with a close-talking microphone. Note the banded nature of the voiced sounds and the clear high-frequency signature of unvoiced sounds.

Figure 2-2 shows the spectrogram of a speech signal sampled at 16 kHz (8 kHz Nyquist cutoff), with FFT's taken over 32ms windows with an overlap of 16ms between windows. There are number of things to note in this image. First of all, in the voiced regions, we see a strong banded structure. This results from the product of the impulse train in frequency from the glottis $P[w]$ multiplying the vocal tract transfer function $V[w]$. The bands correspond to the peaks of the impulse train. Since the pitch is in general continuous within a voiced region due to human limitations, these bands are continuous as well. Notice also how much longer the voiced regions are with respect to the unvoiced regions. In the unvoiced regions themselves, we see that the energy is strongly biased towards the high frequencies. It appears, however, that the energy of the unvoiced regions is almost as strong as that of the voiced regions. This somewhat misleading effect comes from two factors: first, this data was taken with a "close-talking microphone," the typical sort of headset microphone used for nearly all speech recognition work, and second, the signal has been "preemphasized," i.e., high-pass filtered, to increase the visibility of the unvoiced regions.

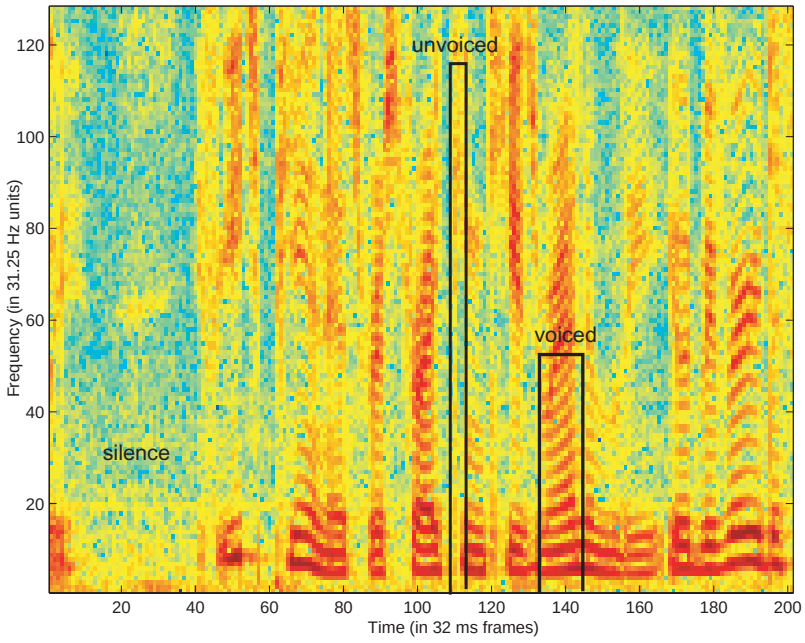

Figure 2-3: Spectrogram of a speech signal sampled at 8kHz with a close-talking microphone. Note that unvoiced sounds are now much less visible.

When we now look at the same piece of the signal sampled at 8kHz (figure 2-3), we now have half the frequency range to work with (4 kHz cutoff). As a result, it is much more difficult to distinguish the unvoiced components from silence. This problem only becomes worse when we move from close-talking microphones to far-field mics – whereas the power of the periodic signals carries well over distance and noise, the noise-like unvoiced signals are quickly lost. At even a few feet away from a microphone, many unvoiced speech sounds are nearly invisible.

Our goal is to robustly identify the voiced and unvoiced regions, as well as to group them into chunks of speech to separate them from silence regions. Furthermore, we want to do this in a way that is robust to low sampling rates, far-field microphones, and ambient noise. Clearly, to work in such broad conditions, we cannot depend on the visibility of the unvoiced regions. There has been a variety of work on trying to find the boundaries of speech, a task known in the speech community as "endpoint detection." Most of the earlier work on this topic has been very simplistic as the speech recognition community tends to depend on a close-talking, noise-free microphone situation. More recently, there has been some interest in robustness to noise, due to the advent of cellular phones and hands-free headsets. For instance, there is the work of Junqua et al. [18] which presents a number of adaptive energy-based techniques, the work of Huang and Yang [16], which uses a spectral entropy measure to pick out voiced regions, and later the work of Wu and Lin [34], which extends the work of Junqua et. al by looking at multiple bands and using a neural network to learn the appropriate thresholds. Recently, there is also the work of Ahmadi and Spanias [1], in which a combination of energy and cepstral peaks are used to identify voiced frames, the noisy results of which are smoothed with median filtering. The basic approach of these methods is to find features for the detection of voiced segments (i.e., vowels) and then to group them together into utterances. We found this compelling, but noted that many of the features suggested by the authors above could be easily fooled by environmental noises, especially those depending on energy.

We thus set out to develop a new method for voicing and speech detection which was different from the previous work in two ways. First, we wanted to make our

low-level features independent of energy, in order to be truly robust to different microphone and noise conditions. Second, we wished to take advantage of the *multi-scale dynamics* of the voiced and unvoiced segments. Looking again at the spectrograms, there is a clear pattern that distinguishes the speech regions from silence. It is not in the low-level features, certainly – the unvoiced regions often look precisely like the silence regions. In speech regions, though, we see that voicing state is transitioning rapidly between voiced (state value 1) and unvoiced/silence (state value 0), whereas in the non-speech regions, the signal simply stays in the unvoiced state. The dynamics of the transitions, then, are different for the speech and non-speech regions. In probabilistic terms, we can represent this as follows:

$$P(V_t = 1|V_{t-1} = 1, S_t = 1) \neq P(V_t = 1|V_{t-1} = 1, S_t = 0) \tag{2.2}$$

This is clearly more than the simple HMM can model, for in it the current state can depend only on the previous state, not on an additional parent as well. We must turn instead to the more general world of dynamic Bayesian nets and use the "linked HMM" model proposed by Saul and Jordan [29]. The graphical model for the linked HMM is shown in figure 2-4. The lowest level states are the continous observations from our features, the next level up (V_t) are the voicing states, and the highest level (S_t) are the speech states.

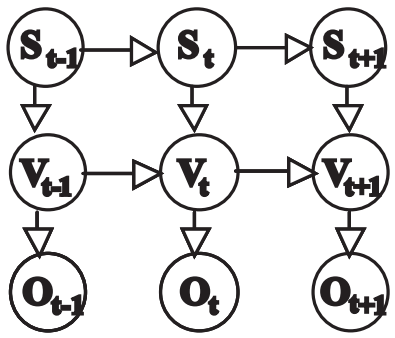

Figure 2-4: Graphical model for the linked HMM of Saul and Jordan.

This model gives us precisely the dependencies we needed from equation 2.2. Note that as in the simple HMM, excepting the initial timestep 0, all of the states in each layer have tied parameters, i.e.,

$$P(V_t = i|V_{t-1} = j, S_t = k) = P(V_{t+1} = i|V_t = j, S_{t+1} = k) \quad (2.3)$$

$$P(S_t = i|S_{t-1} = j) = P(S_{t+1} = i|S_t = j) \quad (2.4)$$

$$P(O_t = x|S_t = i) = P(O_{t+1} = x|S_{t+1} = i) \quad (2.5)$$

$$(2.6)$$

In a rough sense, the states of the lower level, V_t, can then model the voicing state like a simple HMM, while the value of the higher level S_t will change the transition matrices used by that HMM. This is in fact the same model used by the vision community for modeling multi-level dynamics, there referred to as switching linear dynamics systems (as in [26]). Our case is nominally different in that both hidden layers are discrete, but the philosophy is the same. If we can afford exact inference on this model, this can be very powerful indeed: if there are some places where the low-level observations $P(O_t|V_t)$ give good evidence for voicing, the higher level state will be biased towards being in a speech state. Since the speech state will have much slower dynamics than the voicing state, this will in turn bias other nearby frames to be seen as voiced, as the probability of voicing under the speech state will be much higher than in the non-speech state. We will see this phenomenon later on in the results.

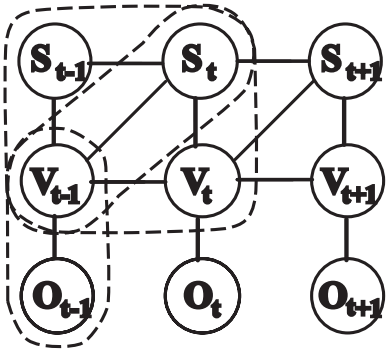

Figure 2-5: The clique structure for the moralized graph for the linked HMM.

In our case, inference and thus learning are fairly easy, as both of our sets of hidden states are discrete and binary. The clique structure for the moralized, triangulated

graph of the model is shown in figure 2-5. The maximal clique size is three, with all binary states, so the maximum table size is $2^3 = 8$. This is quite tractable, even though we will have an average of two of these cliques per timestep. The junction tree resulting from these cliques is shown for a four-timestep linked HMM in figure 2-6. It is easy to see by inspection that this tree satisfies the junction tree property [17], i.e., that any set of nodes contained in both cliques A and B are also contained in all cliques between A and B in the clique graph. Doing inference on this tree is analagous to the HMM except for an additional clique for each timestep (cliques 3, 5, etc.). To flesh out this analogy, we use node 9 as the root of the tree and collect evidence from the observations and propagate them to the root (the forward pass); then propagate the results back to the individual nodes (backward pass). Scaling is achieved by simply normalizing the marginals to be proper probability distributions; the product of the normalizing constants is then the log likelihood of the data given the model. This is an exact parallel to $\alpha - \beta$ scaling in HMMs [27].

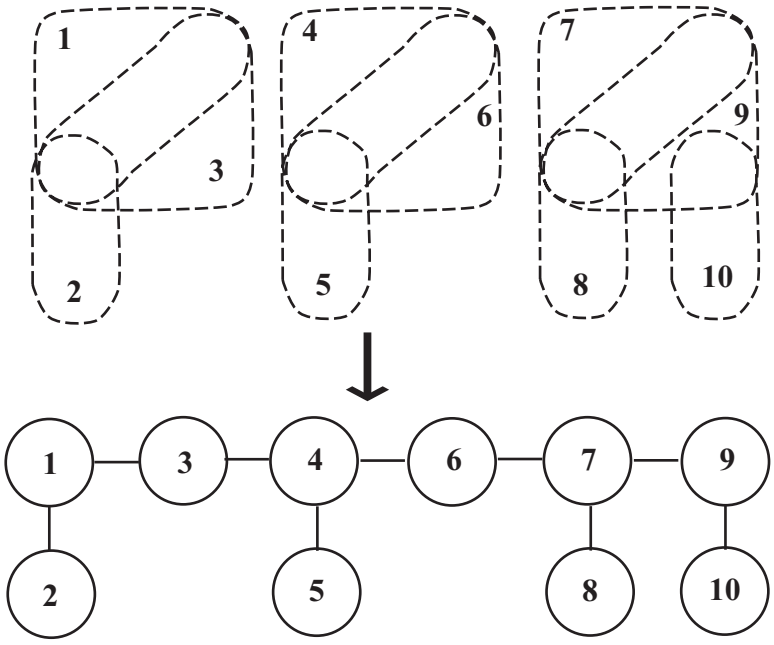

Figure 2-6: The clique structure and resulting junction tree for a four-timestep linked HMM. To prevent clutter, the nodes are not shown and the cliques have been spaced apart between timesteps. The root of the tree is node 9.

Since the cliques containing hidden nodes connect to each other via two nodes, the

primary effort in inference will involve marginalizing two-by-two-by-two tables onto two-by-two tables, then using these to update two-by-two-by-two potentials. The number of operations per timestep $O(t)$ is then

$$O(t) = 2\left[N_{s,1} + N_{s,1}N_{s,2}^2 + N_{s,1}^2 N_{s,2}\right], \tag{2.7}$$

where $N_{s,1}$ is the number of states in the lower hidden layer and $N_{s,2}$ is the number of states in the upper hidden layer. The first term is for computing the likelihoods of each state from the features, while the second two are for updating the two 3-cliques belonging to each timestep. The factor of two is for doing the forward and backward passes of the junction tree algorithm. Since both our hidden layers have binary states, this results in 36 operations per timestep.

For a simple HMM, the order of operations would be

$$O(t) = 2\left[N_{s,1} + N_s^2\right]. \tag{2.8}$$

With a binary voiced/unvoiced state, this would require only 12 operations per timestep; with another binary HMM on top of this for the speech/non-speech decision the total would be 24 operations per timestep. We will show later, though, that such a model would not be as effective as the full linked HMM.

On the other hand, we could fully represent the linked HMM model with a *four-state* HMM, representing each combination of voicing and speech states as a separate state. For instance, state 1 would be [speech=0,voice=0], state 2 would be [speech=0,voice=1], and so on. We could then tie the observation models for the voiced and unvoiced states (for states [speech=0,voice=0] and [speech=1,voice=1]), resulting in the same $2[2 + 4^2]$ or 36 operations per frame that we had for the linked HMM.

While 36 operations per frame versus 24 is a significant difference, it certainly does not make our model intractable: it is because of our small state size that we are saved from an explosion in the number of operations. We can thus apply the standard junction tree algorithm for inference without any need for approximations.

2.1.1 Features

We are using three features for the observations: the non-initial maximum of the normalized "noisy" autocorrelation, the number of autocorrelation peaks, and the normalized spectral entropy. These are all computed on a per-frame basis – in our case, we are always working with 8 kHz speech, with a framesize of 256 samples (32 milliseconds) and an overlap of 128 samples (16 milliseconds) between frames.

Noisy Autocorrelation

The standard short-time normalized autocorrelation of the signal $s[n]$ of length N is defined as follows:

$$a[k] = \frac{\sum_{n=k}^{N} s[n]s[n-k]}{(\sum_{n=0}^{N-k} s[n]^2)^{\frac{1}{2}}(\sum_{n=k}^{N} s[n]^2)^{\frac{1}{2}}} \tag{2.9}$$

We define the set of autocorrelation peaks as the set of points greater than zero that are the maxima between the nearest zero-crossings, discounting the initial peak at zero ($a[0]$ is guaranteed to be 1 by the definition). Given this definition, we see a small number of strong peaks for voiced frames because of their periodic component, as seen in figure 2-7. Unvoiced frames, on the other hand, are more random in nature, and thus result in a large number of small peaks (see figure). We thus use both the maximum peak value and the number of peaks as our first two features.

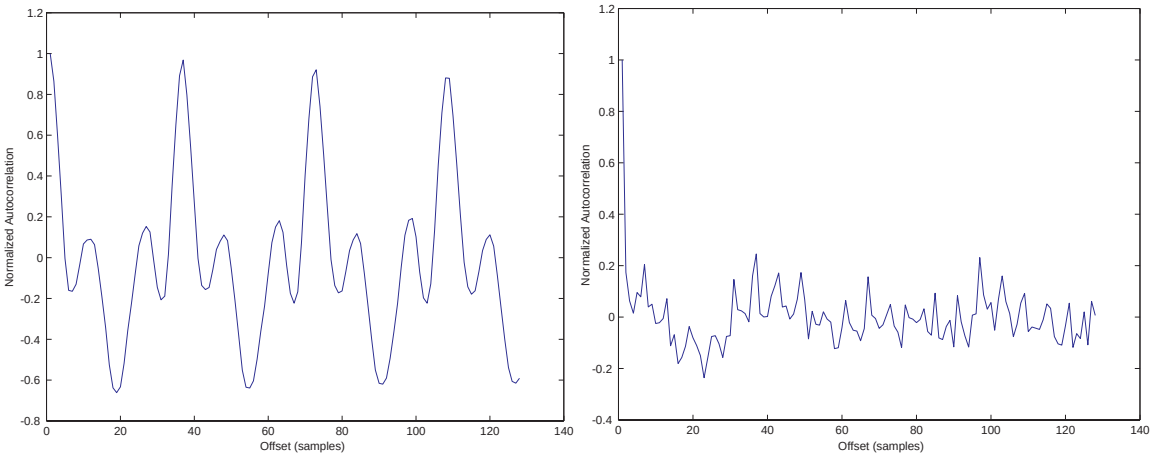

Figure 2-7: Autocorrelation results for a voiced (left) and an unvoiced (right) frame.

There is one significant problem to the standard normalized autocorrelation, though – very small-valued and noisy periodic signals will still result in strong peaks. This is a necessary consequence of the normalization process. As much of the data we are analyzing comes from the LDC callhome database, which is composed entirely of international telephone calls, we see many forms of channel noise that have low energy but are still periodic. Furthermore, they have a fairly noisy structure for each period. An example is shown in figure 2-8 below. In the non-speech regions, we see a very light band of periodic energy at a low frequency, but in the autocorrelogram, we see rather strong correpsonding peaks, which could make the resulting features very attractive to the voiced model.

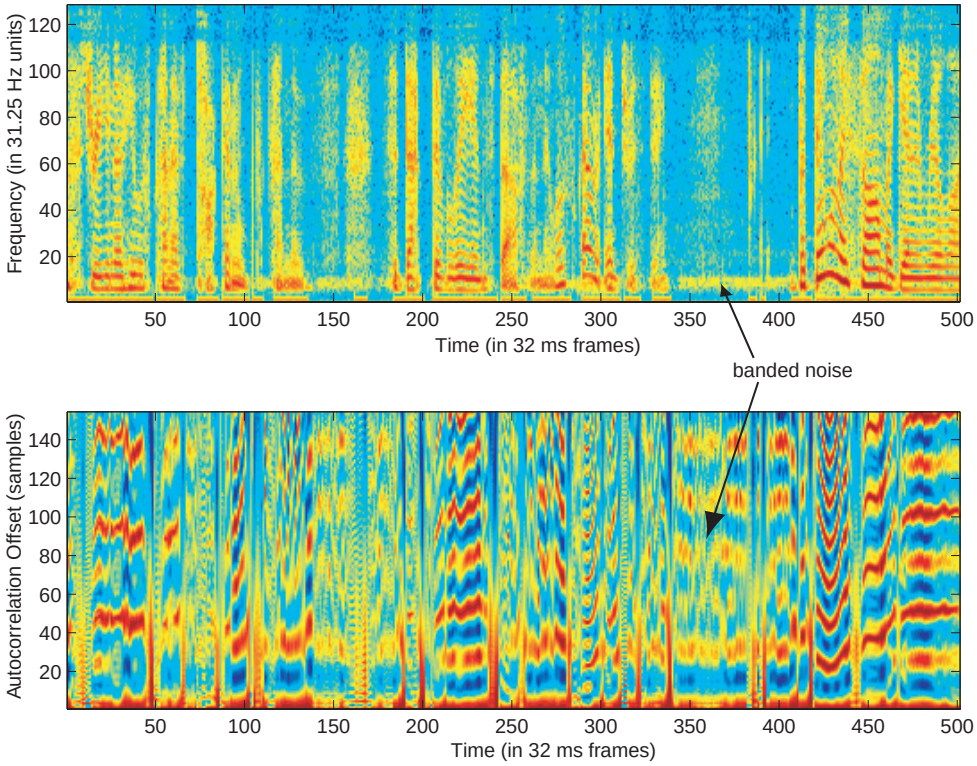

Figure 2-8: Spectrogram (top) and normalized autocorrelogram (bottom) for telephone speech showing a low-power periodic noise signal. Note the light spectral band in the non-speech regions and the strong resulting autocorrelation peaks.

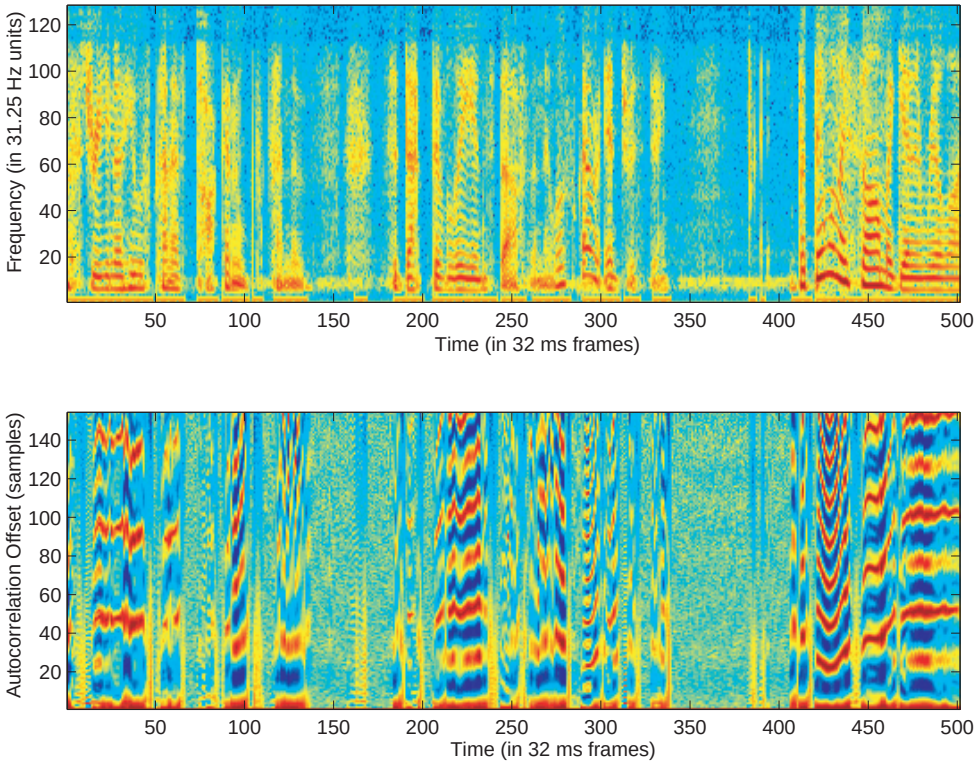

Figure 2-9: Spectrogram (above) and our new *noisy* autocorrelogram (below) for telephone speech showing a low-power periodic noise signal. Note how the autocorrelation peaks in the voiced regions are unaffected while the peaks in the non-speech regions have practically disappeared (compare to standard autocorrelogram in figure 2-8).

We could deal with this by simply cutting out frames that were below a certain energy, but this would make us very sensitive to the energy of the signal, and would result in a "hard" cutoff for a frame to be considered as voiced. This would lead us to a significant loss in robustness to varying noise conditions. We instead devised a much softer solution, which is to add a very low-power Gaussian noise signal to each frame before taking the autocorrelation. In the regions of the signal that have a strong periodic component, this has practically no effect on the autocorrelation. In lower power regions, though, it greatly disrupts the structure of a low-power, periodic noise source. In figure 2-9, we see the result of this procedure. The lower the signal power, the greater the effect will be on the autocorrelation, and thus we have a soft rejection of low power periodic components. To estimate the amount of noise to use, we use a two-pass approach – we first run the linked-HMM to get a rough segmentation of voicing and use the resulting non-speech regions to estimate the signal variance during silence. We then add a Gaussian noise signal of this variance to the entire signal and run the segmentation again.

Spectral Entropy

Another key feature distinguishing voiced frames from unvoiced is the nature of the FFT magnitudes. Voiced frames have a series of very strong peaks resulting from the pitch period's Fourier transform $P[w]$ multiplying the spectral envelope $V[w]$. This results in the banded regions we have seen in the spectrograms and in a highly structured set of peaks as seen in the first panel of figure 2-10. In unvoiced frames, as seen in the right panel, we see a fairly noisy spectrum, be it silence (with low magnitudes) or a plosive sound (higher magnitudes). We thus expect the entropy of a distribution taking this form to be relatively high. This leads us the notion of spectral entropy, as introduced by Huang and Yang [16].

To compute the spectral entropy, we first normalize $P[w]$ to make it into a proper distribution.

$$p[w] = \frac{P[w]}{\sum P[w]} \tag{2.10}$$

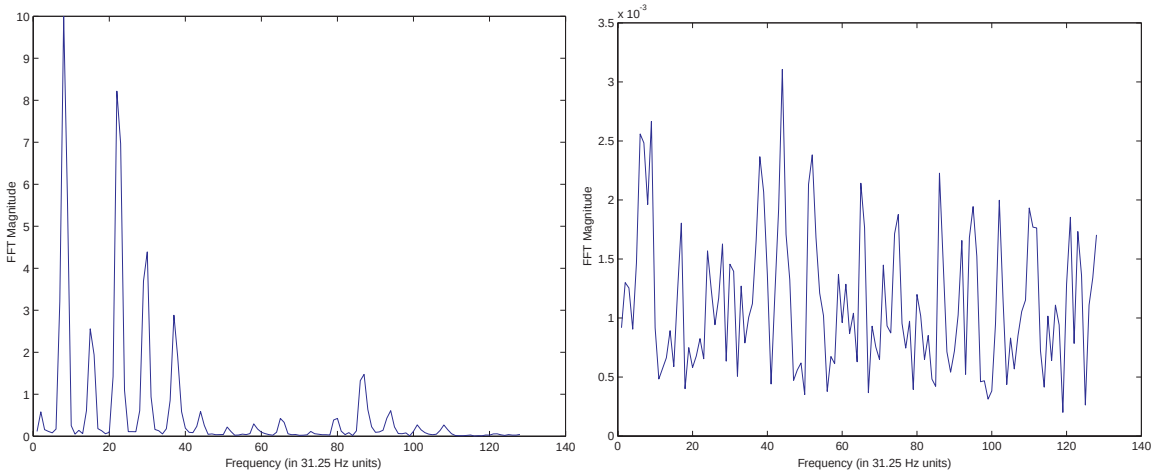

Figure 2-10: FFT magnitude for a voiced (left) and an unvoiced (right) frame. The spectral entropy for the first frame is 3.41; the second has an entropy of 4.72.

Normalizing in this way makes this feature invariant to the signal energy, as by Parseval's relation we are normalizing out the energy of the signal. We can then compute the entropy of the resulting distribution:

$$H_s = -\sum_w p[w] \log p[w]. \tag{2.11}$$

In figure 2-10, H_s is 3.41 for the voiced frame and 4.72 for the unvoiced frame – as we would expect, the entropy for the flat unvoiced distribution is much higher. Of course, there is some variation in the range of entropy values for various signals, so we maintain a windowed mean and variance for this quantity and then normalize the raw H_s values by them.

We can take this one step further and compute the *relative* spectral entropy with respect to the mean spectrum. This can be very useful in situations where there is a constant voicing source, such as a loud fan or a wind blowing across a microphone aperture. The relative spectral entropy is simply the KL divergence between the current spectrum and the local mean spectrum, computed over the neighboring 500 frames:

$$H_r = -\sum_w p[w] \log \frac{p[w]}{m[w]}, \tag{2.12}$$

where $m[w]$ is the mean spectrum. The performance gain from using the relative entropy is minimal for our synthetic noise environments, as the additive noise has a flat spectrum. In outdoor conditions, though, it makes a significant difference, as we will show in our experiments.

2.1.2 Training

With our features selected, we are now ready to parametrize and train the model. We choose to model the observations with single Gaussians having diagonal covariances. It would be a simple extension to use mixtures of Gaussians here, but since the features appear well separated we expected this would not be necessary. Furthermore, reducing the number of parameters in the model greatly reduces the amount of training data necessary to train the model.

We first unroll the model to a fixed sequence length (as in figure 2-6) of 500 timesteps. This is not necessary in principle, as our scaling procedure allows us to deal with chains of arbitrary length, but this allows us to get a more reliable estimate for the parameters of the initial nodes $P(v_0)$ and $P(s_0)$.

Because we can do exact inference on our model, the Expectation-Maximization or EM algorithm [8] is an obvious candidate for learning the parameters. The basic idea is to infer the distribution over the unlabeled hidden nodes (the expectation step) and then maximize the likelihood of the model by setting the parameters to the expectations of their values over this distribution. The tied parameters do not complicate this procedure significantly – it simply means that we accumulate the distributions over each set of nodes sharing the same set of parameters, again as with the HMM [27]. Furthermore, for our training data, the hidden nodes are fully labeled, as the voicing and speech state are labeled for every frame. As a result, the application of EM is trivial. However, we could easily train the model on additional, unlabeled data by using EM in its full form.

We trained the model using several minutes of speech data from two speakers in the callhome database (8000 frames of 8 kHz, 8-bit mulaw data) with speech and voicing states labeled in each frame. Since all states were labeled, it was only necessary to

run EM for one complete iteration.

2.1.3 Performance

To now use the model on a chunk of data, we first unroll it to an appropriate size. We then enter the evidence into the observed nodes, but instead of doing inference, or "sum-product," on the junction tree, we now use the "max-product" algorithm, propagating the *maximum* of each marginal configuration instead of the sum [20]. This is a generalization of the Viterbi algorithm for HMMs, and finds the posterior mode of the distribution over hidden states given the observations. In other words, the sequence produced by the max-product algorithm is the one that has the highest likelihood of having produced the observations we entered.

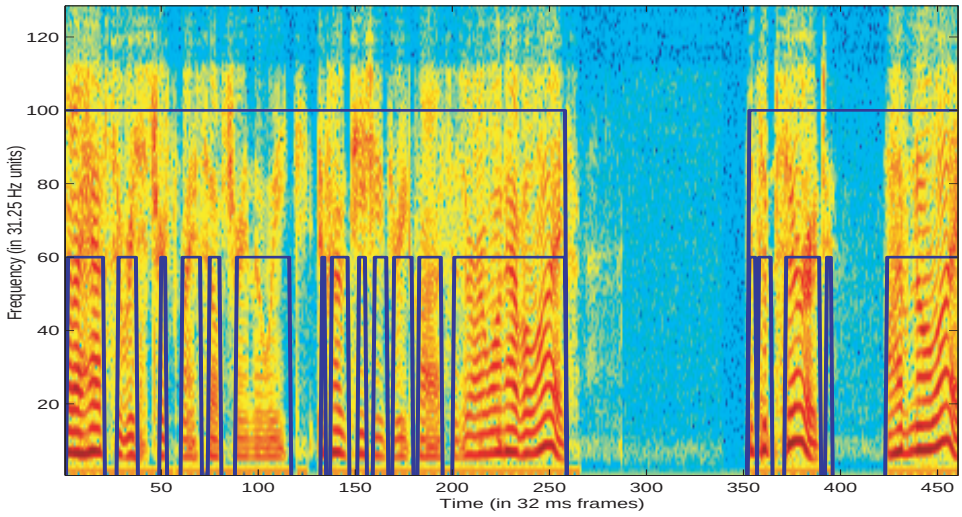

Figure 2-11: Performance of the model on telephone speech. The upper line (at 100) shows the inferred speech state and the lower line (at 60) shows the voicing state.

A first example of our results on telephone speech is shown in figure 2-11. As we had hoped, the model has correctly segmented the many voiced regions, as well as identifying the speech and non-speech regions. While this is encouraging, we wish to see the detailed performance of the model under noise and distance from microphone. Before we do this, though, we would like to point out the major advantages of this model over a simple HMM for detecting voiced/unvoiced states. First of all, we are

also getting the speech and non-speech segmentation. However, one could argue that we could simply have run a second HMM as a separate upper layer, using the posterior state probabilities of the HMM as features (as used in [6] for modeling daily behavior patterns). However, in that case, the information would flow only in one direction – from the lower level to the higher level. An increased posterior for the speech state would not affect the performance of the lower level. To illustrate this, we show the results of applying an ordinary HMM versus our linked HMM in noisy conditions in figure 2-12. Notice how our model is able to more reliably find the voicing states. We can understand why by thinking about the flow of information in the inference process: the "strong" voicing states (chunks 1, 3, and 4), which are captured by both models, are feeding information into the upper (speech state) level, biasing it towards a speech state. This then flows back down to the voicing level, since the probability of a voiced state is much higher when the upper level is in a speech state, i.e.,

$$P(V_t = 1|V_{t-1} = i, S_t = 1) >> P(V_t = 1|V_{t-1} = i, S_t = 0). \tag{2.13}$$

As a result, the model is more lenient about the features when the posterior for the speech state is high, and it is able to capture the presence of the weaker voiced chunk. The simple HMM is unable to take advantage of this higher level information. While these differences appear subtle, they are quite important in terms of reliably capturing the speech features, as we will rely on this low-level voicing segmentation for many of our later results.

Robustness to Noise

In this set of experiments, we show the robustness of our algorithm to noise and compare our results with some related work. We will use the measure of *segmental signal-to-noise ratio*, or SSNR, to evaluate the noise condition. The SSNR is the SNR (in dB) averaged over frames. This avoids one of the basic problems with using the SNR for speech, i.e., the domination of the signal variance by the high power regions of the signal. This results in an overall SNR value being somewhat exagerrated [25].

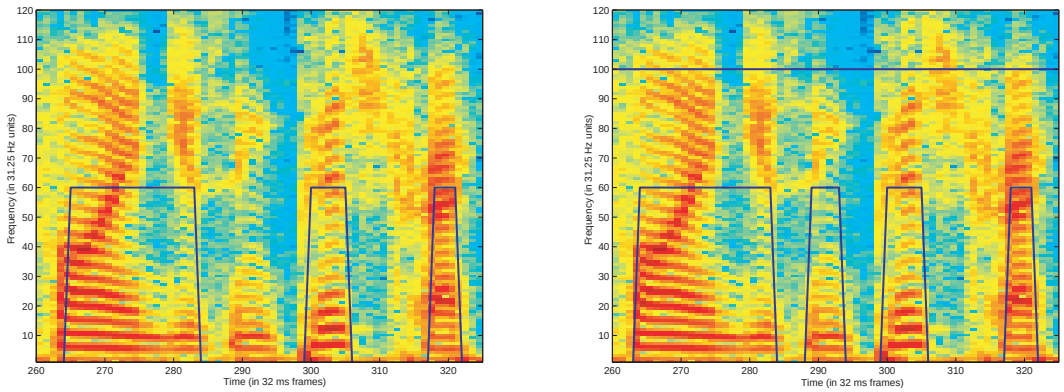

Figure 2-12: Comparison of an ordinary HMM (left) versus our linked HMM model (right) on a chunk of noisy data. Notice how our model is capable of more reliably finding the voiced segments since it can use the increased posterior of being in a speech region.

We can compute the SSNR for K frames of a zero-mean signal $s[n]$ with added noise $w[n]$ as follows:

$$SSNR = \frac{1}{K} \sum_{i=1}^{k} 20 \log \frac{\sigma^2_{s_k[n]}}{\sigma^2_{w_k[n]}}. \tag{2.14}$$

The best reported results in the literature for voicing detection in noise are from Spanias and Ahmadi [1]. Their approach was to use the logical and of two features: an adaptive threshold test for energy and one for the cepstral peak. The threshold for each feature is chosen as the median of that signal over the entire file. They employ no time dynamics, but use a 5-frame median filter to smooth the results of their detection. They report separate errors for V-UV (labeling a voiced frame as unvoiced) and for UV-V (labeling an unvoiced frame as voiced), as well as the total voicing error (V-UV + UV-V), for several SSNR values. They do not attempt to find speech/non-speech regions. As in their work, we hand labeled a small set of speech (2000 frames in our case). Each frame was labeled as being voice/unvoiced and speech/non-speech by both examining the clean spectrogram and listening to the corresponding audio. We then added Gaussian noise of varying power to simulate various noise conditions. As their signals themselves were not available, we implemented their technique so that we could compare its results directly on our data. The resulting conditions

are somewhat different from their paper – whereas their original signal was taken in studio conditions with "infinite" SSNR, we took our data from a vocal microphone about 10 inches from the speaker's mouth, with an estimated SSNR of 20dB.

The errors reported are fractions of the total number of frames with no weighting for signal power. We show a comparison of our results for total voicing error, V-UV, and UV-V in figure 2-13 below.

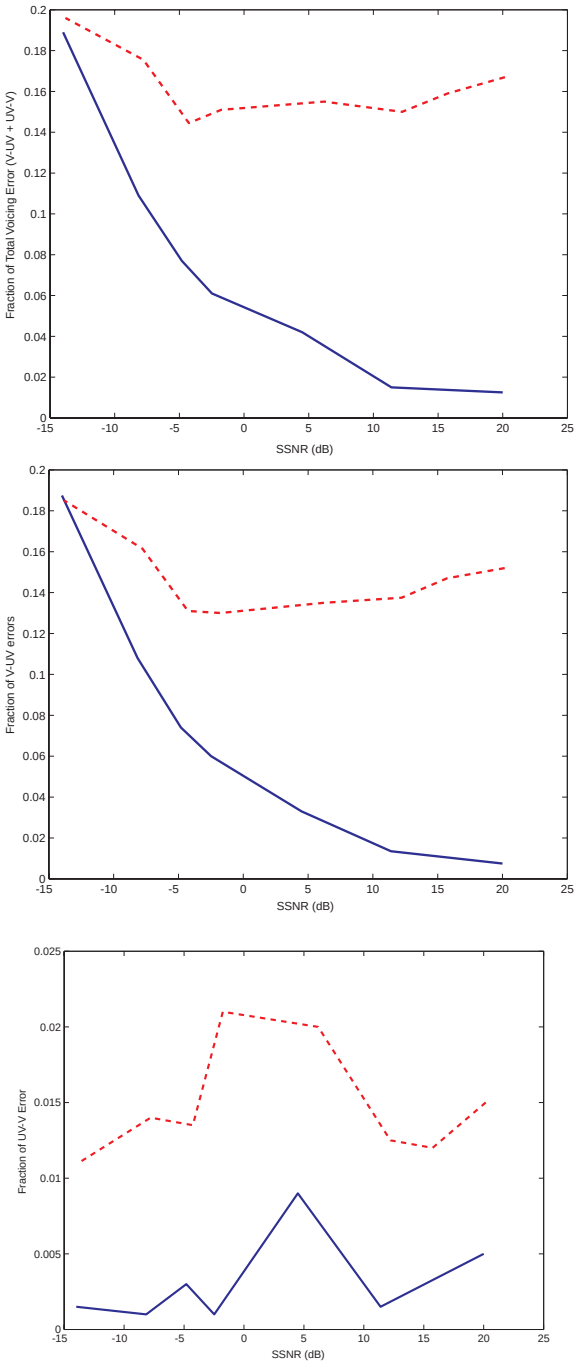

Figure 2-13: Comparison of the voicing segmentation in various noise conditions using our method (solid lines) against our implementation of the Ahmadi and Spanias algorithm [1] (dashed lines).The first plot shows the total fraction of frames having a voicing error (V-UV + UV-V), the second shows V-UV error (voiced frames classified as unvoiced), and the third shows UV-V error (unvoiced frames classfied as voiced). Note that the range on the UV-V error is an order of magnitude smaller than the other plots. The x-axis of each plot is the SSNR value.

It is interesting to note that the results of the Ahmadi and Spanias method do not worsen monotonically with increasing noise. This is due to their heuristic for choosing feature thresholds – under just the right amount of noise, the thresholds achieve their best values. As a result, adding noise can actually improve the performance by shifting the thresholds in the right way. In addition, because their method requires the energy *and* the cepstral peak to be above a threshold, it tends to clip off the beginning and end of many voiced segments, which tend to be lower energy though still clearly voiced.

To show what is happening in increasing noise, we show the performance of our model on our worst case (-14dB SSNR, 0.18 total voicing error) in figure 2-14. While the results are not perfect, they are certainly usable. Note that nearly all the voiced segments have been identified, though some have been shrunk somewhat. This ability of our model is due to its multi-level nature: as the strong voicing candidates push up the posterior for the speech state, this local belief propagates to other voicing states and makes them more likely to identify frames as being voiced.

To further demonstrate why our results degrade gracefully with noise, we show a piece of the original signal with and without noise and the frame-based energy of the two signals in figure 2-15. In the 30dB signal, the energy provides a good clue towards whether the frame is voiced. In the noise condition, however, the energy is almost useless. It is for precisely this reason that we have kept our features as independent of energy as possible.

We would like to make another less formal comparison: Droppo and Acero [11] report an average total voicing error rate of 0.084 for clean speech, which is worse than our figure (0.013) for this condition. However, they use a dynamic model for their segmentation (a single HMM) as part of a pitch tracking algorithm, and we suspect the nature of their method would have at least some of the robustness of ours in noise conditions.

Before we leave this experiment, we would like to show one last performance figure – the performance of the speech segmentation with respect to noise (figure 2-16). The performance is quite robust. Even at -14dB, we are only misclassifying 17% of the

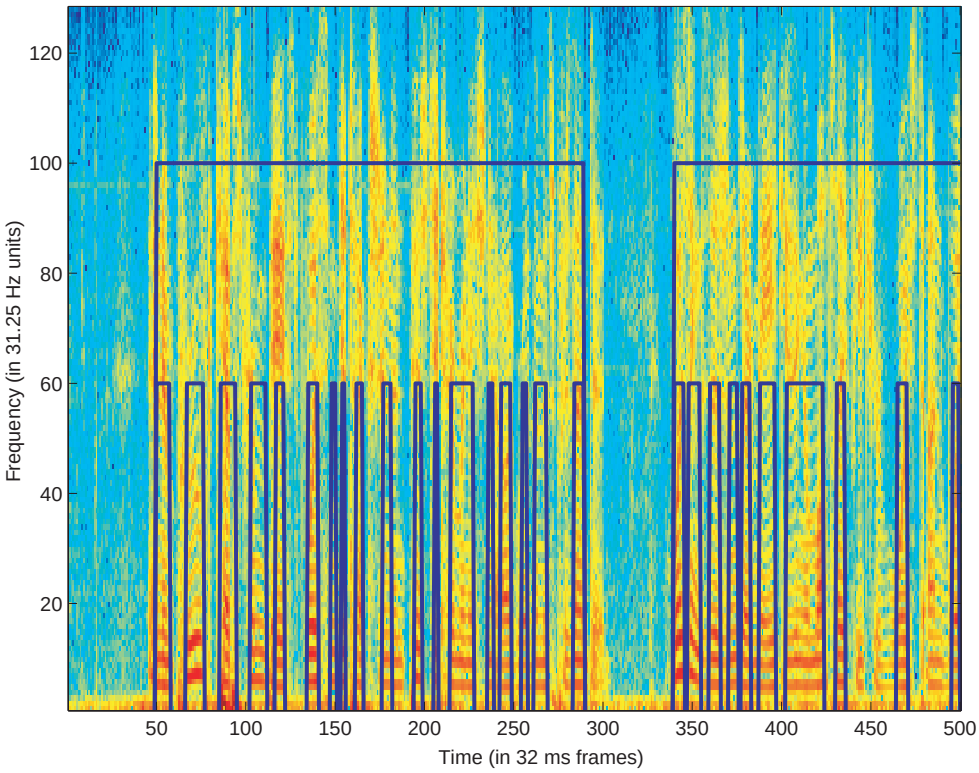

Figure 2-14: Speech and voice segmentation performance by our model with an SSNR of -14 dB. The spectrogram is shown without noise for clarity. Notice that most of the voiced segments are still detected, though some pieces are missing, such as around frame 450.

frames. As with the voicing segmentation, we see that this error is almost entirely made up of S-US errors, i.e., classifying speech frames as non-speech frames. The US-S error is very small, which is again very useful for robustness – our method will not be taking random noisy/high-energy signals and classifying them as speech. The main reason this segmentation works so well is again the multi-level nature of the model: the bits of voicing candidates we can still find increase the posterior for the speech state.

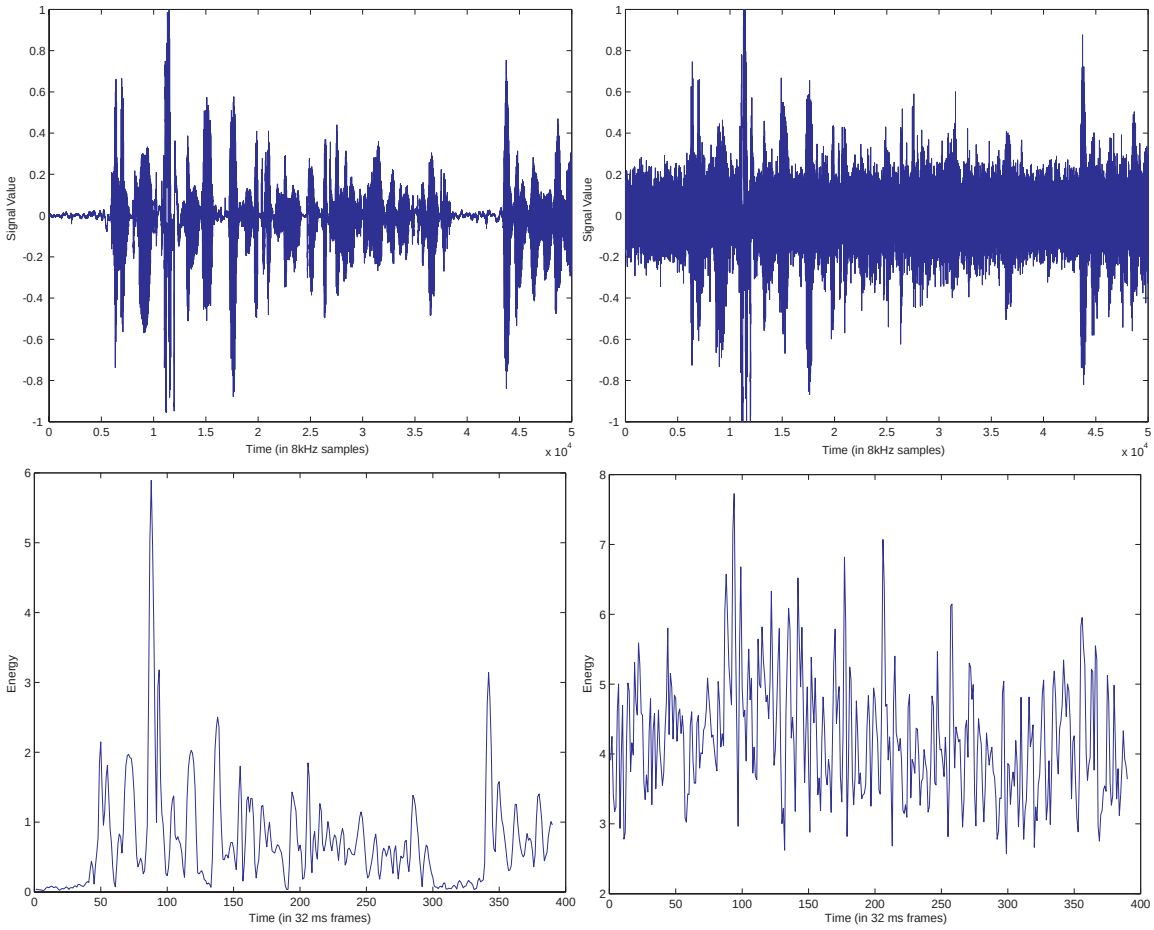

Figure 2-15: Speech signals (top) and corresponding frame-based energy (bottom) for an SSNR of 20dB (left) and -14dB (right). Note how difficult it is to distinguish the voiced regions in the noisy case.

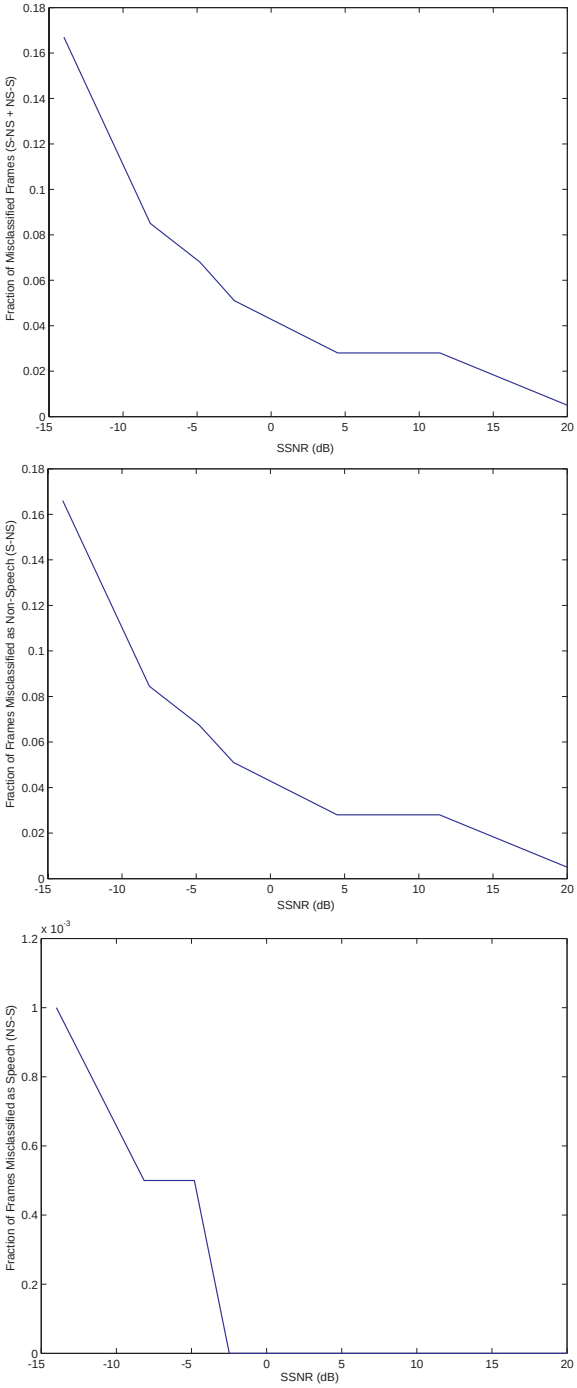

Figure 2-16: Performance of the speech segmentation in various noise conditions. The first plot shows the fraction of frames having a speech segmentation error (S-NS + NS-S), the second shows the S-NS errors (speech frames classified as nonspeech), and the third shows the NS-S error (nonspeech frames classfied as speech). Note that the NS-S error is two orders of magnitude smaller than the S-NS error. The x-axis of each plot is the SSNR value.

Robustness to Microphone Distance

Another important goal for our method is to be robust to microphone distance. In any real environment, distance adds more than Gaussian noise – now the sounds of fans, doors, chairs and the like all become comparable in power to the speech signal as the distance increases. We tested this condition by putting a far-field condenser microphone (an AKG C1000s) on a table in an office environment, then moving successively further away from the mic. The total voicing error and speech error for this experiment are shown in figure 2-17.

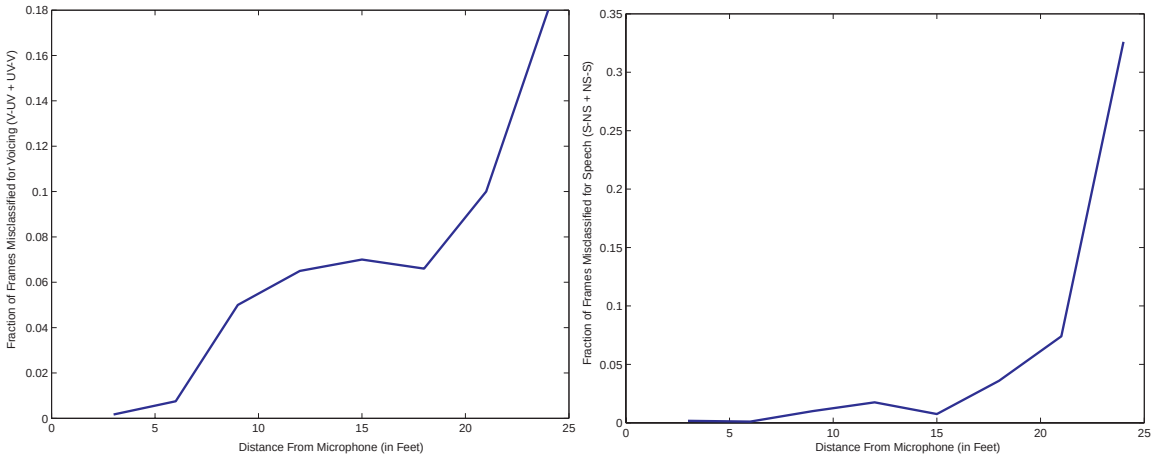

Figure 2-17: Performance of the voicing and speech segmentation with distance from the microphone. The first plot shows the total voicing error (V-UV + UV-V); the second shows the total speech error (S-NS + NS-S). The x-axis of each plot is the distance from the microphone in feet.

We estimate the SSNR of the signal at 24 feet to be -18 dB. However, since the noise no longer has a white spectrum, it is potentially more difficult to contend with. However, our method is still robust to this difficult condition. By 21 feet (about -10 dB of SSNR), we still have less than 10% error in both voicing and speech segmentation. These results will be very useful to us in later sections when we need to work with data recorded at a distance. It is also a novel result – to our knowledge, nobody has shown robustness to distance for a voicing/speech segmentation task.

Robustness to Environment

To further test the robustness of our algorithm to real-world noise sources, we collected data from an outdoor environment. For this experiment, the subject was wearing a "sociometer," a portable audio/accelerometer/IR tag recorder developed by Tanzeem Choudhury, with the housing designed by Brian Clarkson (see figure 2-18). The subject went for a short walk outdoors with two companions, keeping up a conversation for most of the time. The sociometer has a simple electret microphone (the same component found in cellular phones, answering machines, etc.) nominally protected by a foam windscreen and recessed behind a circular aperture in the plastic housing. The microphone sits about 6-7 inches away from the subject's mouth, at which distance environmental sounds are often of the same or greater power than the speech. Furthermore, the wind blowing across the microphone produces a strong resonance which looks precisely like a voiced sound to our features – however, we are able to avoid this error by now using the relative spectral entropy, as this resonance is constant in time and is thus captured by the local mean spectrum.

Figure 2-18: A "sociometer," a portable audio/accelerometer/IR tag recorder developed by Tanzeem Choudhury, with the housing designed by Brian Clarkson. The microphone is about 6-7 inches away from the speaker's mouth.

In table 2.1, we show the results of applying our algorithm to the outdoor data. These values were computed on a per-frame basis over a 2000 frame sequence which involved significant wind noise, revolving doors, packages being dropped, and car noises (though no horns). The speaker's companions were 4-8 feet from the micro-

phone, and thus much of their speech was picked up as well. We show two values for the probability of detection: $P_S(D)$, the probability of detecting voicing/speech for the subject, and $P(D)$, the overall probability of detection voicing/speech for the subject *and* her companions. $P(FA)$ reflects the fraction of false alarms, in this case defined as the non-voiced/speech frames incorrectly classified as voiced/speech.

Table 2.1: Performance of voicing/speech segmentation on outdoor data. $P_S(D)$ is the probability of detecting a voicing/speech frame from the subject, $P(D)$ is the probability of detecting a voicing/speech frame from *any* audible person. $P(FA)$ is the probability of mislabeling a non-voiced/speech segment.

	$P_S(D)$	$P(D)$	$P(FA)$
Voicing	0.971	0.936	0.017
Speech	0.973	0.982	0.007

2.1.4 Applications

Because our voicing and speech segmentation methods show robustness to noise, distance, and environment, there are a variety of possible applications at this stage. The most obvious among these is to use the speech segmentation information to notify speech recognition systems when to listen. Currently, most desktop speech recognizers rely on headset microphones and use energy as an endpoint detection cue. If a desktop microphone is used, any environmental noise sets off the recognizer. By using our technique in place of this, a desktop recognizer could simply pick out the actual speech segments for processing. Our later results on signal separation and localization will add more leverage to such a system.

Another application that is less obvious but a personal favorite is the idea of "Smart Headphones," which we introduced in [2]. The basic idea is that somebody listening to music over headphones loses awareness of their auditory surroundings (figure 2-19). Our proposal was to detect speech events in the auditory environment and pass them through to the user. Our implementation at that time used a simpler, less accurate method for speech segmentation [2], but was still effective in its

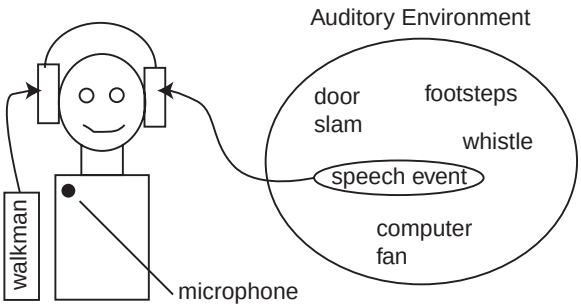

Figure 2-19: The "smart headphones" application. Speech regions are segmented from the auditory environment and passed through the user's headphones, allowing him to remain socially engaged while wearing headphones.

purpose. Preliminary user testing showed that people greatly appreciated being able to understand the speech of others around them while still listening to their music. Our current linked-HMM method for speech segmentation should further improve the effectiveness of this application.

2.2 Probabilistic Pitch Tracking

Determing the pitch of speech signals has long been a challenge for the speech community. Most modern pitch tracking methods stem from the work of Secrest and Doddington from 1983 [31], in which they introduced the use of dynamic programming for making a voiced-unvoiced decision and tracking the pitch. The basic idea was that each potential pitch value and the unvoiced state had a certain cost associated with being chosen as the representation for that frame; furthermore, there was a cost of transitioning from a given candidate to another in successive time frames. The process of optimal pitch tracking, then, was the process of running the Viterbi algorithm on the lattice of possible pitch assignments.

Generations of papers since then have introduced a variety of features resulting in varying degrees of performance increase. However, the main problem with these algorithms has always been the tuning of the dynamic programming parameters – what should the costs be for the individual pitch candidates and the transitions? While more discriminative features result in less sensitivity to the parameter values, tuning is always necessary for different speakers, sampling rates, and noise conditions. In response to these problems, some more recent approaches to pitch tracking have applied probabilistic methods to model some of the parameters, most notably the work of Droppo and Acero [11]. The authors there model the likelihood of a pitch candidate being the true pitch (i.e., the negative of the node cost), but stop short of learning the transition parameters. They instead model the latter as a function of the difference between pitch candidates in successive frames, but leave a scaling factor λ to be chosen empirically.

We follow the approach of Acero and Droppo's work, but continue further down the path to achieve a fully probabilistic pitch tracker which can be trained entirely from data. There is one primary difference in our work – since we have a reliable estimate of the voicing state from our method described above, we do not introduce this as part of the pitch tracker's problem. We instead run the tracker only on those frames our linked HMM model has identified as being voiced. This is following in

the tradition of [1] and [33], who both use a separate voicing determination as a preprocessor to the pitch tracking.

2.2.1 The Model

If we consider the problem of tracking pitch amongst a variety of candidates in each frame, the problem maps precisely to an HMM in which the individual states correspond to possible pitch periods. The problem with this view, of course, is that we have a very large number of states – 160 or so, one for each candidate. This is not intractable in and of itself, but the problem is that training the transition parameters would take a large amount of a data from a large number of speakers, as each person's pitch range will map to a different part of the transition matrix. A speaker with a low pitch will have zero transitions for high pitches and vice versa. Furthermore, when our training is done, all of these different speakers' transitions will be squashed into a single transition matrix, muddying the transition constraints of individual speakers.

As a result, we turn back to the modeling assumption of Droppo and Acero that the pitch transition probability is a function only of the absolute difference between successive pitch periods:

$$P(p_{t+1} = i | p_t = j) \approx f(|p_{t+1} - p_t|) \tag{2.15}$$

But how can we resolve such an assumption with the HMM? Basically, this is just another form of parameter tying, in which all of the states share a transition matrix, albeit one that depends on the state *values*. During learning, we simply accumulate sufficient statistics for the transition matrix over the period difference $|p_{t+1} - p_t|$ instead of the absolute state value p_t.

2.2.2 Features

Because it was easy to train and test various features for each pitch period candidate p, we experimented with a variety of possibilities. We found the best results with a combination of three. The first is the value of the normalized autocorrelation. As we

saw earlier, this will be high at multiples of the pitch period, and usually highest at the actual period. If the signal exhibits very strong periodicity, the second peak is often as high or possibly higher than the first, as we can see in figure 2-20. We will contend with this issue with the third feature.

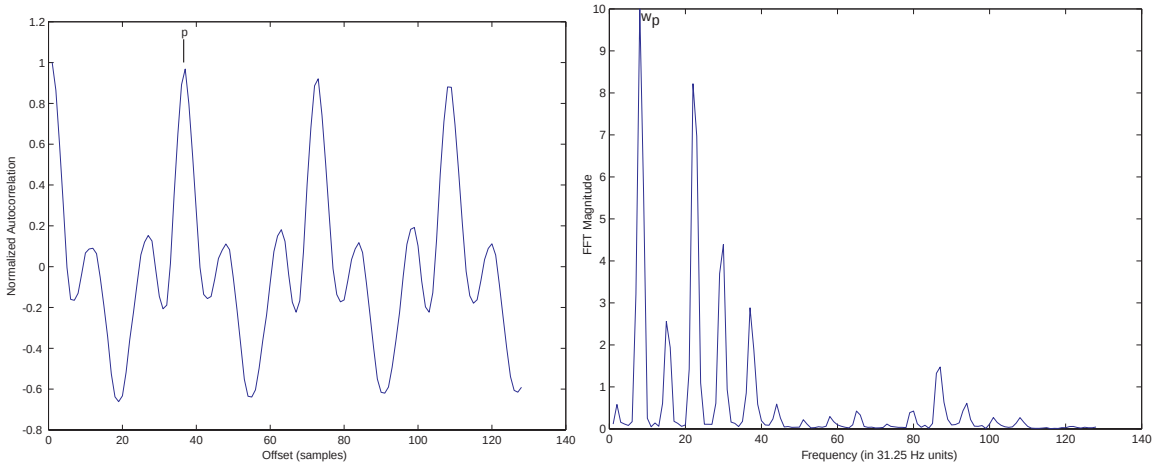

Figure 2-20: The normalized autocorrelation (left) and FFT (right) for a voiced frame. Note how the second autocorrelation peak is almost as high as the first. Also note the correspondence of the spectral peak at w_p with the true period at p.

For the second feature, we use a simple binary measure of the "peakiness" of the candidate:

$$k[i] = sgn(a[i] - \frac{1}{2}(a[i-1] + a[i+1])). \tag{2.16}$$

Though its binary nature makes it appear equivalent to removing all non-peak candidates, remember that we are still treating it probabilistically. Not all of our training pitch values will be on precise peaks, and thus the variance for this feature will be non-zero, allowing non-peak candidates to be chosen (though with low likelihood).

The final feature is in the spectral domain, and takes advantage of our knowledge of speech production. Since we know the horizontal bands in the spectrum are at multiples of the true pitch, we can use this as another piece of evidence to support or refute a particular pitch candidate. Rather than try to determine the spacing in the spectral domain, we simply look at the first peak. The frequency candidate from the N-point FFT $F[w]$ at sampling frequence f_s is related to the period value p as

follows:

$$w_p = \frac{f_s}{p}\frac{N}{f_s} = \frac{N}{p}. \tag{2.17}$$

The feature we use, then, is $F[w_p]$ normalized by the max of $F[w]$. This feature greatly adds to the robustness of our solution. This is because it significantly reduces the likelihood of the candidates at the second and third peaks of the autocorrlation, $2p$, $3p$, and so on, since the values of $F[w_p/2]$ and $F[w_p/3]$ are very small (see figure 2-20).

2.2.3 Performance

We trained the model using these features over 2000 frames of voiced speech from one speaker in the callhome database with all of the pitch values labeled. The resulting model was then used with a variety of different mics for many speakers without retraining the model in any way. Figure 2-21 shows the pitch tracking result in the autocorrelation and frequency domains.

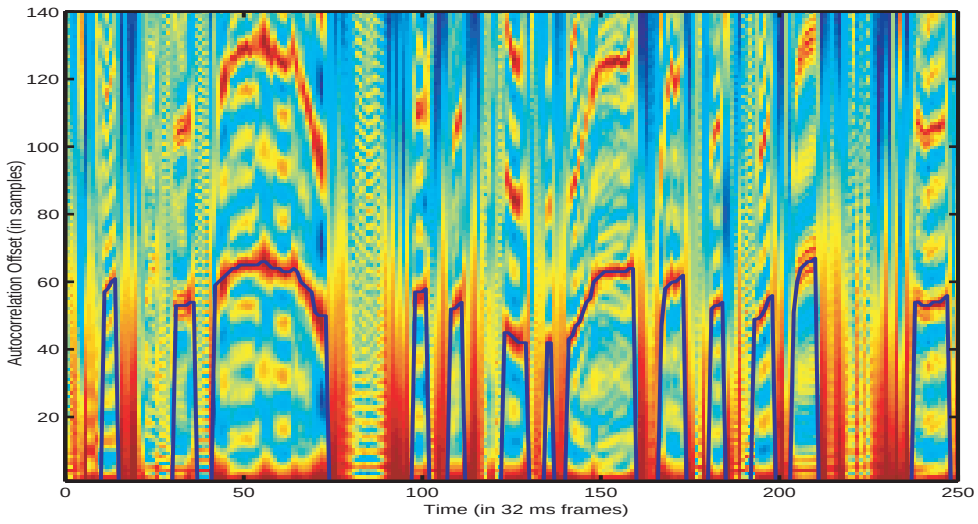

Figure 2-21: Performance of the pitch tracking algorithm in the autocorrelogram.

tTo compare against the published results of Ahmadi and Spanias [1], Droppo and Acero [11], and Wang and Seneff [33], we compute several different measures of

performance on a sequence of 8kHz test data taken from a desktop vocal microphone about 10cm from the speaker's mouth. Note that all three methods are tested only over the set of correctly identified voiced frames. The first is the "Weighted Gross Pitch Error," which Ahmadi and Spanias define as

$$WGPE = \frac{1}{K}\sum_{1}^{K}\left(\frac{E_k}{E_{max}}\right)^{\frac{1}{2}}\left|\frac{f_k - \hat{f}_k}{\hat{f}_k}\right|, \tag{2.18}$$

where K denotes the number of pitched frames that we have correctly detected as being voiced, f_k is the actual frequency in frame k, and $\hat{f}_k$ is our estimated frequency. We show the result of our algorithm and that of Ahmadi and Spanias in figure 2-22. Notice how slowly the performance drops with noise, even at less than -30dB – once again, this will be very useful to us when applying our model in noisy conditions.

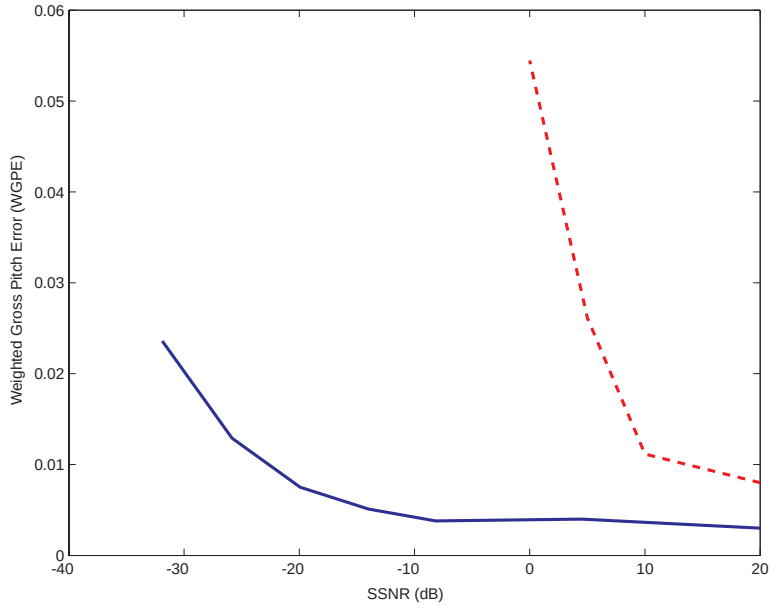

Figure 2-22: Weighted Gross Pitch Error (WGPE) for our model (solid line) vs. Ahmadi and Spanias (dashed line) vs. SSNR (dB).

To compare to the work of Wang and Seneff, we compute the Gross Pitch Error, which is the same as the WGPE but without the weighting:

$$GPE = \frac{1}{K}\sum_{1}^{K}\left|\frac{f_k - \hat{f}_k}{\hat{f}_k}\right| \tag{2.19}$$

The results for this quantity are shown in figure 2-23. Wang and Seneff report an

error of 0.0425 for studio-quality speech (greater than 30dB SSNR), and 0.0434 for telephone quality speech (about 20dB SSNR) - our model again performs significantly better. Droppo and Acero report a standard deviation in pitch error of 0.25% on studio quality speech, compared to our result of 0.13% on 20dB speech. The results for all three algorithms are shown in table 2.2 below.

Table 2.2: Comparison of Gross Pitch Error (GPE) for various pitch tracking algorithms on clean speech.

Our method	Droppo and Acero	Wang and Seneff
0.13%	0.25%	4.25%

There is one important caveat brought up by Wang and Seneff about telephone speech. Because of the characteristics of the telephone channel, speakers with a very low pitch can be very difficult to track, as their fundamental w_p can be greatly reduced or even missing. In these rare cases, we have seen our pitch tracker halve its estimate of the pitch period as a result of following the second spectral peak. The simplest way to deal with these cases would be to loosen the third feature so that it allows frequency values at the second or third multiple of the pitch candidate, which would then bring back the possibility of erroneously doubling the pitch period. A better solution would be to estimate the spacing between the spectral peaks and use this as the feature – though more noisy, it would generalize better to this difficult case.

In summary, the point of this exercise was not so much to develop a pitch tracker that exceeded the current state of the art – this is simply a happy consequence. Our real goal was to develop a method that we could train completely from data and that was robust to varying noise conditions, both of which we have satisfied with our method.

2.3 Speaking Rate Estimation

Another important characterization of a speaker's style is the speaking rate. During spontaneous speech (i.e., not reading or dictation), people tend to be bursty in the

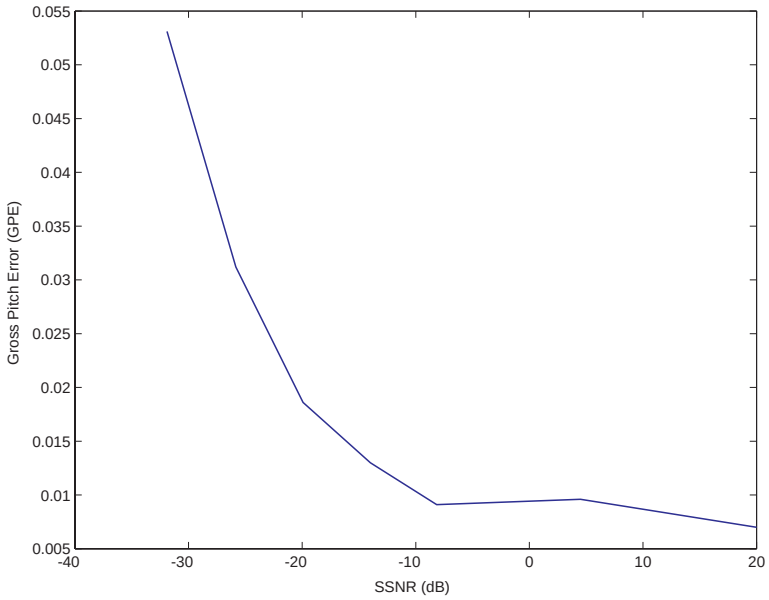

Figure 2-23: Gross Pitch Error (GPE) for our model vs. SSNR (dB).

production process – they will put out a long string of phonemes/words, then pause, then say some more, and so on. Speaking rate is thus characterized with two pieces: the *articulation rate*, which is how fast the speaker is producing phonemes during a productive string, and the *production rate*, which is how fast the speaker is moving from burst to burst. The articulation rate tends to be constant among speakers during natural speech, while the production rate varies greatly according to cognitive load, mood, and so on.

To determine the articulation rate, we look to the work of Pfau and Ruske [12], who showed that the rate of voiced segments was very strongly correlated with the phoneme rate. We thus compute this quantity over the speech segments to give us an estimate of the articulation rate. To characterize the production rate, we look at the gaps between identified speech segments in our model.

We illustrate the results of this technique on two sets of data. In the first set, one speaker is reading a paragraph at different speeds. Because this is read speech, there are almost no production gaps, and only the articulation rate is relevant. The speaker was able to read the paragraph in as little as 21 seconds and as slowly as 46 seconds. The resulting voice segment rates are shown against the length of the

reading in figure 2-24. As we had hoped, the result is an almost linear relationship between speaking time and rate.

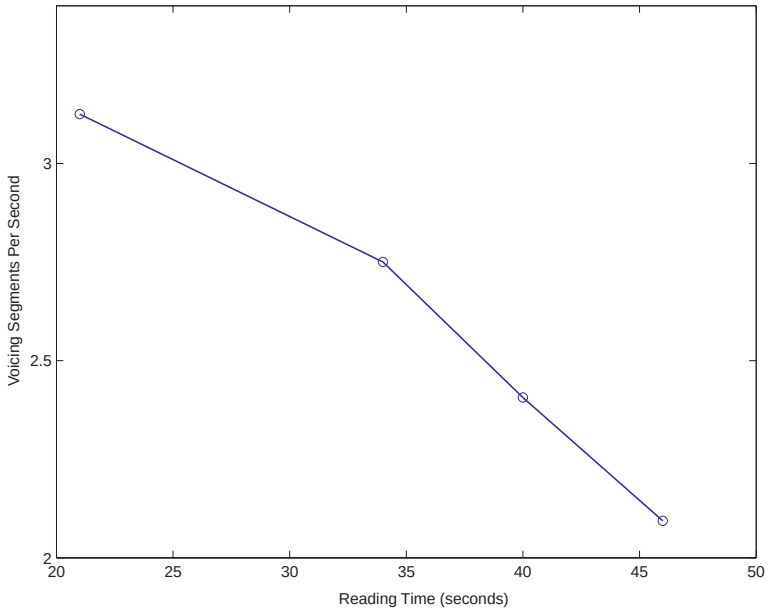

Figure 2-24: Estimated articulation rate, in voiced segments per second, for a paragraph read at varying lengths. The x-axis shows the number of seconds taken to read the paragraph. Note the desired linear relationship between the estimated rate and the time.

The production rate is more difficult to provide ground truth for as it relates to spontaneous speech. We made our best attempt by scripting a short passage and then reading it with two levels of "difficulty" – in one case, the speaker acted as though he was having trouble coming up with the next phrase, while in the other, he spoke at a fairly fluid pace. The lengths and average speech gap lengths are shown in table 2.3. As desired, the mean gap length in the longer sequence is proportionally longer.

Table 2.3: Speaking gap lengths for two sequences of the same text but spoken at different speeds.

sequence length (seconds)	average speech gap (seconds)
31	7.0
64	13.7

While not an ideal test, this shows us that the speech segmentation is reliable enough to recover the varying gap lengths, and that the gap lengths make for a reasonable characterization of the production rate.

2.4 Energy Estimation

The last feature on our list is the speaking energy. In principle this is very easy to compute, and for each frame we can express it as

$$e_{raw} = \left(\sum_{i=1}^{N} h[n](s[n])^2 \right)^{\frac{1}{2}}, \tag{2.20}$$

where N is the framesize and $h[n]$ is a Hamming window of length N. The problem, of course, is that this sort of energy is very sensitive to noise conditions. Even if we only compute the energy for the frames in each voiced segment, we will see a great deal of fluctuation.

Our solution to this problem, which we will use as the basis of our next chapter, is to *integrate* the noisy energy feature over the robustly identified voiced segments. The voicing segmentation will always be our guide, for as we have seen, we can rely on it even in very noisy conditions. In this case, for a K-frame voicing segment, the regularized energy e_{reg} will be

$$e_{reg} = \frac{1}{K} \left[\left(\sum_{i=1}^{K} e_{raw}[i] \right) - e_n^2 \right]^{\frac{1}{2}}, \tag{2.21}$$

where e_n^2 is the per-frame energy of the noise, estimated by averaging the per-frame energy over the non-speech regions. Furthermore, we clamp to zero those frames where the term inside the square root would be less than zero. In figure 2-25, we see the raw energy signal at 20dB (original signal) and -13dB of SSNR and also the regularized energies integrated over the voicing segments. While the regularized energy in the noise case does not match the original case exactly, the resemblance to the clean signal's energy is quite close, despite the very significant level of noise. This consistency allows us to compare speaking energies across very different microphone

placements and environments. Note that in the areas where the speech energy was overwhelmed by noise, we see a zero in the regularized energy. However, this does not mean we discard the voiced chunk – it simply means that when we use this feature, we have to recognize that such a chunk was of lower power than the ambient noise.

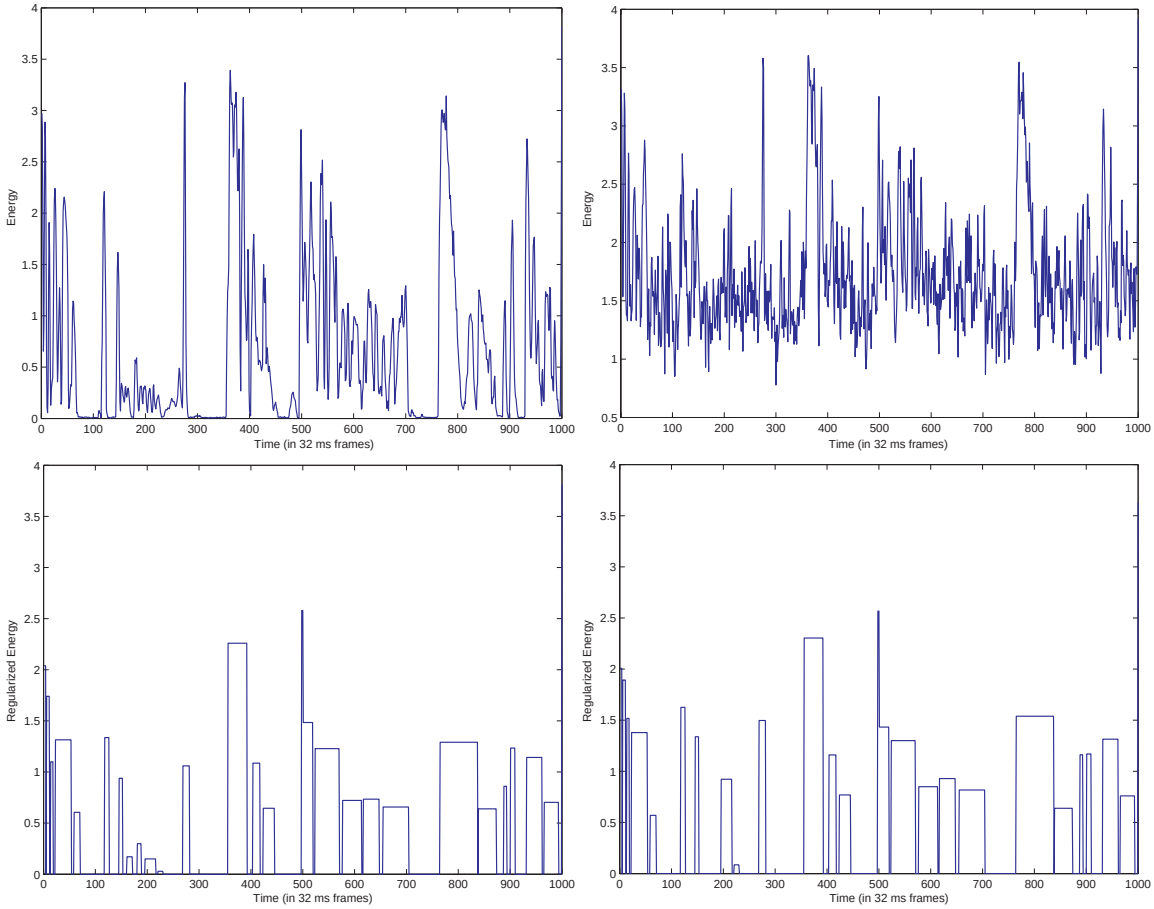

Figure 2-25: Raw energy (top) and regularized energy (bottom) for an SSNR of 20dB (left) and -13dB (right). The integration over voicing chunks allows for a consistent energy estimate despite the presence of significant noise.

2.5 Moving On

At this point, we have methods to determine all of the speech features we were interested in: the presenece of speech and voicing, the pitch, the speaking rate, and the energy. We can now move on to separating speakers and then to conversations and conversational scenes.

Chapter 3

Speaker Segmentation

In order to deal with conversational scenes in real-world settings, we will often have to separate out who is speaking when. The difficulty of this will vary with the microphone situation. We will examine three cases: in the first, there is a microphone on each person, which makes things easier but not trivial, as the speakers will often be in close proximity to each other. In the second, only one of the speakers has a microphone, and we have to segment their speech from that of their conversational partners. In these cases, we must use combinations of the energy from the different microphones to discriminate the speakers. In the third case, we will have two (or more) synchronized microphones, and can use direction-of-arrival (DOA) estimation to separate the speakers.

In all of these situations, due to the nature of conversations, there will be times when multiple speakers are speaking at the same time. In the cases where we are *choosing* between two speakers, we will not include such frames in our evaluation, as either answer would be correct. In the cases where we are detecting whether or not a target speaker is speaking regardless of what other speakers are doing, we *will* count these frames since there is a clear correct answer.

3.1 Energy-Based Segmentation

The segmentation of speakers by energy has not received much attention in the literature, as it has rarely been a case of interest – again, typical speech systems assume a close-talking microphone in a quiet room. As a result, the signal power from the user's speech completely overwhelms all other (assumed small) signals. While simple frame-based energy thresholding can be effective in this case, even here it is necessary to use hysteresis in the speech/non-speech decision in order to prevent clipping the beginnings and ends of voiced segments [28].

As computers and microphones become increasingly ubiquitous, though, we see them as becoming part of our clothing and being worn further away from our mouths. Already we see policemen and couriers carrying radio devices on their shoulders. In such cases, the microphone is often six inches or further from wearer's mouth, whereas their conversational partner may be as little as two feet away. This means the signal power ratio is 16:1 for speaker to partner following the $\frac{1}{r^2}$ power law, and the magnitude ratio is only 4:1. When we then consider additive noise, the problem becomes quite challenging. We will show how we can deal with this case by again resorting to our voicing segmentation mechanism, once again using it to integrate over the noisy features. We will show results for when either one or both of the users are wearing microphones.

3.1.1 Energy Segmentation in Noise with Two Microphones

When both speakers are wearing microphones, we can use the log energy ratio between the microphones as our discriminative feature:

$$r_e[i] = \log(\frac{e_1[i]}{e_2[i]}) = \log(e_1[i]) - \log(e_2[i]) \tag{3.1}$$

for each frame i. In the experiments below, we have taken two separate streams of telephone-quality audio from the callhome database and mixed them at a 4:1 ratio before adding Gaussian noise. We then segment the audio in two ways: first using the raw r_e value for each frame, then using the r_e value averaged over each voicing

segment. To compute an optimal threshold, we first find the ROC curve for each detector and then choose the threshold t^* such that we minimize the overall error for both misses and false alarms:

$$t^* = \arg\min_t \left((1 - P_D(t))^2 + P_{FA}(t)^2 \right)^{\frac{1}{2}}, \tag{3.2}$$

where $P_D(t)$ is the probability of detection (the fraction of frames from the target speaker that we correctly classify) for a given threshold and $P_{FA}(t)$ is the probability of a false alarm (the fraction of frames not from the target that we incorrectly classify). Note that we are not considering voicing segments in which there was more than a 50% overlap between the two speakers, since this is a situation where we are choosing between the speakers and either answer would be correct. Furthermore, in this and all other experiments, we give the raw energy method a a boost by giving it the benefit of our voiced/unvoiced decision – we only evaluating its classification in the voiced regions. The results of the comparison are shown in figure 3-1. As we expected, regularizing the energy with the voicing segments makes for a significant improvement in segmentation performance. If we are willing to accept more false alarms, we can achieve even higher performance. We can plot the probability of detection versus the probability of a false alarm for various threshold values in a region-of-convergence (ROC) plot. Such plots are shown in figure 3-2 for SSNRs of 0.89 dB and -14.7 dB.

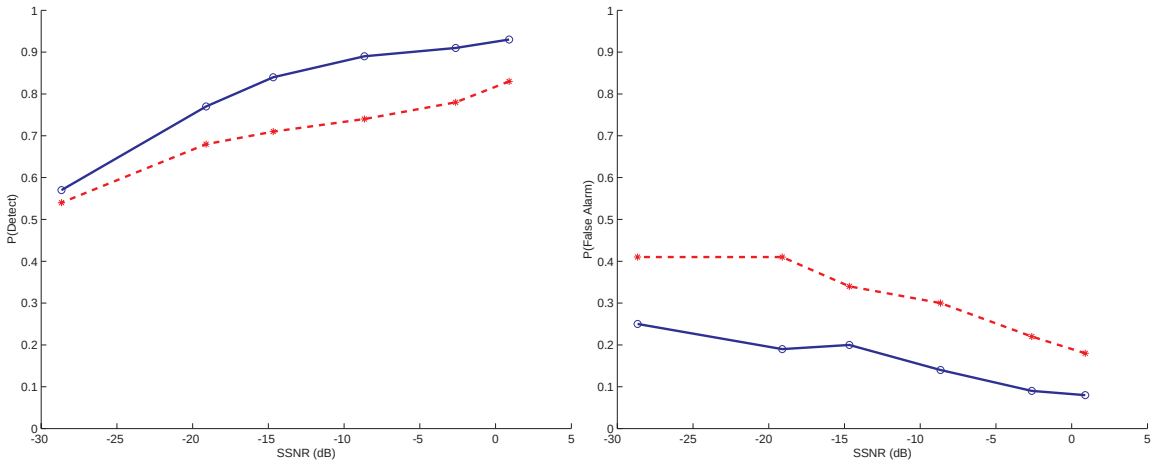

Figure 3-1: Speaker segmentation performance for our model (solid line) and raw energy (dashed line) with two microphones where the signals are mixed at a 4:1 ratio with varying amounts of noise. The plots show the probability of detection vs. SSNR level (left) and probability of false alarm vs. SSNR level (right) using optimal threshold choices.

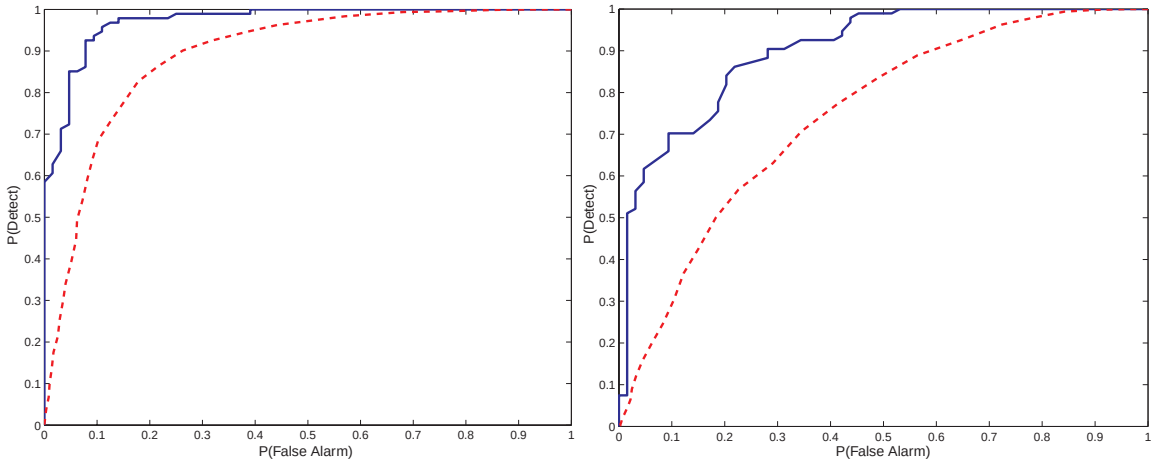

Figure 3-2: ROC curves for speaker segmentation performance with two microphones for our model (solid line) and using the raw energy (dashed line) where the signals are mixed at a 4:1 ratio. The plots show the ROC curves for an SSNR of 0.89 dB (left) and -14.7 dB (right).

To some degree, though, a per-frame error metric does not truly do our method justice. Just as in the case of voicing segmentation, it makes a big difference whether we can provide smooth results or if our estimate is jumping between possibilities on every frame. Note that we could flip back and forth every third frame and still yield 75% performance, but this would yield a far more different qualitative result than getting 75% of the voicing chunks correctly.

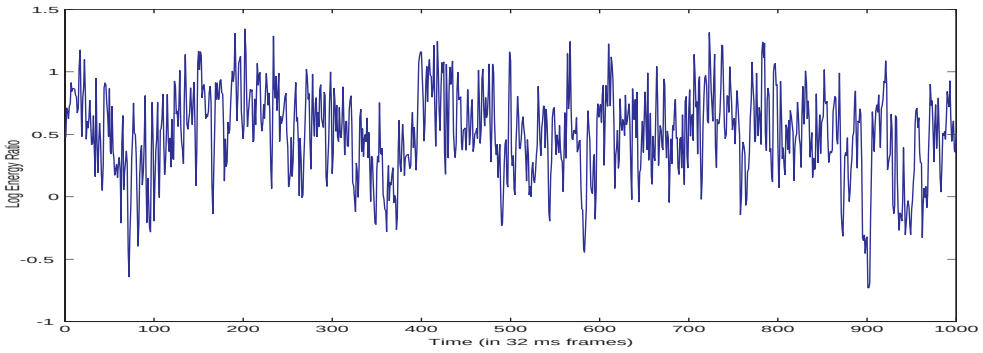

Figure 3-3: The raw per-frame energy ratio for part of our test sequence at an SSNR of -14.7 dB. Note how difficult it is to tell when the different speakers are speaking.

If we look at the raw r_e values for an SSNR of -14.7 dB in figure 3-3, it quickly becomes clear that no threshold could give us a very smooth result. We show this explicitly in figures 3-4 (raw) and 3-5 (regularized), where we show the segmentations produced by the raw energy versus our method. As expected, our method gives far smoother results since it is making its decision by integrating information over the voicing regions. The results from the raw energy, on the other hand, are quite unusable.

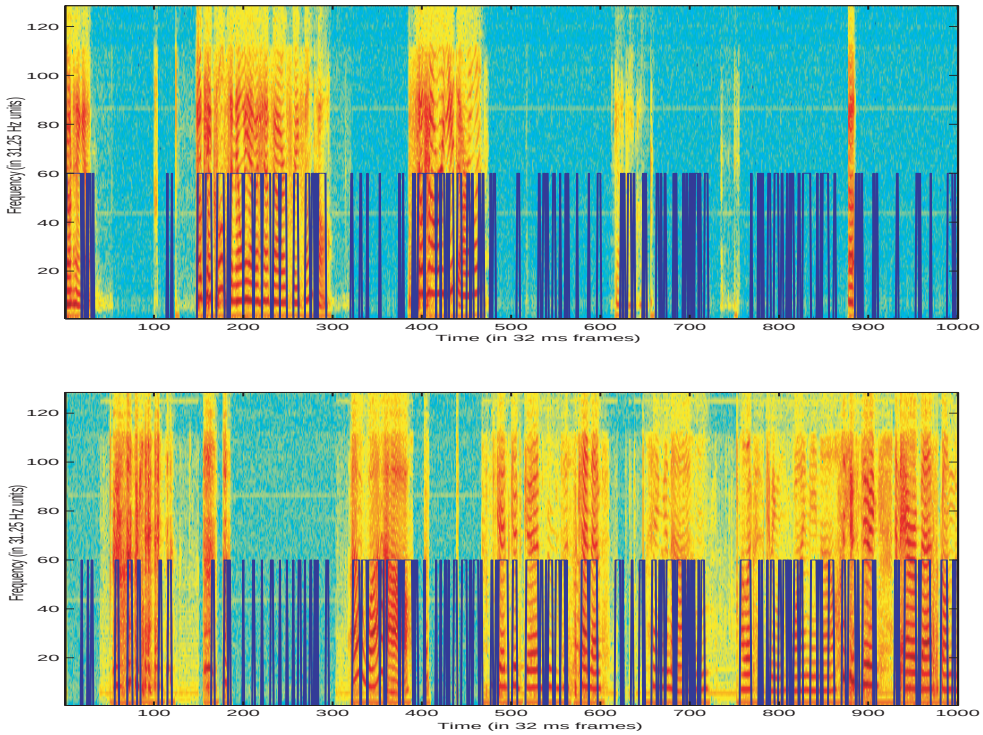

Figure 3-4: The speaker segmentation produced by using raw energy for speaker 1 (top) and speaker 2 (bottom) using two microphones mixed at a 4:1 ratio and at an SSNR of -14.7 dB. The segmentation is overlaid on the original, noise-free, unmixed spectrograms of the target channels for clarity.

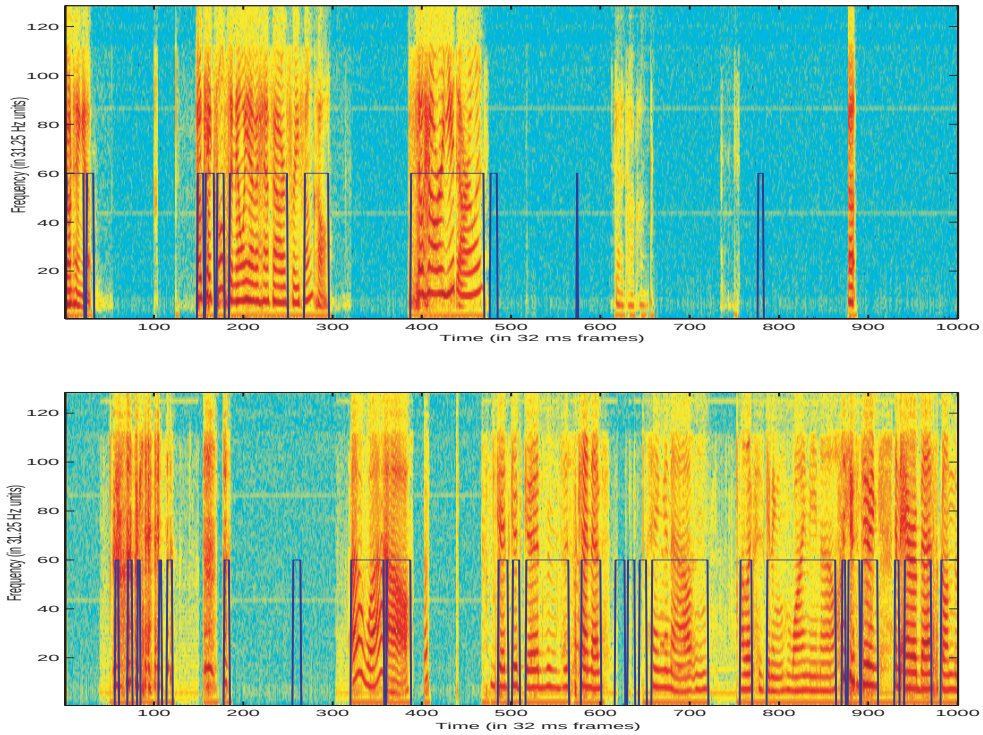

Figure 3-5: The speaker segmentation produced by using our regularized energy approach for speaker 1 (top) and speaker 2 (bottom) using two microphones mixed at a 4:1 ratio and at an SSNR of -14.7 dB. The segmentation is overlaid on the original, noise-free, unmixed spectrograms of the target channels for clarity.

3.1.2 Energy Segmentation in Real-World Scenarios with One and Two Microphones

Though the conditions we presented in the previous experiments were rather harsh (4:1 mixing with up to -14.7 dB of SSNR), the noise sources were still synthetic. As a reuslt, we wanted to test our techniques in a real setting. We collected a half hour of data with three subjects each wearing a sociometer. Subjects 1 and 3 had a ten minute conversation, after which all three subjects had a 5 minute conversation. All of the data was taken in an open work environment with other voices in the background, typing sounds, etc. After examining the data, we realized that the microphone of one of the subjects (subject 3) was not working properly and produced no usable data. This is important to keep in mind, since in the two-microphone case (subjects 1 and 2) some of the data was coming from subject 3, and it was not possible to use her data to account for these voice segments. The ROC curve we present in figure 3-6 is for detecting frames coming from speaker 1, using all possible thresholds on the log energy ratio between speakers 1 and 2. Frames from speaker 3 mislabeled as coming from speaker 1 are thus considered among the false alarms. Furthemore, in this experiment, since we are not choosing between two speakers, we count *all* frames produced by speaker 1, whether or not there was overlap from the other speakers.

Our method works quite well for this scenario, which is not surprising given that the speaker separation (about 4 feet to speaker 2 and 6 feet to speaker 3) and the noise conditions were much gentler than in our earlier experiments. The raw energy still does not work very well – though we can get quite a high probability of detection, it is only at the expense of a significant number of false alarms.

We now move to the one-microphone case, where we use only the energy from speaker 1's sociometer to decide whether the signal is coming from him or from one of the other speakers. For our method, then, we will be integrating this over the voicing segments, whereas for the raw method, we label each frame individually according to its energy. The ROC curves for the probability of detection and false alarm for all possible thresholds are shown in figure 3-7. Again, as we are detecting

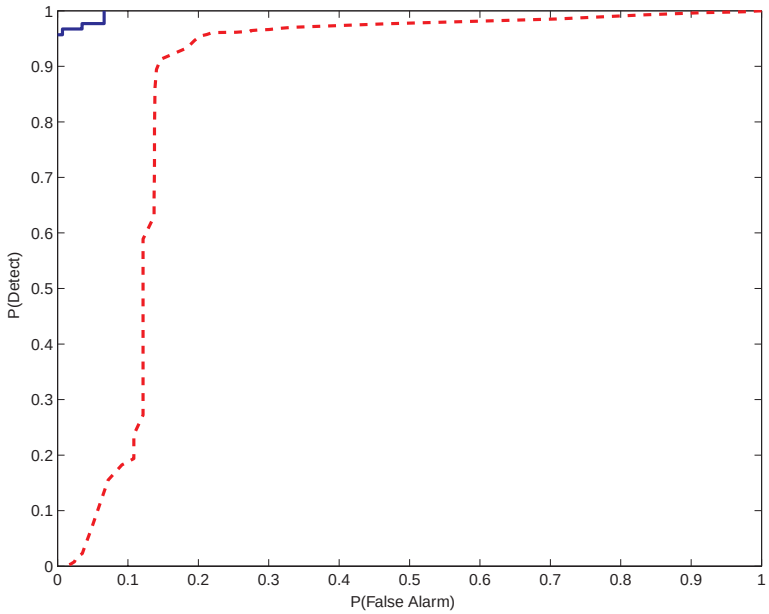

Figure 3-6: ROC curves for speaker segmentation performance with our method (solid line) and with raw energy (dashed line) using two sociometers where the speakers are about five feet apart.

whether the target speaker is speaking, we include frames with overlapping speakers in our evaluation. Our method still performs very well, though noticeably worse than in the two-microphone case. We can still get a probability of detection of greater than 90% with a less than 5% false alarms. Once again, our method performs significantly better than the raw energy.

3.2 DOA-Based Segmentation

The prospect of using phase/direction-of-arrival (DOA) information for speaker segmentation has been better explored than energy. The basic idea here is to use the offsets between the arrival times of the sound at the different microphones to reconstruct the direction the speech came from. The work of Khalil et al. [19] and that of the PictureTel corporation are good examples: both use DOA estimation to steer a camera to the current speaker in the context of a teleconferencing system. As a result, the system averages the estimation over several seconds, and only moves the camera once the mean estimated direction has become stable. This is fine for a tele-

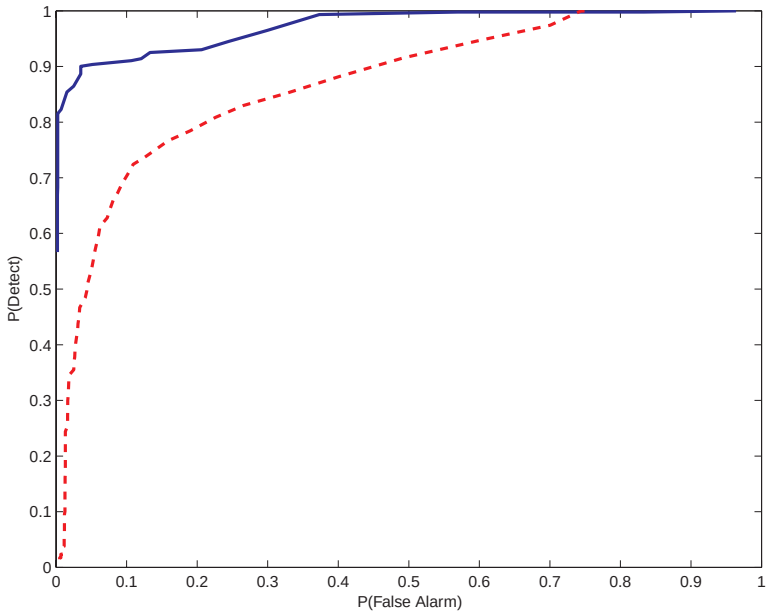

Figure 3-7: ROC curves for speaker segmentation performance with our method (solid line) and with raw energy (dashed line) using the energy from only one sociometer.

conferencing system – we would not want the camera to jump from person to person for every short "uh-huh" or "ok." For our purposes, though, we would like to get as many of these changes as possible, as they can give us significant information about the progress of the interaction. We will show how we can do this by computing DOA information over entire voiced segments instead of individual frames.

3.2.1 Computing the DOA

The basic tool for computing the DOA is the normalized cross-correlation. For two signals $s_1[n]$ and $s_2[n]$ of length N, we can write this as:

$$c[k] = \frac{\sum_{n=k}^{N} s_1[n]s_2[n-k]}{(\sum_{n=0}^{N-k} s_2[n]^2)^{\frac{1}{2}}(\sum_{n=k}^{N} s_1[n]^2)^{\frac{1}{2}}} \tag{3.3}$$

If two signals match exactly at some offset k but have unequal powers, $c[k]$ will have a value of one. While this is quite intuitive and easy to compute, it is subject to a number of constraints. First of all, we must consider the spacing of the microphones, l. Since sound travels at a velocity v_s (331 m/s at room temperature), it travels v_s/f_s in every sample. Then the distance between the microphones in samples is

$$c = \frac{l f_s}{v_s}, \tag{3.4}$$

while the total correlation range is

$$c_r = 2\frac{l f_s}{v_s} + 1. \tag{3.5}$$

The factor of two and the 1 come because the sound could be coming from either direction. The correlation range is often quite small, particularly at the low sampling rates we are working at (8 kHz). Furthermore, there is a constraint on the maximum frequency f_{max}, since if the period of an incoming signal (in space) is less than twice the microphone spacing, we will have aliasing: for example, a period of precisely the microphone spacing will result in a maximum at $c[0]$ when the signal is coming along the vector of microphone 1 to microphone 2, the opposite direction, or along their perpendicular bisector. As a result, we need to use the constraint

$$f_{max} = \frac{f_s}{p_{min}} = \frac{f_s}{2c} = \frac{v_s}{2l}. \tag{3.6}$$

Furthermore, the cross-correlation signal tends to be quite noisy, as can be seen in the first panel of figure 3-9. This is primarily due to acoustic reflections and the variations between the microphone responses. The typical methods all involve finding the frame-based cross-correlation peaks and then regularizing them using some dynamic constraints. The problem, as we mentioned earlier, is that such constraints are either (1) so strong that they smooth over speaker changes or (2) so weak that many frames of spurious noise in the cross correlation are misclassified as the sound coming from some other location.

As before, our approach will be to combine information over the whole of a voicing segment, as for the most part the voicing segments are coming from separate speakers. In this case, though, instead of averaging the noisy frame-based values over the voicing segment, we will instead compute the cross correlation over the *entire* segment. It is well known in the DOA community that longer sequences result in more accurate DOA computations; the difficulty is in knowing what segments to use. In our case,

the choice of regions is clear from the voicing segmentation.

There is one additional improvement we have made due to these longer correlations. Since the signal power can vary a great deal among the different frames, a high energy frame could dominate the correlation computation and thus undo the advantage of using many frames. We thus normalize each frame-sized chunk of the target signal by its energy, resulting in two signals with instantaneous power approximately equal to their average power. We will show the marked improvement this small change makes.

3.2.2 DOA Segmentation with Two Microphones

In the experiments below, we had two microphones spaced at 0.4m and sampling synchronously at 8kHz. This resulted in a total correlation range of 15 samples (7 in each direction plus 0), and a maximum frequency of 413 Hz, which is fairly low but sufficient for the speakers in our experiment. The approximate geometry of the speakers with respect to the microphones is shown in figure 3-8.

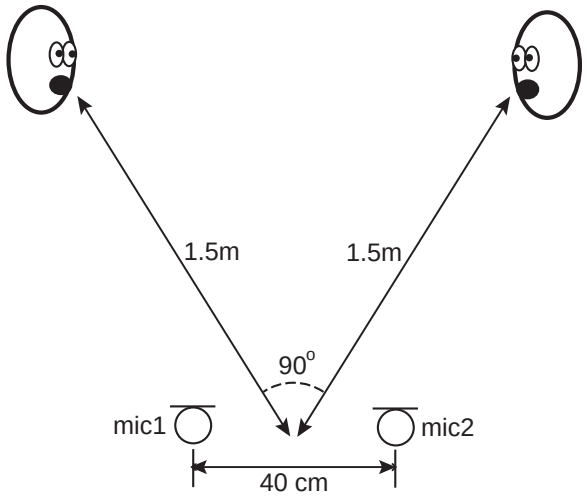

Figure 3-8: The microphone geometry for the DOA-based segmentation experiments.

In figure 3-9, we show a plot of the peaks of the original cross-correlation signal and the results of computing the correlation over the voicing segments. While our result is still somewhat noisy, it is far cleaner than the original. It is fairly clear from the original why it is so difficult to assign a workable dynamic constraint to this

problem.

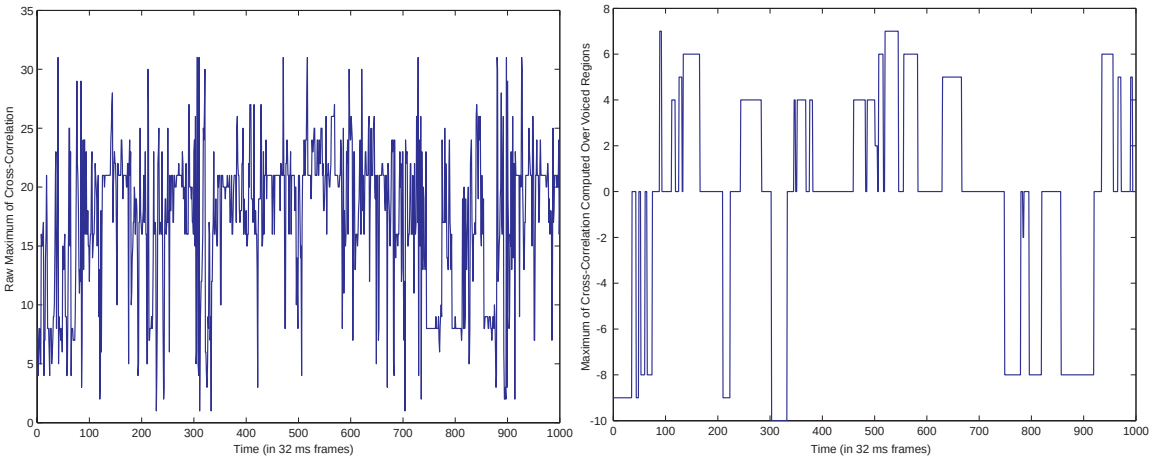

Figure 3-9: The peaks of the normalized cross-correlation over time. The left panel shows the per-frame correlation as is used in typical approaches; the right panel shows our regularization via computing the correlation over entire voicing segments.

In figure 3-10, we show the ROC curves for our method versus the raw per-frame correlation peak, both without (left) and with (right) our energy normalization. Once again, since we are choosing between speakers 1 and 2, we do not include frames where both are speaking in our evaluation, as both answers would be correct. Furthermore, to make a more fair comparison, we give the raw phase method the benefit of knowing where the voicing segments occur, so it is not penalized for spurious values that do not correspond to voicing. Regardless, in both cases, we show a marked improvement in terms of an optimal threshold, but with the normalization there is a significant additional gain over the entire course of the ROC. With our method, we can achieve an almost 90% probability of correct detection with less than 1% false alarm rate. Once again, the other advantage of our method is that we know it will produce *smooth* results, i.e., we will prevent the location estimate from jumping about without putting artificial dynamic constraints on the interaction.

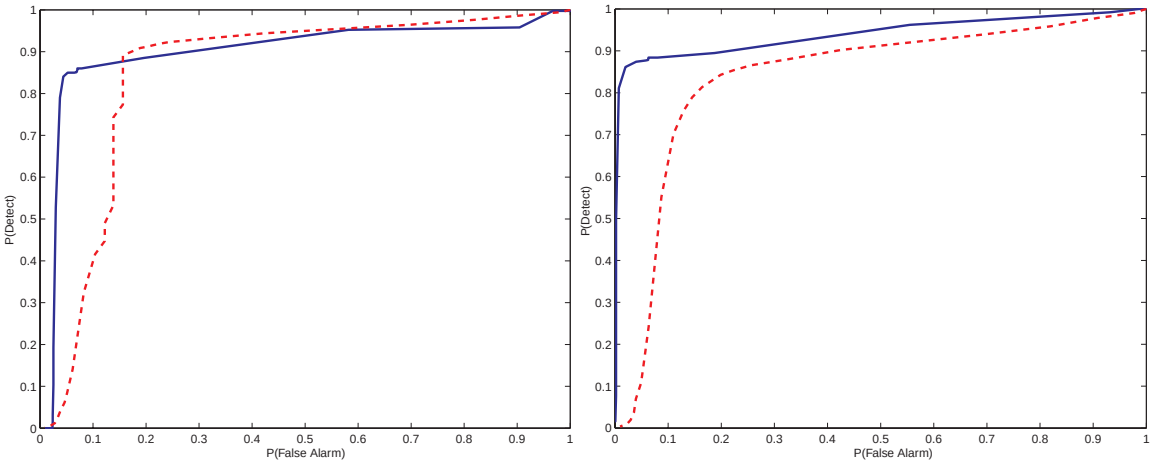

Figure 3-10: Comparison of ROCs for DOA-based speaker segmentation using our method (solid line) without (left) and with (right) energy normalization, compared against using raw DOA estimates (dashed line). Note the raw method is not penalized for misclassifications in unvoiced frames.

Chapter 4

Finding Conversations

One of the many questions we had about the nature of conversations was what signified a conversation. In other words, what was it about two streams of audio that made them part of the same conversation? Clearly, it must have something to do with the sychronony of the two streams, i.e., the inherent dynamics of the conversation process. In this chapter, we investigate this phenomenon in detail.

To formalize the notion of synchrony, we chose to look at the per-frame voicing segmentation values and see how predictable they were from each other. Did the fact that one speaker was speaking decrease the probability of the other speaker speaking, or did it not affect the other at all? The natural measure for this type of interaction is the mutual information [7], and we compute our alignment measure $a[k]$ for an offset of k between the two voicing streams as follows:

$$a[k] = I(v_1[t], v_2[t-k]) \tag{4.1}$$

$$= \sum_{i,j} \log p(v_1[t]=i, v_2[t-k]=j) \frac{p(v_1[t]=i, v_2[t-l]=j)}{p(v_1[t]=i)p(v_2[t-k]=j)}, \tag{4.2}$$

where i and j range over 0 and 1 for unvoiced and voiced frames, respectively. As we expected, this measure produced a peak at the correct alignment of two streams. What we did not expect and what we will show in this chapter is how strong and how robust that peak would be. We show how this measure can be used to find and

precisely align conversations from separated audio streams from among thousands of false candidates with very high accuracy, even in the presence of significant noise. We also show the results from real-world data in which the streams are not perfectly separated.

Our measure is quite simple, and it is really the dynamics inherent to human interactions that allow us to achieve the performance we report. Furthermore, the underlying robustness of our voicing/speech segmentation model allows us to use this method even when the signals are quite noisy. We conclude this chapter by describing some possible applications of this interesting new result.

4.1 Finding Conversations from Separate Streams

For the first set of experiments, we again used speech from the callhome database. To some degree, we did not expect this data to be perfectly aligned to begin with, as all of the conversations are international phone calls and are thus subject to some delay. In figure 4-1, we show the values of $a[k]$ for various ranges of k over two-minute segments of speech (7500 frames): in the first plot, each tick corresponds to one frame, so the entire range is 1.6 seconds; in the second, each tick corresponds to 10 frames, so the range is 16 seconds; in the third, it is 100 frames and 160 seconds (about three minutes), and in the final, it is 1000 frames and a total range of 1600 seconds or about half an hour. As expected, the peak over the 1.6 second range is fairly broad, as there is some delay in the channel. Furthermore, as we described in the speaker segmentation results, the speakers sometimes overlap in their speech. The remaining ranges, though, are quite remarkable – even over a half hour range, the peak for the correct alignment is very strong. In fact, the second peak we see in this case is an outlier: as will be evident from our ROC curves, there are very few false alarms for the correct alignment.

To see why this works so well, we must examine the probability tables for v_1 and v_2 under correct (i.e., $k = 0$) (table 4.1) and random alignments (table 4.1). In the aligned case, notice how the joint probability of both speakers speaking is 0.052, or

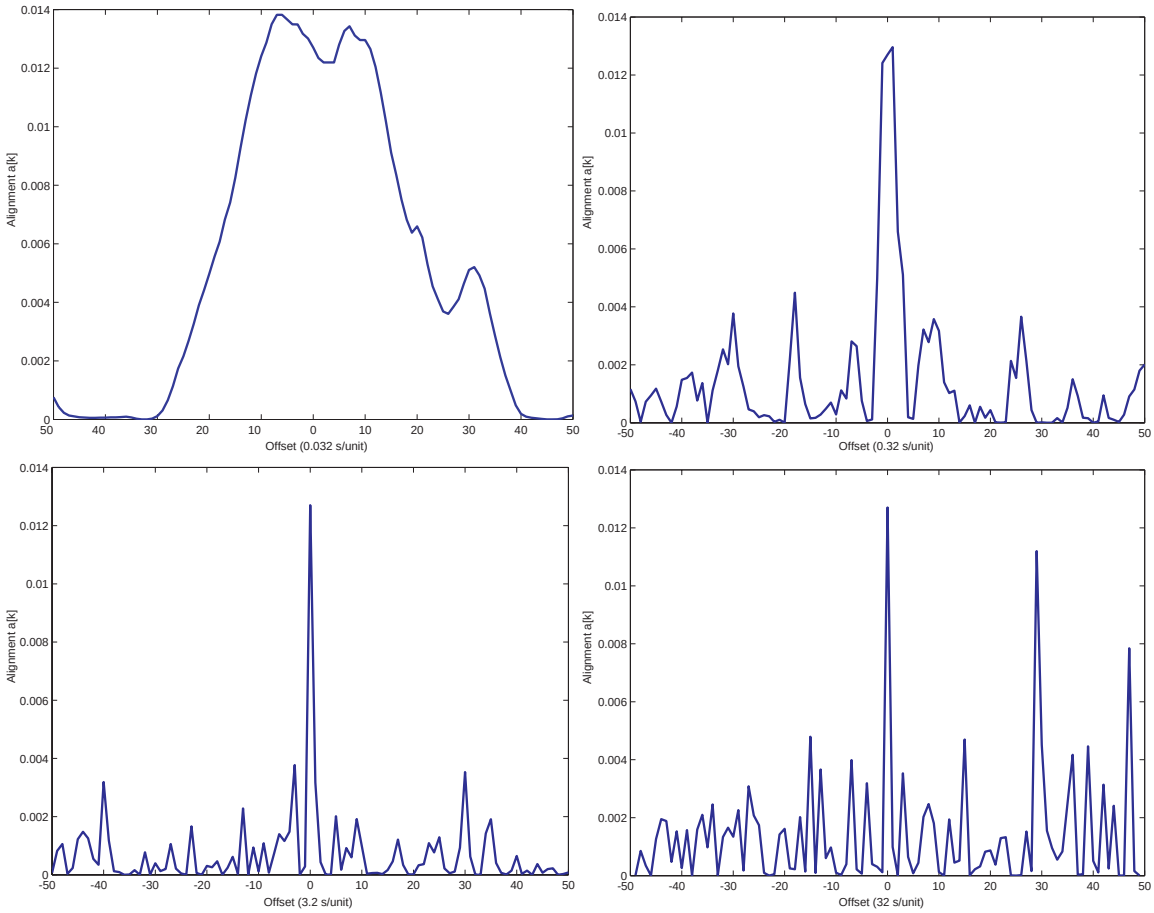

Figure 4-1: Values of our alignment measure $a[k]$ for various ranges of offset k over a two-minute segment (7500 frames) for a telephone conversation from the callhome database: in the first plot (upper left), each tick corresponds to one frame, so the entire range is 1.6 seconds; in the second (upper right), each tick corresponds to 10 frames, so the range is 16 seconds; in the third (lower left), it is 100 frames and 160 seconds (about three minutes), and in the final (lower right), it is 1000 frames and a total range of 1600 seconds or about half an hour.

almost zero, whereas in the non-aligned case, it is significantly higher at 0.119.

To further illustrate this point, we show the values of v_1 and v_2 over 1000 frames in figure 4-2. While there is some overlap, the synchronization is fairly clear. One of the surprising aspects is that often the little slivers of voiced segments, as in the 600-800 range in the figure, are in the same region but precisely offset from one another – somehow humans in conversation have the ability to synchronize very tightly in their vocalizations, a phenomenon noticed long ago by ethnographers [21]. This is why we have used the voicing segmentation instead of the speech segmentation for our

	$v_1 = 0$	$v_1 = 1$
$v_2 = 0$	0.307	0.406
$v_2 = 1$	0.234	0.052

Table 4.1: Probability table for v_1 (whether speaker one is in a voiced segment) and v_2 from the callhome data when the two signals are perfectly aligned ($k = 0$).

	$v_1 = 0$	$v_1 = 1$
$v_2 = 0$	0.394	0.319
$v_2 = 1$	0.167	0.119

Table 4.2: Probability table for v_1 (whether speaker one is in a voiced segment) and v_2 from the callhome data when the two signals are not aligned ($k = 40000$).

alignment measure. As to be expected, the latter gave far less dramatic results.

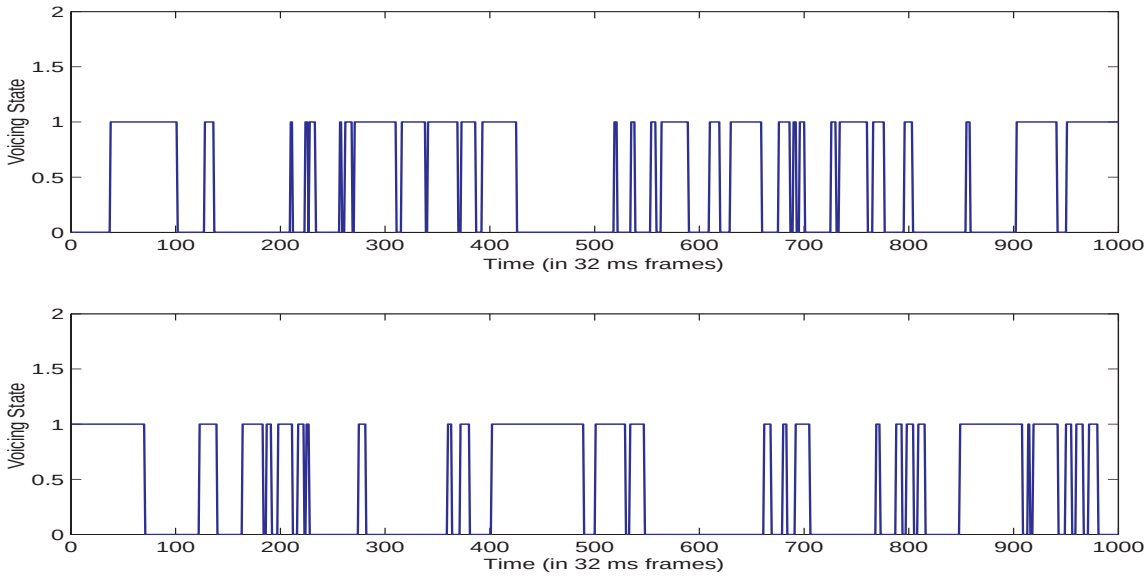

Figure 4-2: Voicing segmentations for both speakers when perfectly aligned. Note how speaker two's voicing segments are a noisy complement to those of speaker one.

We now investigate the robustness of this detection result under a variety of conditions – additive noise, different segment lengths, and different stepsizes. We first look at the ROC curves for various levels of noise in figure 4-3. These ROC's are found over four hours of speech from eight different speakers (four conversational pairs). For each conversation, eighty target locations were chosen uniformly through the conversation, and eighty test locations were chosen uniformly from the other speaker, one of which was the correct alignment. The false cases thus outweighed the correct ones at an 80:1 ratio. The performance in the noise-free case is the strongest, with a probability of detection of greater than .99 with a probability of false alarm of less than .01. Even in the worst noise scenario, at an SSNR of -20 dB, we can achieve a detection rate of greater than 90% with a false alarm rate of less than 10%.

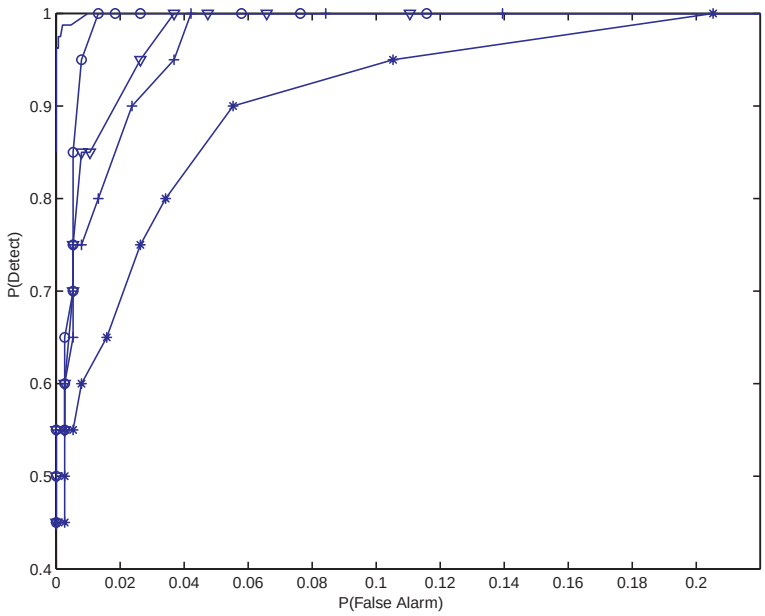

Figure 4-3: ROC curves for detecting conversations in varying SSNR conditions, tested over four hours of speech from eight different speakers (4 conversation pairs). Key for SSNR values: (-) 20 dB, (o) -12.7 dB, (v) -14.6 dB, (+) -17.2 dB, (*) -20.7dB.

One interpretation of why this works so well is that each voice segmentation pattern over a given chunk of time is like a pseudorandom sequence or key, and the correctly aligned partner's pattern is a noisy complement of this sequence. The longer the sequence, the less likely it is that any impostor sequence will be able to achieve a good fit to the original key. To investigate this further, we examined how the

performance would change with a shorter segment, as shown in figure 4-4. As we expected, the ROC curve is not as strong, but could still make for quite a reliable detector.

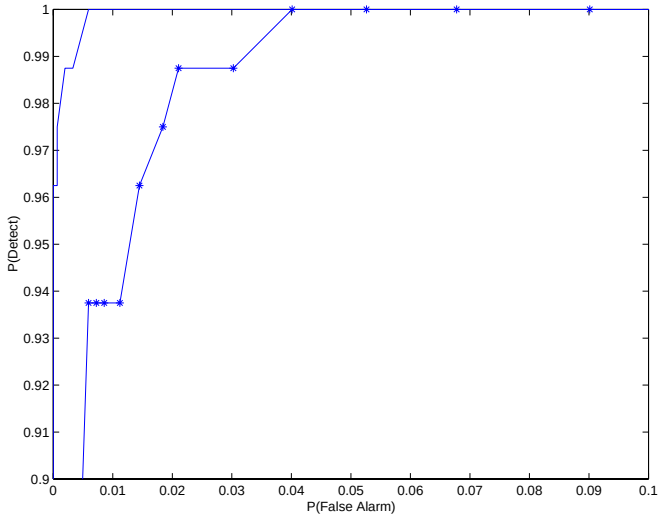

Figure 4-4: ROC curves for detecting conversations with different segment lengths: two minutes (-) and one minute (*).

Finally, in the interests of computational savings, we consider what would happen if we were not willing to look at all possible alignments and "skipped" by various values, thus preventing us from hitting the exact alignment on the mark. Naturally, this will reduce the strength of the peak, and we were interested to see how much this would affect the performance. The results are shown in figure 4-5. With a skip value of 20 frames (0.32 seconds), we haven't lost that much in performance, but with an offset of 40 (0.64 seconds) the drop is significant. Even at 20 frames, though, we only need to test three offsets per second, which makes for a much reduced computational load (1/20th).

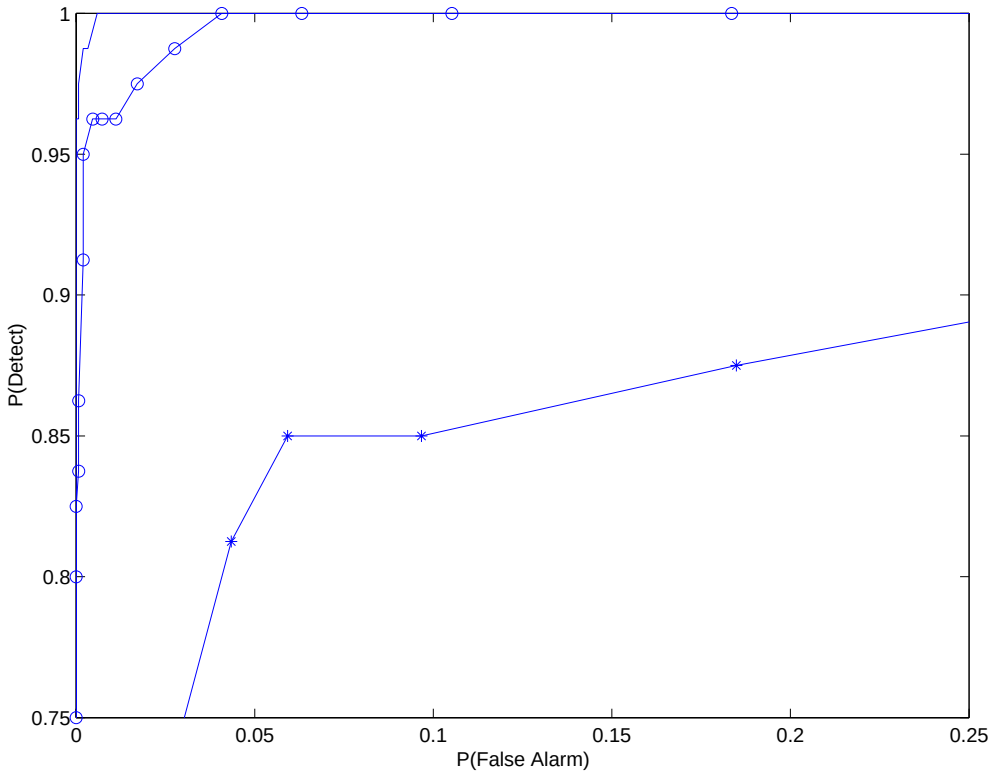

Figure 4-5: ROC curves for conversation detection at different offset skip sizes, where a skip size of 20 means we only test the alignment every 20th frame. ROCs shown are for 0 frames (-), 20 frames (o) and 40 frames (*).

4.2 Finding Conversations in Mixed Streams

To further test our method with open-air sources and noise, we tested our alignment measure on the conversation data we collected earlier with the sociometers. Speakers two and three had a half an hour and twenty minutes of data respectively, but only 5 minutes of this was spent in conversation between the two. We thus set up the same set of experiments from the previous section for this case. In figure 4-6, we show the peaks for a 1.6 second, 16 second, 160 second, and 11 minute range. This time, we have taken the alignment measure over one minute segments (3750 frames).

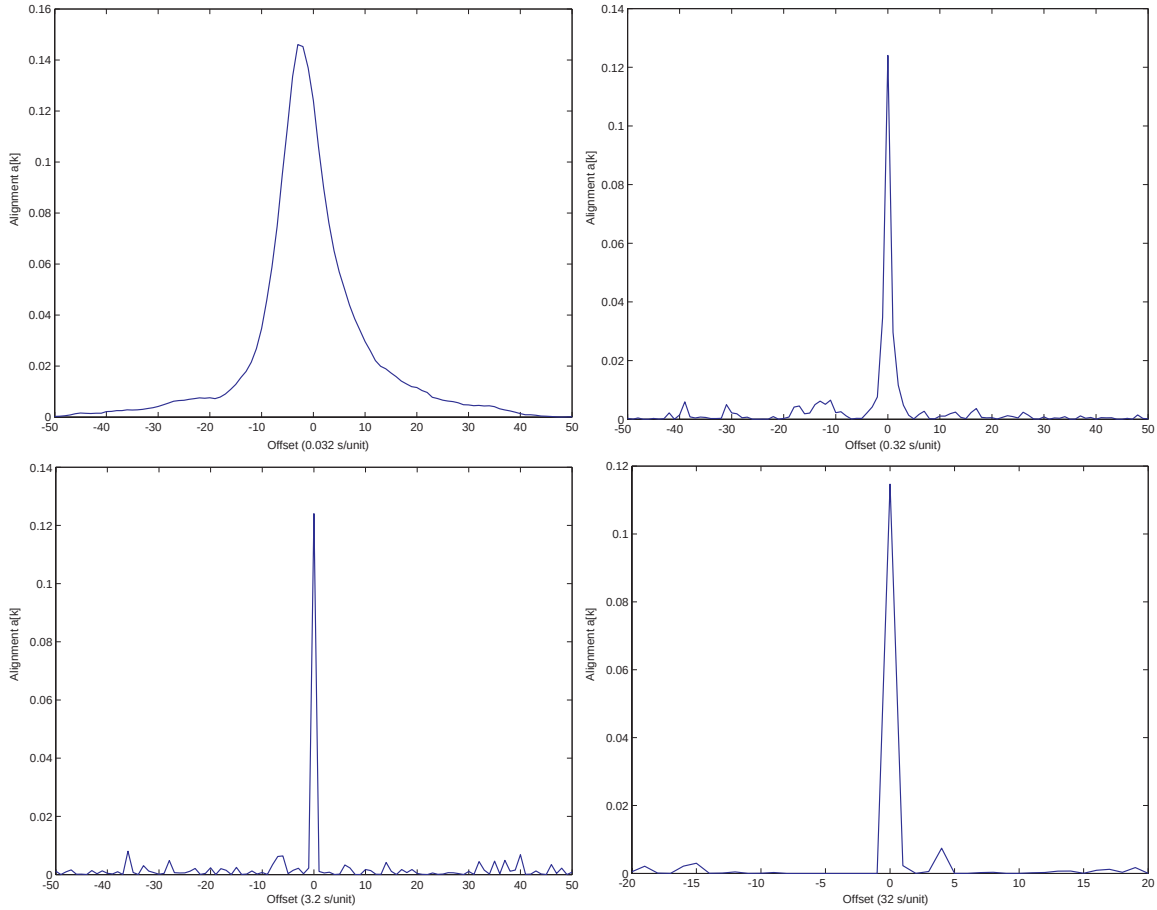

Figure 4-6: Values of our alignment measure $a[k]$ for various ranges of offset k over one-minute segments (3750 frames) for the sociometer data: in the first plot (upper left), each tick corresponds to one frame, so the entire range is 1.6 seconds; in the second (upper right), each tick corresponds to 10 frames, so the range is 16 seconds; in the third (lower left), it is 100 frames and 160 seconds (about three minutes), and in the final (lower right), it is 1000 frames and a total range of 640 seconds or about 11 minutes.

The peaks are much stronger than in the callhome data, but this should not be surprising – remember that in this case, there is significant bleed from one speaker to the other's microphone, and thus in terms of voicing segmentation the two data streams have almost identical voicing values when aligned. We illustrate this in figure 4-7. Here we see the voicing segmentations for both user's microphones when they are perfectly aligned. As expected, the signals are almost identical.

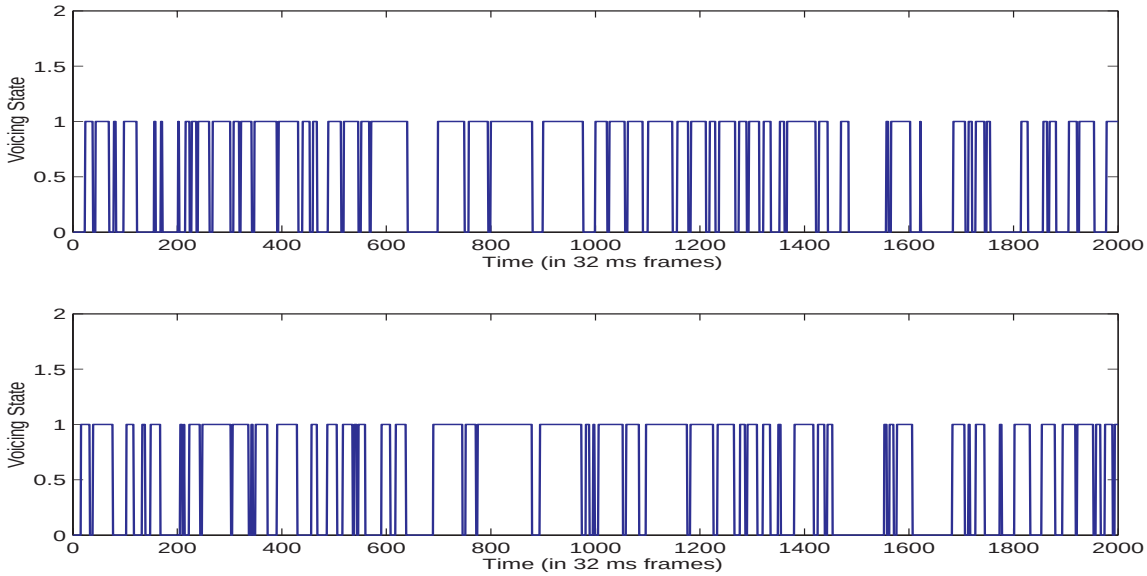

Figure 4-7: Voicing segmentations from the sociometer data for both speakers when perfectly aligned. Note how the two segmentations are almost identical, as both users' microphones are picking up both voices.

The probability tables under perfect alignment are also quite different as a result. In table 4.2 below, we show the joint probability table for v_1 and v_2 when they are perfectly aligned. Notice how the joint probability of both speakers speaking now overwhelms all of the other cases, as we saw in figure 4-7.

	$v_1 = 0$	$v_1 = 1$
$v_2 = 0$	0.243	0.146
$v_2 = 1$	0.099	0.513

Table 4.3: Probability table for v_1 and v_2 from sociometer data when the two signals are aligned. Note that $P(v_i = 1)$ only means that voicing was detected on speaker 1's sociometer; the voice signal causing this could have come from speaker 1 or speaker 2.

We note that this sort of alignment makes the solution somewhat less useful than the former case. Consider the case of two people sitting together and a third and fourth person having a conversation in their proximity. In this case, we would register a conversation between all possible pairs. This is not really what we want. If we instead use the techniques of the previous chapter to separate out the speaker's voice from the others, we will get weaker peaks but have robustness against conversations the target subject is not involved in.

We now examine the ROC curves for our data for several lengths in figure 4-8. For this test, since we had much less data to work with, we chose 9 possible target alignments (distributed over the 5 minutes of the conversation) with 71 false possibilities (distributed over the twenty minutes of the other subject's data) and 1 correct possibility for each test. For one minutes, thirty seconds, and even 15 seconds, the ROC is perfect – we can achieve a probability of detection of 1.0 with no false alarms. When we start increasing the step size, i.e., the misalignment of the best match, we begin to see a performance decrease. The ROCs for this are shown in figure 4-9. The upper plot shows skip sizes of 20 frames and 30 frames for one-minute segments. Given that the peak width is only about 40 frames (see figure 4-6), we cannot push this offset much farther. Even at 30 frames the results are quite usable.

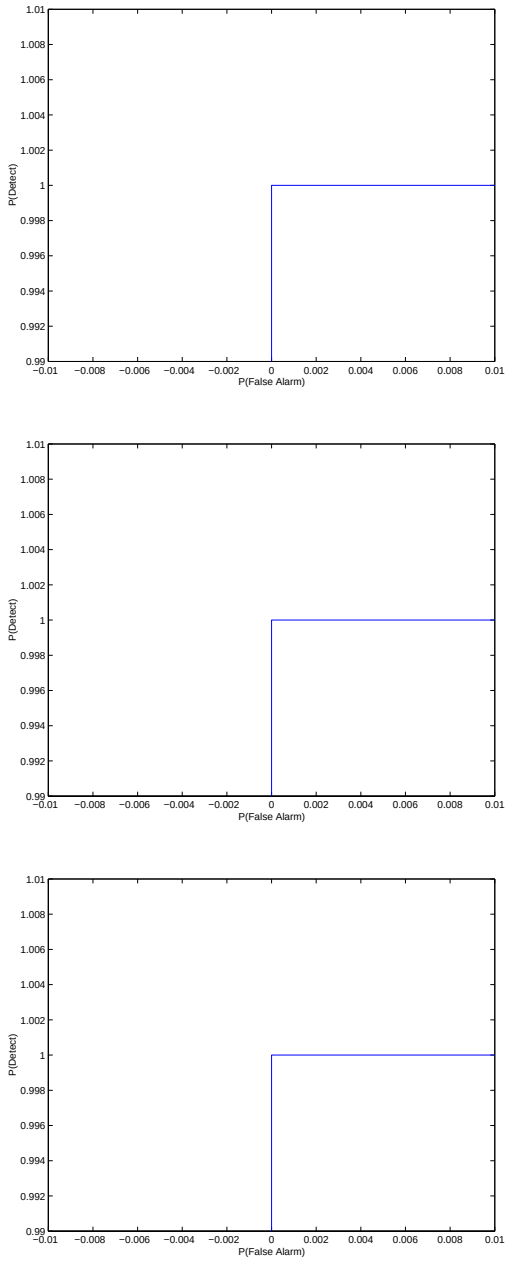

Figure 4-8: ROC curves for detecting conversations with different segment lengths: one minute (top) thirty seconds (center) and fifteen seconds (bottom). Note we have perfect recognition (100% detection with no false alarms) in all cases.

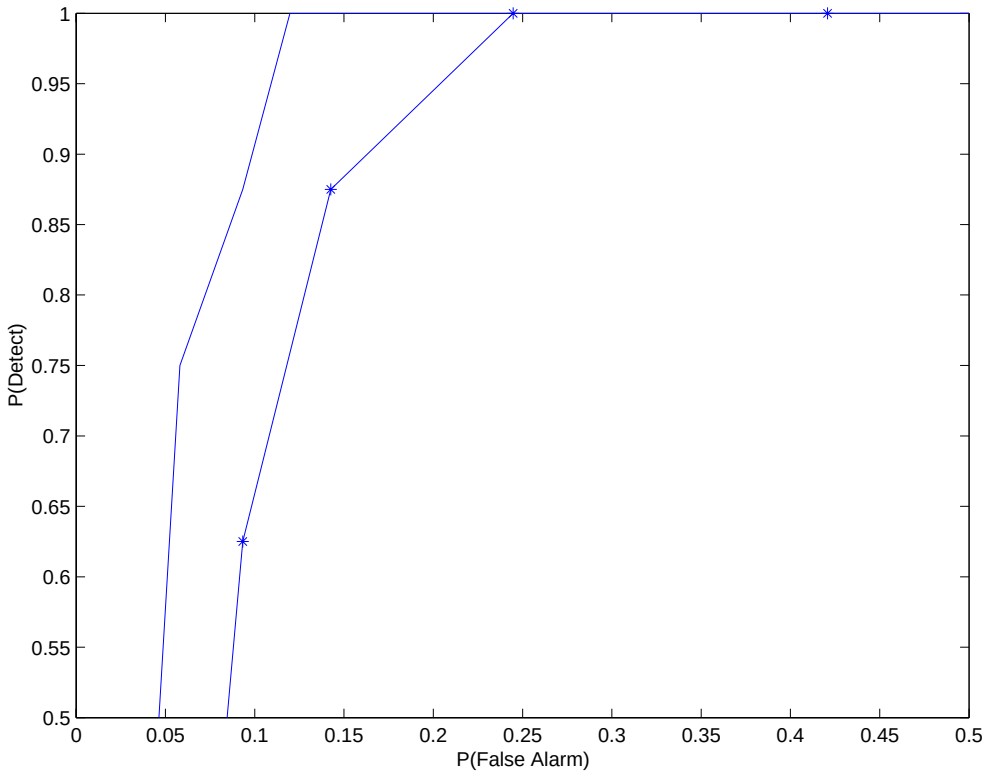

Figure 4-9: ROC curves for conversation detection at different skip sizes for a one-minute segment length, where a skip size of 20 means we only test the alignment every 20th frame. ROCs shown are for 20 frame (-) and 30 frame (*) skips.

4.3 Applications

There are a variety of possibilities for applying these interesting results. The first is for audio alignment, a common problem for situations in which we have unsynchronized microphones. We may have a number of audio streams recording a conversation but not know their precise alignment – this method could be very useful in this scenario.

The obvious application, though, is for finding conversations. From what we have shown, we can very reliably find conversational pairs over many false alarms, even when we are skipping along with an offset of twenty frames. One possibility for using this is to find the conversations among many audio streams from individuals wearing microphones, as in the sociometer data. In this case we have a relatively small number of streams, but many different possible time offsets, as the devices' clocks may be quite a bit off from one another.

A more important scenario is for security. Consider having microphones peppering a large airport, recording synchronously or perhaps close to it. If two people are coordinating an attack over cellphone, we would be able to detect the fact that they are having a conversation and then listen to both sides of the interchange. In this case, we have many possible streams to consider pairwise, but negligible time offset. If engineered correctly, such a system could help prevent attacks before they happen.

Chapter 5

Conversational Scenes

With all of our features in hand and the ability to segment speakers and find conversations, we are finally ready to consider conversational scenes. For the remainder of this study, we will be focusing exclusively on data from the LDC English callhome database. As we described in our introduction, this database consists of half-hour international phone conversations between two American-accented acquaintances, typically family members or close friends. As a result, it is an excellent repository for natural interactions and perfect for our purposes of conversational scene analysis.

We began our analysis by running our voicing, speech, energy, and pitch features on 29 half-hour files (a total of 29 hours of speech data considering both speakers). Upon examining the results, one of the first things we noticed was the surprising amount of overlap between the speakers, as shown in figure 5-1. One speaker starts speaking right on top of the other to respond, and first speaker does not pause for this "turn," in fact, she doesn't break stride at all. This tells us that any "transcript" of conversational data would be at the least misleading – conversation is an inherently multichannel phenomena. While there is a high degree of synchrony, as we saw in the last chapter, the speaking process is far from mutually exclusive.

Another important thing we noticed was that for a majority of the data, there was a clear sense of one speaker or the other holding the floor. This was typically not constant throughout the data, but changed at a relatively slow pace (on the order of minutes). This, then, becomes the driving force behind our conversational scenes – we

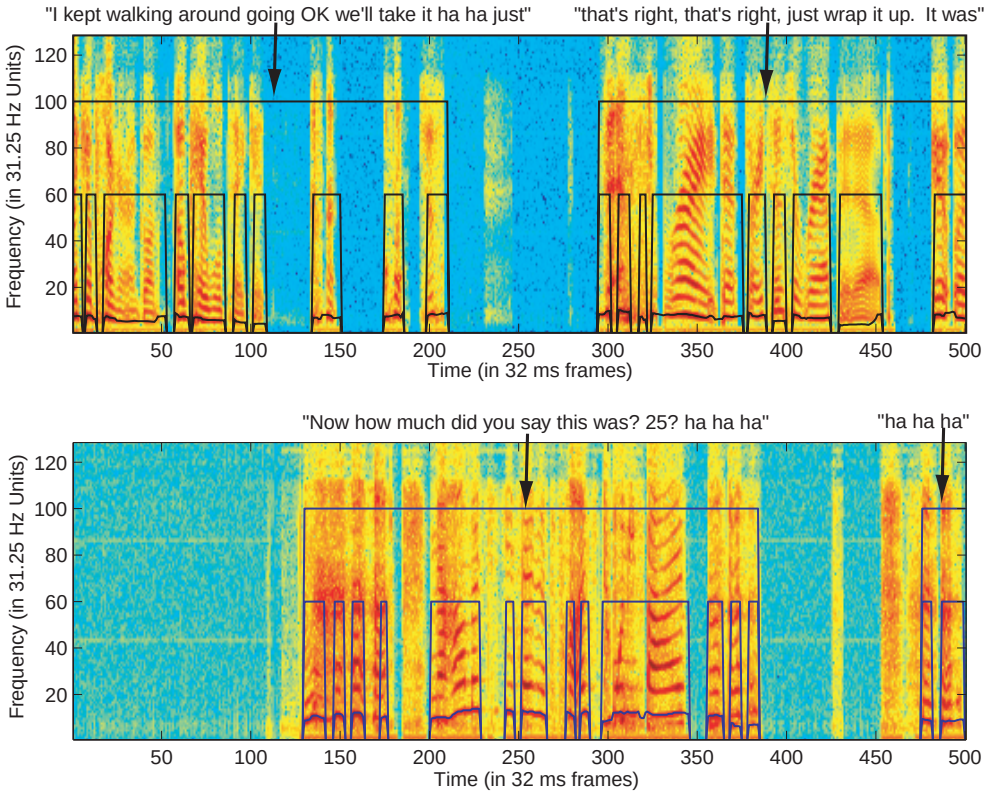

Figure 5-1: Voicing, speech, and pitch features for both speakers from an eight-second segment of a callhome conversation. Speaker one is shown on top; speaker two is below. The features are shown overlaid on the spectrograms for each speaker. The words corresponding to each speech segment are shown as well. Note that the speakers overlap with each other and do not completely wait for each other to take "turns."

will attempt to find the locations where one actor is holding the floor and thus taking on the dominant role in the scene, and also those locations in which there is no clear winner. The boundaries between these regions will segment the conversation into scenes. In the last chapter, we will use the distribution of these scenes to describe and recognize conversation types. Furthermore, we can characterize the scenes in detail by looking at how the features of one actor or another change from scene to scene.

5.1 Identifying Scenes: Roles and Boundaries

We started out by taking three half-hour conversations and marking out all the places where there was a definite dominance by one speaker or the other, labeling places with equal back-and-forth as a third state, and leaving ambiguous areas unlabeled. We thus had a labeling for both scene boundaries and the roles for each participant. Marking the boundaries was often somewhat arbitrary, in that the flow would often take a 10-15 seconds to shift from one speaker to the other – a fact that we should keep in mind when we evaluate the performance of our automatic methods.

The primary feature we found corresponding to these roles was the individual speaking times. We computed this as the fraction of time spent in voicing segments by each speaker over a 500-frame block (8 seconds):

$$f_v = \frac{\sum_{i=1}^{500} v_i^1}{500}. \tag{5.1}$$

We show this feature for both speakers for conversation EN 4807 in figure 5-2. Though the signal is noisy, the dominance of one speaker versus the other is quite clear. Another interesting feature of this data is the strong degree of symmetry between the speakers: when one speaks more, the other speaks a little less, and vice versa. It is not necessary that the data come out this way, but it is another marker of the inherent dynamics of human interactions: one speaker tends to yield the floor as the other takes over.

The HMM seemed like an obvious candidate for this problem – we had three possible states (speaker 1 dominates, speaker 2 dominates, and neither dominates), a noisy feature exhibiting some stationarity per state, and a need to determine where the state changes occurred. We thus used a three-state HMM as our basic model. Because of the observed symmetry between the speaking fractions, we used only a single Gaussian observation, the difference between the voicing fractions, i.e.,

$$y_t = f_{v,1} - f_{v,2} \tag{5.2}$$

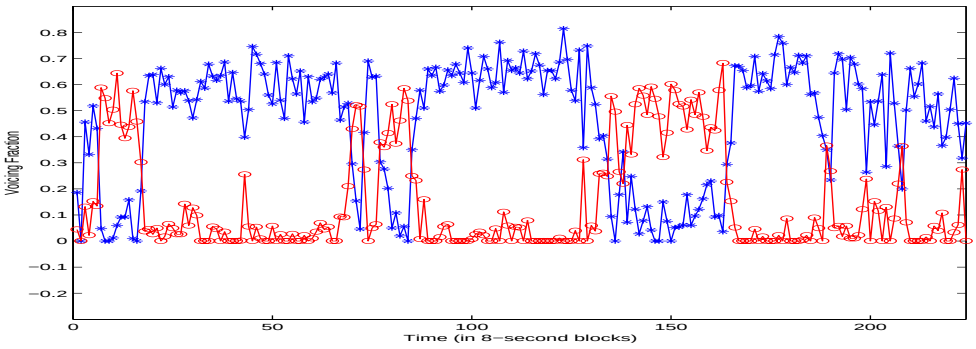

Figure 5-2: Plots of the voicing fraction for each speaker in 500 frame (8 second) blocks for conversation EN 4666. The entire sequence is half an hour long. The dark line (*) is the voicing fraction for speaker one, the lighter line (o) is the fraction for speaker two. Note how the dominant speaker shifts over time and how at some points there is no dominant speaker.

We trained this model on two of the three labeled conversations and evaluated its performance on the third. The results of the evaluation are shown in table 5.1.

Table 5.1: Scene Labeling Performance for the HMM.

% Correctly Labeled	Mean Distance from Boundary
96.23	0.9 frames (7.2 seconds)

To see what these results really mean, it helps to look at a few segmented conversations. We start with figure 5-3, the test sequence. Here we see a series of fairly long scenes, often dominated by speaker two. In the end, we see a rapid exchange of scenes and some ambiguous regions in which no speaker is dominating. The next is figure 5-4, which is quite different from the first. Here, the two speakers are shooting back and forth for almost the entire conversation, excepting two short segments during which speaker two has the upper hand. We will have much more to say about characterizing these different types of conversations in the next chapter, but for now we just want to pay attention to the scene boundaries. Notice how in both examples, the model does a good job of determining when a given speaker has begun dominating.

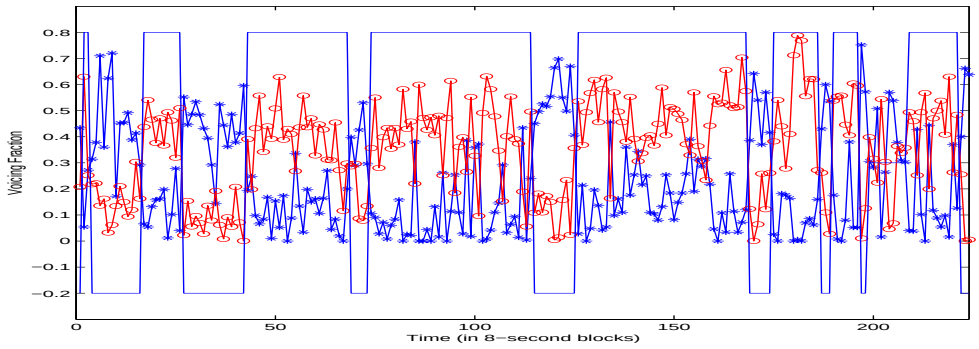

Figure 5-3: Results of scene segmentation for conversation EN 4807. The solid line weaving through the speaker fractions represents which speaker is dominating. When the line is high, speaker two is holding the floor, when it is low, it is held by speaker one. When it is at the center, neither speaker holds the floor.

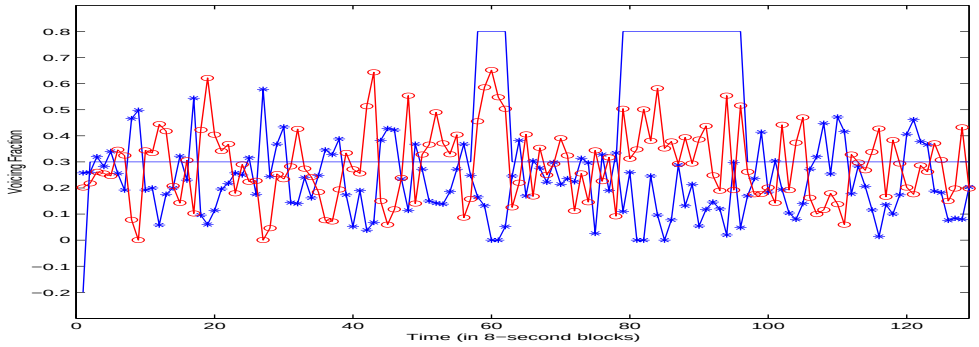

Figure 5-4: Results of scene segmentation for conversation EN 4838.

5.2 Predicting Scene Changes

Since we are able to find the scene boundaries quite accurately *a posteriori*, this led us to the question of how well we might be able to *predict* the scene changes. Given the nature of the changes, it seems highly unlikely that we will be able to predict them far in advance, but there seems to be hope in predicting them just as they are about to happen, as the dominating speaker begins to trail off.

To clarify our terms, we are only interested in scene changes that involve one speaker giving up the floor to the other, and will not consider changes that go from either speaker to a neutral position. We will see in the results, though, that our detector tends to find the latter in spite of its limited training. Furthermore, we are

detecting whether or not a scene change is going to begin at the current time. Finally, we are only interested in the performance of the detector for a given speaker giving up the floor while that speaker is actually holding the floor – we will not penalize it for false alarms that occur while the other speaker is dominating, as these are easy to cull in a real system.

Because of the limited amount of training data and our desire for generality, we will implement this detector with a single, diagonal Gaussian using three features: the change in f_v from the previous frame for the speaker of interest, the same for the other speaker, and the fraction of the total speaking time represented by f_v; i.e.,

$$\frac{f_{v,i}}{f_{v,i} + f_{v,j}}. \tag{5.3}$$

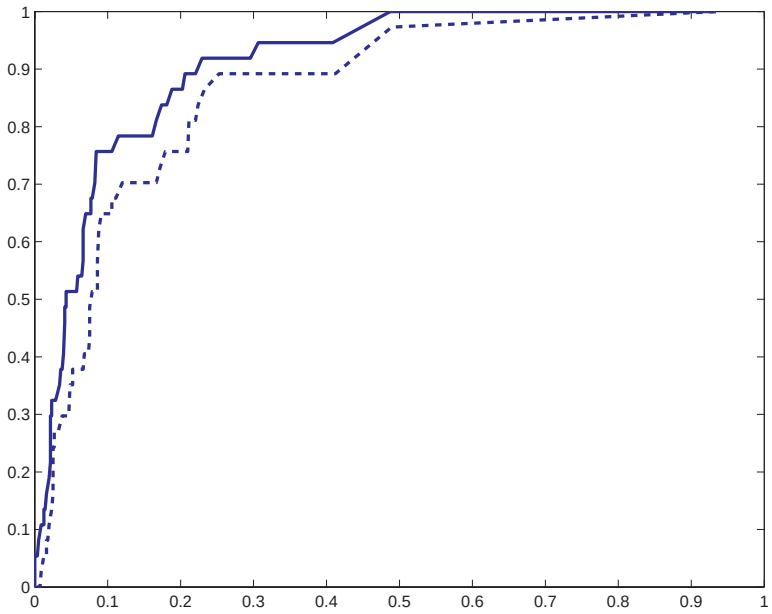

Figure 5-5: The ROC curves for the prediction of scene changes for a slack of 0 ($\pm$8 seconds, dashed line) and 1 ($\pm$16 seconds, solid line) 1000-frame blocks.

We tried a variety of other features including speaking rate, pitch change, energy change, and the like, but none were as effective as the three above. We also found that the detector worked disproportionately better on 1000-frame blocks as opposed to 500-frame blocks, and thus we show only the results for the former. In figure 5-5 we see the ROC curves for detecting a speaker change at the correct frame (a 16 second

span) and for getting it within one frame (a 48 second span). While the results are far from perfect, they are rather interesting. The task we are trying to do here is quite difficult – even while labeling, we often had to go back and forth through the data to find the best scene change time. It is not surprising, then, that predictor yields many false alarms. The nature of these false alarms is quite interesting as well. Figure 5-6 shows a series of scenes and the log likelihood of our detector plotted underneath it. The false alarms at 70 and 170 both show speaker 1 *almost* giving up the turn, but then beginning to dominate again. Note also at 72 though the classification is technically incorrect in that speaker 1 did not give up the floor to speaker 2, he did indeed give up the floor to move to a neutral interaction.

While these points are incorrect in terms of our definition, they could still be quite useful for application purposes. Imagine, for instance, that we are developing any sort of interface that must interrupt the user – a cellphone, an email program, perhaps a robot butler. If such a system could detect these lulls in conversations, it could use these times to interrupt instead of barging in during a story from one of the speakers. Such socially aware systems could make for a far more pleasant presence of interactive technology in our environment.

5.3 Scene-Based Features

Now that we can accurately identify scene boundaries and the roles of the actors in the scene, it becomes interesting to look at the individual features in more detail. While we will not attempt to make an ontology of different speaker states in this work, we will demonstrate with a simple example the sort of browsing and characterization power our scene-based analysis gives us.

Conversation EN 4807 is different from most of the other callhome files in that we have one speaker, a daughter (speaker 2), speaking to both of her parents (speaker 1). While at first this tempted us to discard the data, we realized that it instead made for a very interesting scenario. We noticed when listening to the data that the daughter sounded quite bored when talking to her father, even while she was holding the floor.

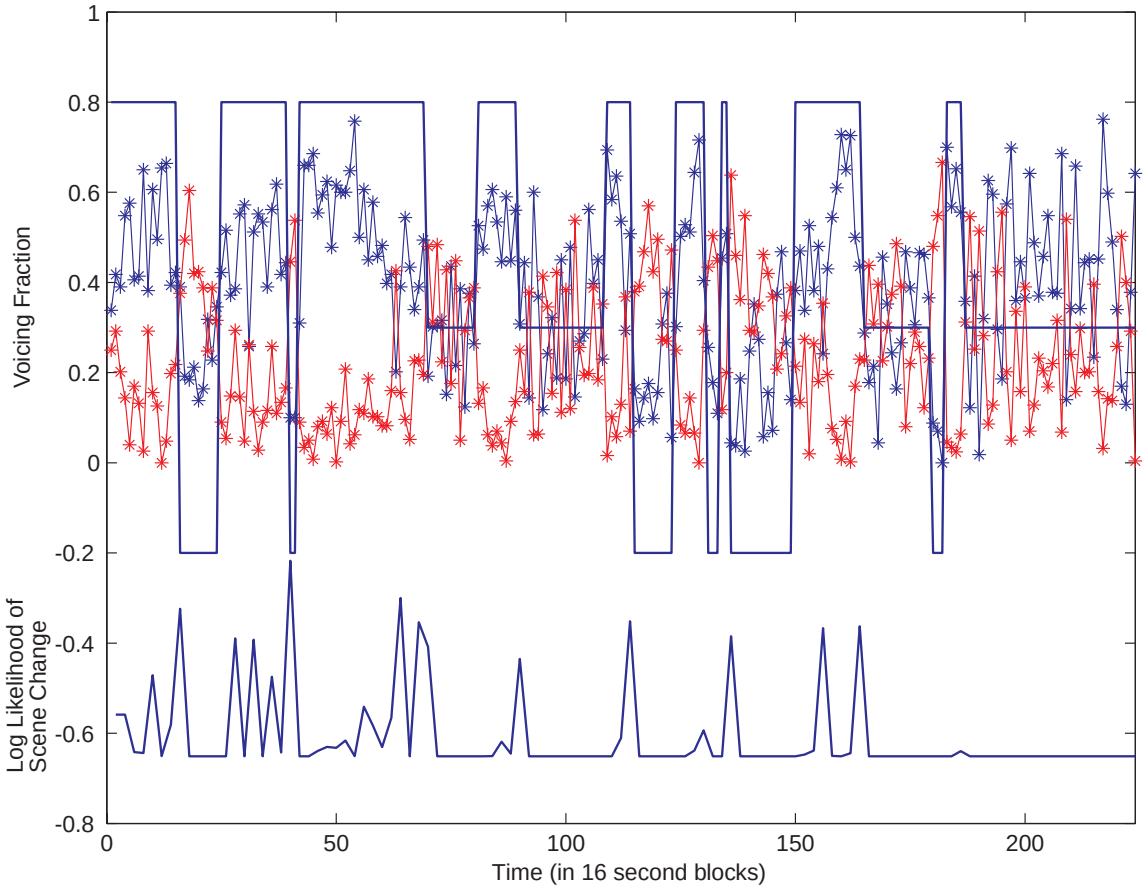

Figure 5-6: The log likelihood (shifted down for clarity) of speaker two giving up the floor to speaker one given the current features, plotted along with the actual scene boundaries. Notice that there are several false alarms, but also that these typically correspond to places where speaker two almost gave up the floor.

On the other hand, she sounded very lively indeed when talking to her mother. In table 5.3, we show the daughter's pitch, energy, and speech gap features for two scenes where she is holding the floor. We see that the energy mean and variance are both much higher with her mother, and that the speech gap length is much smaller. The pitch and pitch variance also seem to have risen somewhat, though perhaps not that significantly (our pitch resolution is 31.25Hz). In any case, these changes correlate well with our impressions of her greater engagement with her mother.

When we examine another pair of scenes, in which it is the parents who are dominating, (table 5.3), the results are even more dramatic. When reacting to her mother, the energy/energy range and pitch/pitch range for the daughter are signifi-

	Father	Mother
Pitch (Hz)	444±90	472±98
Energy	.36±30	.78±.67
Speech Gap(sec)	1.03	.66

Table 5.2: Speaker two's variations across different conversational partners in two scenes where she is holding the floor.

cantly higher, and the gap between speaking chunks has dropped by almost half.

	Father	Mother
Pitch (Hz)	372±51	555±99
Energy	.18±10	.85±.78
Speech Gap(sec)	3.13	1.85

Table 5.3: Speaker two's variations across different conversational partners in two scenes where the partner is holding the floor.

While this result is preliminary, we believe it paves the way for a very interesting avenue of research – examining how a given speaker's characteristics vary across scenes and conversational partners. This could be an invaluable tool for users to be able to assess their interactions with others, perhaps seeing and repairing patterns of interactions that they didn't expect. Though it is often obvious to observers how we treat others differently, it is seldom so obvious to us. Perhaps our analysis tools can help bridge some of this gap.

Another use for this sort of analysis is for indexing interactions based on our reactions. For instance, I may be interested in perusing all of my conversations with Joe, but only in those scenes in which he was holdig the floor and I was more interested than usual. Or perhaps I would like to find the conversations where I was dominating and *he* was more interested than usual. In either case, this could prove to be a powerful means of indexing through a vast store of past interactions.

Chapter 6

Conversation Types

Now that we can effectively segment conversations into scenes, we can think about how to characterize the overall conversation type. Clearly the roles taken by the individuals play an important part, i.e., whether one speaker holds the floor more than the other. Also important are the lengths of the scenes that make up the conversation. In this chapter, we will show how we can quantify these two notions and use them as a space for categorizing and browsing conversations.

6.1 Features

The first feature we are interested in characterizing is the level of dominance in the conversation. We begin our investigation of this feature by looking at the histogram of how long each actor (or the neutral state) holds the floor throughout an entire conversation. We will call this plot the *dominance histogram* for a conversation, and we show two examples in figure 6-1. In these histograms, bin one reflects the amount of time speaker one holds the floor, bin two is the neutral time, and bin three is time held by speaker two.

We can see how the histogram provides a nice summary of the level of dominance in the interaction, and we thus condense this into the single quantity of *dominance level* by the following:

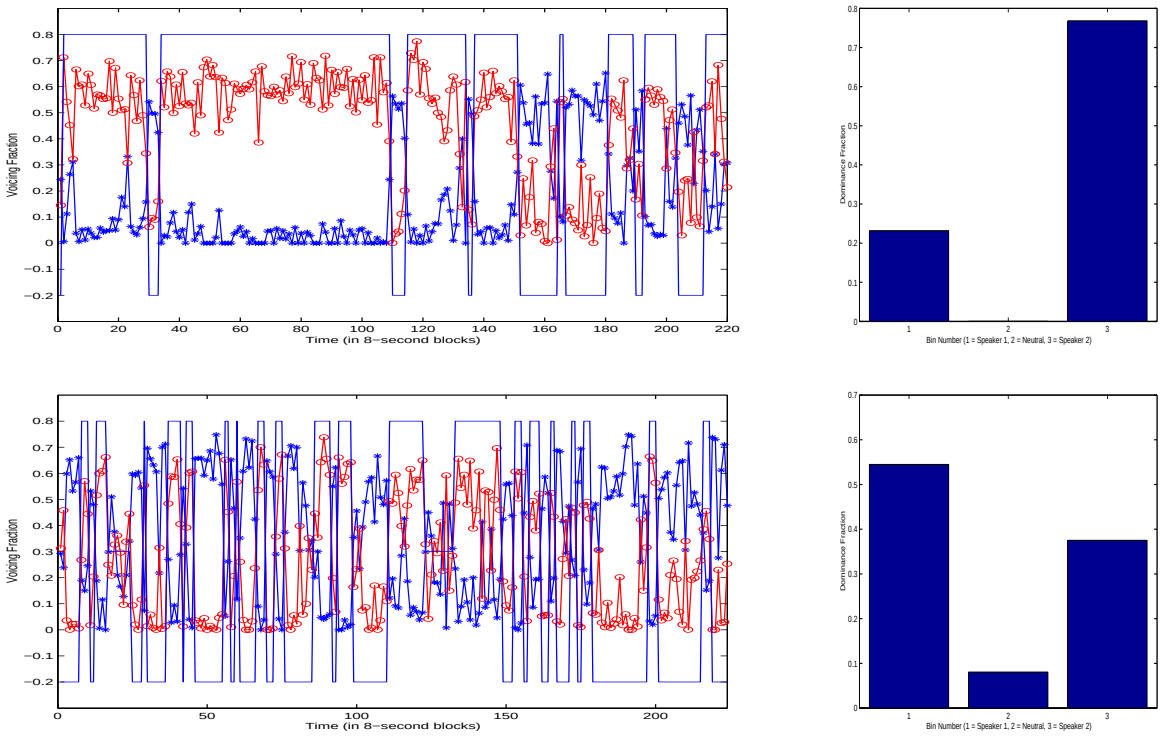

Figure 6-1: Results of scene segmentation and dominance histogram for two conversations, EN 4705 and EN 4721. Bin one shows dominance by speaker one, bin two shows a neutral state, and bin three shows dominance by speaker two.

$$l_d = |P(d=1) - P(d=2)|\,, \tag{6.1}$$

where $P(d=i)$ refers to the probability of dominance by speaker i as computed from the histogram. This feature summarizes the amount of dominance in the conversation, regardless of whether it was speaker one or two who was doing the dominating.

The second measure we are interested in is how much interaction there is between the two speakers. For instance, it is possible that speaker one always holds the floor and speaker two never does, but the scenes are quite short. This situation could occur if the dominance keeps flipping from speaker one to the neutral state. Some measure of the average scene length is clearly what we are after, but we need to be careful about how we compute it. If there are two 20 minute scenes in a conversation followed by two two-minute scenes, the average scene length is 10 minutes, which does not really reflect the measure we want. To further confound the issue, we really

should consider the neutral state as being composed of many tiny scenes of length one. How then to compute a mean scene length that is not always one or close to it?

There is a simple solution to this issue. This problem maps exactly to the standard probabilistic notion of random incidence [10]. Instead of computing a mean over the distribution of scene lengths, let us consider the distribution of scene lengths we would see if we chose a random point in the conversation and then considered the length of the containing scene. We can write the distribution of scene lengths found in this manner, $p_i(l)$, as:

$$p_i(l) = \frac{lp(l)}{\sum_l lp(l)} = \frac{lp(l)}{E[l]}, \tag{6.2}$$

where $p(l)$ is the original distribution of scene lengths. If we take the expectation of this quantity, we find

$$E_i[l] = \sum_l l\frac{lp(l)}{E[l]} = \frac{1}{E[l]}\sum_l l^2 = \frac{E[l^2]}{E[l]}. \tag{6.3}$$

Empirically, we found this mean incident scene length to much better reflect the length of the average scene, and have thus used this as the second feature.

6.2 Describing Conversation Types

In figure 6-2, we show the scatterplot of 29 conversations along the dimensions of dominance level and mean incident scene length. There are two main features of interest: the first is the broad stripe of conversations from the lower left corner (low dominance, short scene length) to the upper right (high dominance, long scene length). This represents a continuum from "chatty" conversations, where both parties are going back and forth at an equal level with short exchanges, to "lecturing" situations, where one person is always holding the floor and the scenes are quite long.

The other cluster to note is the group of two conversations in the upper left corner (low dominance, short scene lengths). These represent a "storytrading" situation, where each speaker holds the floor for quite a while, but both get nearly equal

amounts of speaking time. Since these are international calls between friends and family members, we expected to see more conversations like this, where each side would tell their various stories uninterrupted. For these conversations, at least, it seems that such updating usually occurs with shorter scenes.

Furthermore, it is interesting that we do not see any conversations in the lower right corner, i.e., with high dominance but short scenes. This would be entirely possible – it would mean a conversation switching quickly between a given speaker dominating and the neutral state. Similarly, we see no conversations with medium dominance and long scene lengths: again possible, but not observed. This may be due to the nature of the callhome database: perhaps in face-to-face conversations, we would see different gaps in the feature space. This may, in fact, be a useful way to characterize a given communication channel. For example, long delays or frequent dropouts could discourage rapid exchanges between the speakers, thus resulting in a different distribution in this feature space. Perhaps we could then evaluate the quality of a channel or the nature of a relationship by examining this distribution.

To see if the data supported our impressions, we automatically clustered the data by fitting a mixture of three full-covariance Gaussians with the EM algorithm. The results are shown in figure 6-3. Indeed, the continuum we noticed has been modeled by two of the clusters, one on the "chatty" end and one towards the "lecturing" end. The two storytrading conversations make up the third cluster. Though the dataset is quite small, this gives us some confirmation of our preliminary observations.

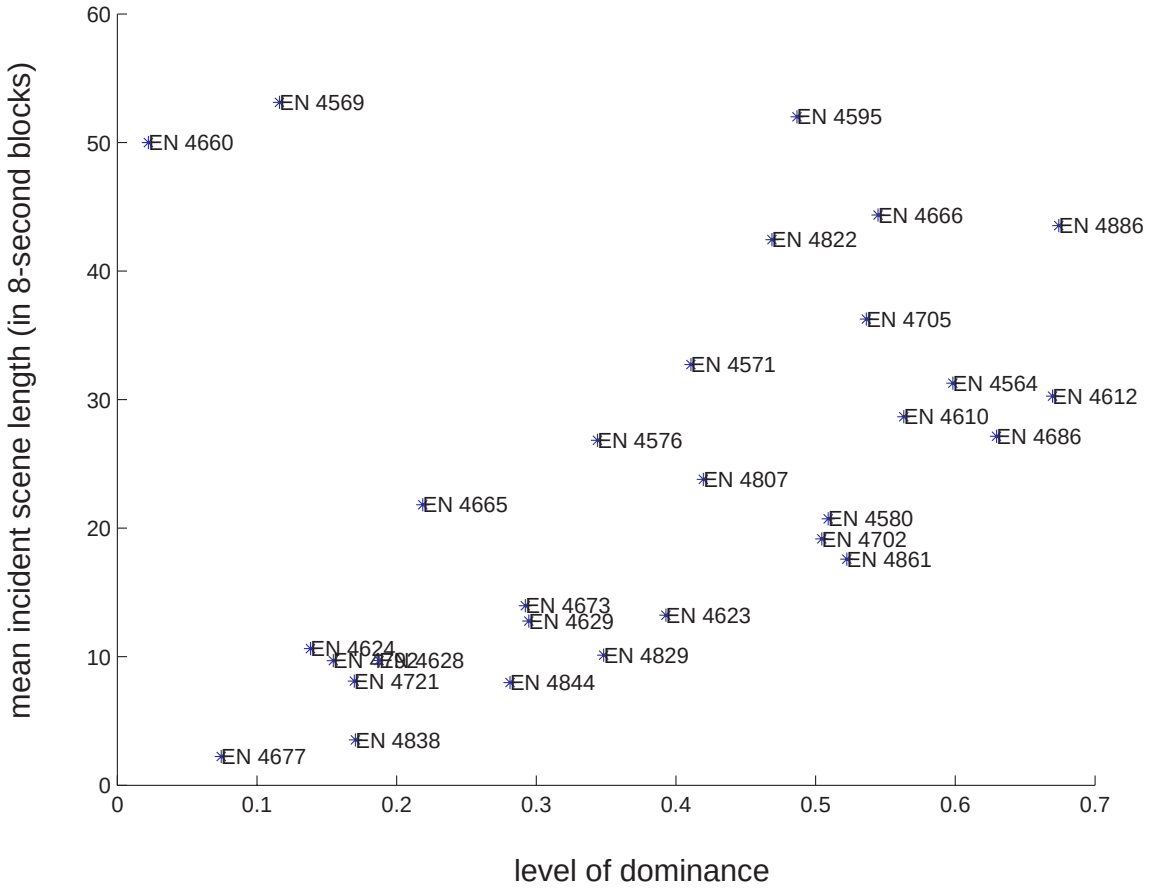

Figure 6-2: A scatterplot of all 29 conversations. Note the continuum between "chatty" conversations in the lower-left corner (low dominance, short scenes) and "lectures" in the upper-right (high dominance, long scenes). Also notice the two "storytrading" conversations in the upper-left corner (low dominance, long scenes).

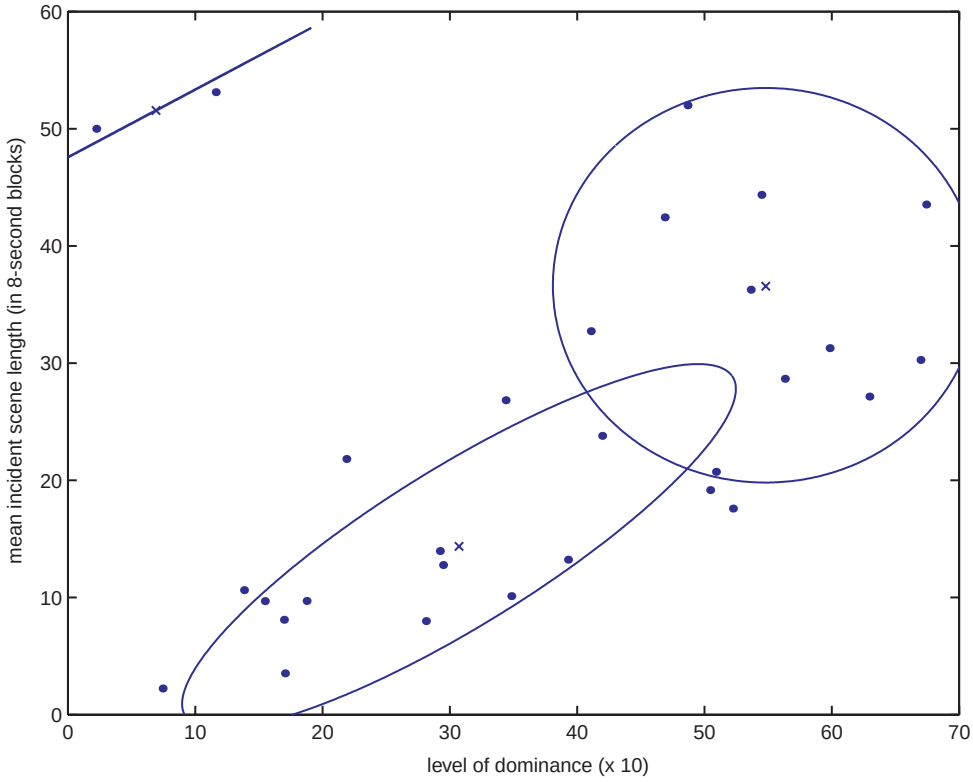

Figure 6-3: The result of clustering the conversations using the EM algorithm with a mixture of three full-covariance Gaussians. The ellipses correspond to isoprobability contours of the fitted Gaussians. Note that two of the clusters model the "chatty"-"lecture" continuum, while the third covers the storytrading conversations.

6.3 Browsing Conversations By Type

We now examine the potential of these qualitative categories as a means to browse conversational data. Looking again at the scatterplot of figure 6-2, let us seek out a chatty conversation with a minimum of dominance and short scenes. We choose conversation "EN 4677." The scene segmentation for this conversation is shown in figure 6-4. Indeed, this conversation is almost entirely spent in the neutral state, and shows continuous, rapid exchanges between the speakers.

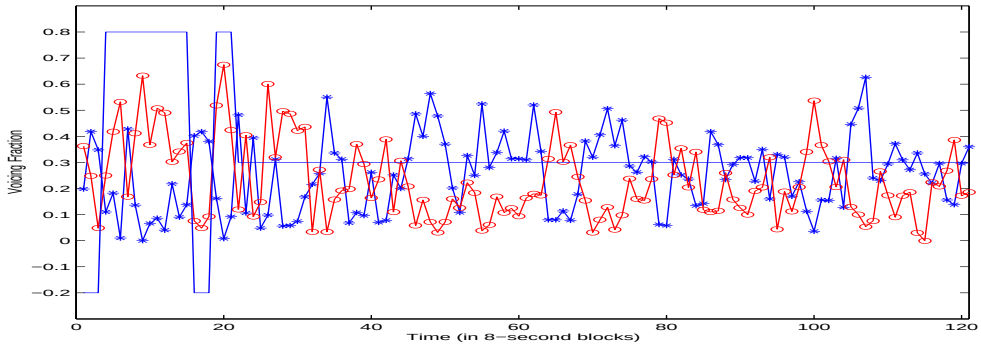

Figure 6-4: Results of scene segmentation for conversation EN 4677, a low dominance, short scene length conversation, i.e., on the "chatty" end of the chat-lecturing continuum. Most of the time is spent in a neutral state, with neither speaker dominating.

Now let us seek out a "storytrading" interaction, where both parties are spending a long time holding the floor. We choose conversation "EN 4569," and show its scene segmentation in figure 6-5. Here we see that speaker two dominates during the first half of the conversation with short interrupts from speaker one, after which speaker one takes over and tells his own story for the rest of the conversation.

Finally, we seek out a "lecture" interaction, where one speaker is dominating over the course of fairly long scenes: conversation "EN 4666." We show its segmentation in figure 6-6. In the figure, we see that speaker one dominates for almost the entire discussion, with only a few short interludes from speaker two.

While looking through conversations between strangers in this way may not seem very useful, imagine this were a set of our own conversations with a number of our colleagues – or perhaps a series of conversations between suspected terrorists. The

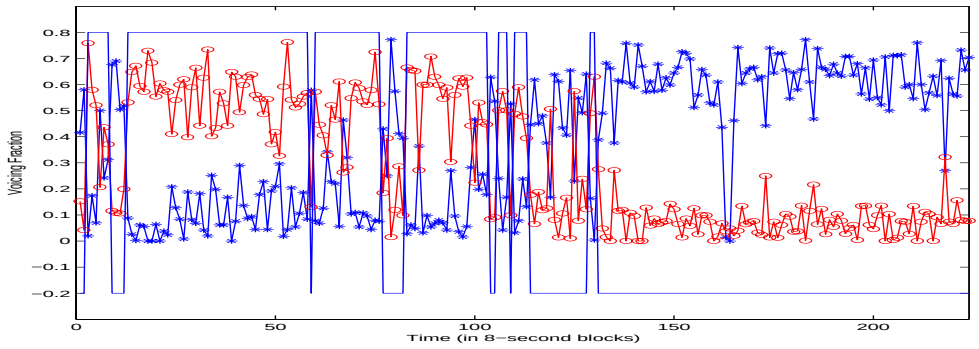

Figure 6-5: Results of scene segmentation for conversation EN 4569, a low-dominance, long scene length conversation, which we describe as "storytrading." Both speakers take significant amounts of time holding the floor, but in equal proportions.

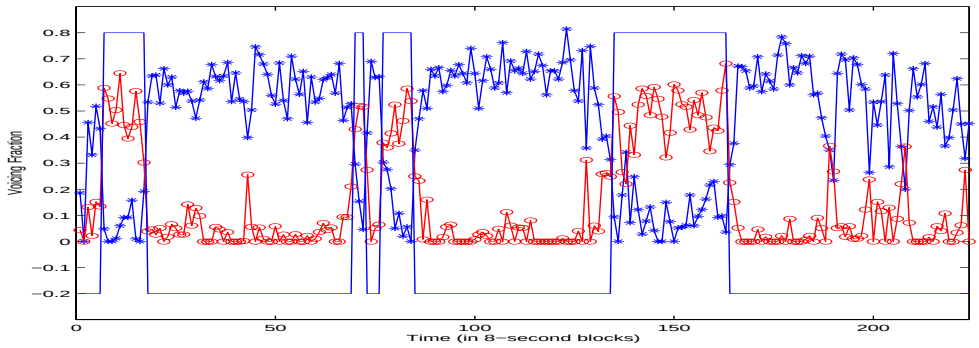

Figure 6-6: Results of scene segmentation for conversation EN 4666, with high dominance and long scene lengths, i.e., on the "lecturing" end of the chat-lecturing continuum. Speaker one holds the floor almost exclusively in this conversation with a few short interludes from speaker two.

analysis we have shown would quickly let us pinpoint the discussion in which Carol was explaining her thesis idea in detail, or instead when one suspect was rapidly exchanging information with another. We could then look even more closely at the scenes as we showed in the end of the last chapter, examining the different ways in which the actors acted and reacted. Overall, we believe these multiscale features of conversations could make for very powerful search and browsing tools.

Chapter 7

Conclusions and Future Work

We have covered a great deal of material and experiments in this work, and we would like to reemphasize here the principal contributions. First is our energy-independent voicing and speech segmentation algorithm based on the linked-HMM architecture. As we have shown, this method significantly outperforms the related work in terms of voicing segmentation and gives us the additional information of speech segmentation. The algorithm is extremely robust to noise, distance from microphone, and environment without the need for any retuning, as we have shown in a series of experiments. The key to this method's success was exploiting the dynamics of the speech production process, by modeling the different switching behaviors between voiced and unvoiced frames in speech and non-speech regions. The results of this segmentation then allowed to us to compute a number of other features. Though we introduced new methods for probabilistic pitch tracking, speaking rate estimation, and normalized energy estimation, all of these depended heavily on the preprocessing by the voice and speech segmentation module.

We then showed a number of new results for speaker segmentation with energy and phase, now using the results of our voicing segmentation to integrate or accumulate noisy features over time. In both cases, we were able to show marked improvements over use of the raw signal. This came about from being able to use the dynamics of the voiced-unvoiced segments as a guide for integrating features, instead of resorting to artificial dynamic constraints that could skip over abrupt speaker changes.

Our next major contribution was in terms of finding conversations. We showed how we could very robustly detect when two audio streams were involved in a conversation against thousands of possible false alarms, using only two minute segments of conversation. Once again, the voicing segmentation was the key feature for this process, along with the dynamics of conversational exchanges as represented by our alignment feature. As a result, we were able to perform this task with very high accuracy even under significant noise conditions.

Our last set of contributions were in the domain of conversations. We first showed how we could reliably break a conversation into scenes, i.e., regions during which one speaker was dominating or there was a neutral manner of interaction. We did this by regularizing the noisy voicing features using the dynamics of the scene changes, as represented by a simple HMM. We showed a strong accuracy for this both in terms of the dominance classification and in the segmentation boundaries. We also showed that we could predict when the conversational scene was about to change with reasonable accuracy though with a good number of false alarms. However, these false alarms were still interesting in that they often signified "lulls" in the interaction. At the least, these are good potential locations for an agent to interrupt a conversation. Last, we showed preliminary results on investigating the variation in speakers' characteristics across conversational partners and scenes. While this requires further study, it seems to hold great promise as a means for computationally investigating the differences in our relationships with others.

Finally, we developed two features with which we could summarize the overall nature of a conversation – dominance level and mean incident scene length. We showed how these features could be used to categorize and browse conversations. This has the potential to be a powerful mechanism for search and browsing of large spans of multimedia data.

Given where we have come to and what still lies ahead, the possibilities for future work are many. Among the first we wish to pursue is the further investigation of people's variations in style across different conversational partners. The nature of the callhome data made this difficult, as most people were speaking with only one

partner. We are already in the process of developing mechanisms to collect data of this kind. Next, we would like to study the possibilities of using our analysis as a browsing tool. Already in the course of this work we have found our features very helpful when searching for various examples for this document – we are certain that they could be useful to others as well. Furthermore, as our features are at several timescales, from 0.032-second frames to half-hour conversation features, they could very powerful for "zooming" in and out of multimedia content. Another interest is the integration of visual information with the auditory modalities we have investigated here. Certain types of phenomena such as head nods and gaze direction could be very useful in giving us additional information about reactions and engagement. Finally, we would like to integrate speech recognition information into our work. While we have said repeatedly that conversations are not made up of words alone, the words do still play an important role, and we believe that the combination of our scene analysis and some topic spotting based on noisy recognition results could make for a very powerful search/browsing tool for multimedia data.

While this study has not covered every possible aspect of conversational scene analysis, we feel we have a covered a significant amount of initial territory, and that we have opened the door to an interesting new area of study. The feature processing techniques we have introduced are powerful and robust to real-world conditions – they are certainly not restricted to the world of close talking microphones. Furthermore, we think our notion of conversational scenes is a natural and powerful one, and hopefully one that can be used by many other scientists seeking to do this sort of automatic analysis in the years to come. The scene characterizations and conversation types we have developed and recognized, while preliminary, point the way for a well-defined analysis of conversations and conversational scenes. With these techniques and results in hand, we can continue to develop quantitative means for characterizing the subtleties of human interactions.

Bibliography

[1] Sassan Ahmadi and Andreas S. Spanias. Cepstrum-based pitch detection using a new statistical v/uv classification algorithm (correspondence). *IEEE Transactions on Speech and Audio Processing*, 7(3):333–338, 1999.

[2] Sumit Basu, Brian Clarkson, and Alex Pentland. Smart headphones: Enhancing auditory awareness through robust speech detection and source localization. In *ICASSP 2001*, Salt Lake City, Utah, 2001. IEEE. Submitted to ICASSP'01.

[3] Sumit Basu, Irfan Essa, and Alex Pentland. Motion regularization for model-based head tracking. In *Proceedings of 13th Int'l. Conf. on Pattern Recognition*, pages 611–616. August, 1996.

[4] Sumit Basu, Nuria Oliver, and Alex Pentland. 3d lip shapes from video: A combined physical-statistical model. *Speech Communication*, 26:131–148, 1998.

[5] Albert Bregman. *Auditory Scene Analysis*. MIT Press, 1990.

[6] Brian Clarkson and Alex Pentland. Unsupervised clustering of ambulatory audio and video. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'99)*. 1999.

[7] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, 1991.

[8] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

[9] George Doddington. Speaker recognition – identifying people by their voices. *Proceedings of the IEEE*, 73(11), November, 1985.

[10] Alvin W. Drake. *Fundamentals of Applied Probability Theory.* McGraw-Hill, New York, 1967.

[11] James Droppo and Alex Acero. Maximum a posteriori pitch tracking. In *Int'l Conf. on Speech and Language Processing (ICSLP'98)*, 1998.

[12] Pfau T; Ruske G. Estimating the speaking rate by vowel detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98*, pages 945–8, 1998.

[13] Peter Heeman, Donna Byron, and James Allen. Identifying discourse markers in spoken dialog. In *the AAAI Spring Symposium on Applying Machine Learning and Discourse Processing*, Stanford, 1998.

[14] Julia Hirschberg and Christine Nakatani. A prosodic analysis of discourse segments in direction-giving monologues. In *Associal for Computational Linguistics*, 1996.

[15] Julia Hirschberg and Christine Nakatani. Acoustic indicators of topic segmentation. In *International Conference on Speech and Language Processing (ICSLP'98)*, 1998.

[16] Liang-sheng Huang and Chung-ho Yang. A novel approach to robust speech endpoint detection in car environments. In *Proceedings of ICASSP'00*, pages 1751–1754. IEEE Signal Processing Society, 2000.

[17] Michael Jordan and Chris Bishop. *An Introduction to Graphical Models.* 2002 (in press).

[18] Jean-Claude Junqua, Brian Mak, and Ben Reaves. A robust algorithm for word boundary detection in the presence of noise. *IEEE Transactions on Speech and Audio Processing*, 2(3):406–412, 1994.

[19] F. Khalil, J.P. Jullien, and A. Gilloire. Microphone array for sound pickup in teleconference systems. *Journal of the Audio Engineering Society*, 42(9):691–699, 1994.

[20] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[21] Stephen C. Levinson. Conversational structure. In *Pragmatics*, pages 284–369. Cambridge University Press, Cambridge, 1983.

[22] John Makhoul, Francis Kubala, Timothy Leek, Daben Liu, Long Nguyen, Richard Schwartz, and Amit Srivastava. Speech and language technologies for audio indexing and retrieval. *Proceedings of the IEEE*, 88(8):1338–1353, 2000.

[23] Nuria Oliver, Eric Horvitz, and Ashutosh Garg. Hierarchical representations for learning and inferring office activity from multimodal information. In *IEEE Workshop on Cues in Communication, in conjunction with CVPR'01*. IEEE, 2001.

[24] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[25] Douglas O'Shaugnessy. *Speech Communication: Human and Machine*. Addison-Wesley, 1987. The Book for speech production basics.

[26] Vladimir Pavlovic, James Rehg, and John MacCormick. Learning switching linear models of human motion. In *Neural Information Processing Systems (NIPS)*, 2000.

[27] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989. cwren; HMM theory, implementation optimizations, and speech applications.

[28] Deb Roy. Learning words from sights and sounds: A computational model ph.d. thesis. Technical report, MIT Department of Media Arts and Sciences, 1999.

[29] L. K. Saul and M. I. Jordan. Boltzmann chains and hidden markov models. In *Neural Information Processing Systems 7 (NIPS 7)*, 1995.

[30] Klaus R. Scherer, Judy Kovumaki, and Robert Rosenthal. Minimal cues in the vocal communication of affect. *Journal of Psycholinguistic Research*, 1:269–285, 1972.

[31] Bruce G. Secrest and George R. Doddington. An integrated pitch tracking algorithm for speech systems. In *Proceedings of ICASSP'83*, pages 1352–1355. IEEE Signal Processing Society, 1983.

[32] A. Waibel, M. Bett, M. Finke, and R. Stiefelhagen. Meeting browser: Tracking and summarizing meetings. In *DARPA Broadcast News Workshop*, 1998.

[33] Chao Wang and Stephanie Seneff. Robust pitch tracking for prosodic modeling in telephone speech. In *Int'l Conf. on Spoken Language Processing (ICSLP'00)*, 2000.

[34] Gin-Der Wu and Chin-Teng Lin. Word boundary detection with mel-scale frequency bank in noisy environment. *IEEE Transactions on Speech and Audio Processing*, 8(5):541–553, 2000.