Automatic Utterance Segmentation in Spontaneous Speech

by

Norimasa Yoshida

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

Master of Engineering in Computer Science and Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2002

Author . . . .
Department of Electrical Engineering and Computer Science
September 22, 2002

Certified by . . . .
Deb Roy
Assistant Professor of Media Arts and Sciences
Thesis Supervisor

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

Automatic Utterance Segmentation in Spontaneous Speech

by

Norimasa Yoshida

Submitted to the Department of Electrical Engineering and Computer Science
on September 22, 2002, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Electrical Engineering

Abstract

As applications incorporating speech recognition technology become widely used, it is desireable to have such systems interact naturally with its users. For such natural interaction to occur, recognition systems must be able to accurately detect when a speaker has finished speaking. This research presents an analysis combining lower and higher level cues to perform the utterance endpointing task. The analysis involves obtaining the optimal parameters for the signal level utterance segmenter, a component of the speech recognition system in the Cognitive Machines Group, and exploring the incorporation of pause duration and grammar information to the utterance segmentation task. As a result, we obtain an optimal set of parameters for the lower level utterance segmenter, and show that part-of-speech based N-gram language modeling of the spoken words in conjunction with pause duration can provide effective signals for utterance endpointing.

Thesis Supervisor: Deb Roy
Title: Assistant Professor of Media Arts and Sciences

Acknowledgments

These pages are dedicated to the following people, without whom I would not have been able to complete this work.

First and foremost, I would like to thank Deb Roy, my thesis advisor, for his support, wisdom and guidance throughout the entire project. I would like to thank the Cognitive Machines Group, particularly Niloy Mukherjee for his friendship and his valuable suggestions, and Peter Gorniak for his Unix expertise. I have grown tremendously during my time with the Cognitive Machines Group, and I have truly enjoyed working with an exceptional group of people.

I would also like to thank my academic advisor Peter Szolovits, and Anne Hunter for their guidance on classes, and helping me navigate through MIT.

I would like to thank Jolie Chang for her friendship, support, and for having confidence in me. Alex Park, for his friendship and his extensive knowledge in statistical modeling. Rich Moy, my thesis buddy, for keeping me on track and focused. Fred Lee, for his confidence and infections positive attitude. And Justin Lin, for his unique perspective on life. You guys have been my pillars of support during the ups and downs of this incredible journey.

Thanks also to Eugene Chiu, Yifung Lin, Jon Lau, Edmund Chou, Peter Weng, Alen Chen, Wen Fu, Cat Chen, Fung Teh, Jordan Low, the JSU members, AAA Volleyball, the next house crew, and to all of my countless friends that I have met at MIT. Every single one of you has made these last 5 years at MIT an unforgettable experience. I am very grateful to have shared so much with you.

Lastly and most importantly, my mother, my father and my sister. Your unfailing confidence in my abilities and the values you have instilled in me have brought me to where I am today. You give me the inspiration and courage to keep reaching for my dreams.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Continued improvements in speech recognition technology are allowing speech recognition systems to enter the commercial sector, where they are utilized in a variety of settings. Such settings include the public domain, where speech recognition systems must interact smoothly with untrained speakers. In order to allow untrained humans to interact more naturally and effectively with computers, recognition systems must take steps to understand spontaneous speech. A major hurdle for this problem involves segmenting the speech signal into meaningful semantic units.

In the ideal case, an untrained speaker should be able to interact with computer systems in the same rich, conversational manner humans use to communicate with each other. This involves being able to pause mid-sentence to formulate one's thoughts, restarting a badly worded sentences, and speaking in an spontaneous manner. In reality however, many spoken language systems restrict the user to limited interactions because current speech systems have difficulty dealing with the rich variation in spontaneous speech. At the utterance segmentation task, the primary challenge lies in the irregular and unpredictable nature of spontaneous speech, which makes it difficult to tell where commands, or utterances, begin and end.

Effective speech segmentation is particularly important in spontaneous speech interactions of longer durations and higher complexity, since users often stop mid-sentence in order to formulate the remainder of their thoughts. Ideally, an utterance segmenter would take advantage of a variety of information sources to determine sentence endpoints. Current speech segmenters have difficulty handling such utterances because they focus solely on signal level information without using other cues. The task of this thesis is to improve the signal level segmenter of the speech system, and combine it with higher level information in order to segment speech into meaningful semantic units.

1.2 Background

1.2.1 Speech Endpointing

The goal of speech endpoint detection is to separate acoustic events of interest in a continuously recorded signal from other parts of the signal. [6]. The problem we tackle in this thesis is a related, but more challenging problem of detecting the endpoint of an utterance–or command–which may not correspond to speech endpoints. As part of the utterance endpointing–or utterance segmentation–task, speech endpointing is a related research area, and is the basis for our signal level utterance segmenter for our speech recognition system.

Current speech systems use a variety of methods to classify the incoming audio as speech or nonspeech sections. Short time energy based endpointers have been a popular and computationally inexpensive way to determine utterance segments, and are present in the literature combined with a variety of augmentations [4, 7]. Other standard models, including our system use a slight variation of the same measured features as those used for recognition [6].

As a more detailed example of a standard speech endpointer, the JUPITER telephony system [3], maintained by the Spoken Language Systems Group at the Massachusetts Institute of Technology uses energy in order to perform utterance endpointing. A threshold duration of low energy is used to determine the end of an utterance, at which point an auditory cue is given to the speaker to indicate that the recognition system has determined an utterance endpoint. If the user has not finished with his command when the endpoint is determined, he must wait until JUPITER has responded to the incomplete utterance before repeating his command, this time being careful to complete the entire utterance before the endpointer determines an endpoint. This system sidesteps some of the more complex aspects of utterance segmentation by constraining the user interaction to a strict turn taking dialog format.

As speech recognition systems become widely used, it becomes desirable to have an utterance endpointing system which allows users to have less constrained, natural interactions with the system.

1.2.2 Language Models

A growing body of research in the Cognitive Science indicates that higher-level processes in our brain affect our perceptual capabilities at astonishingly basic levels [5]. In vision, this provides support for Ullman's classic theory on how perception may be a simultaneous bi-directional bottom-up and top-down search. Higher order processes can affect lower order processes by providing context for the sensory inputs, while lower order processes feed higher level processes information from its sensory inputs [11]. Because evidence points to higher level processes having an impact on lower level tasks such as speech segmentation, we incorporate linguistic level information into the task of utterance segmentation.

A simple example can illustrate the effectiveness of incorporating syntax into our segmentation task. When a speaker pauses in the middle of a sentence, we can often guess that he is not done speaking. Many factors contribute to this decision, but one of the stronger cues is the syntax and semantics of the language. This information seems to be related to our understanding of grammar. We recognize that the sentence is grammatically incomplete, so we disregard the long pause duration, and wait for the sentence to be completed. An example below which is transcribed from actual speech data illustrates this phenomenon.

"please" *"place"* *"a magnet over the large ball"*

From just the syntax and semantics of the words above, we can say with a high degree of confidence that the three segments belong in one utterance. No additional acoustic information was necessary for this inference, and suggests grammar can be a strong indicator of semantic segmentation.

There are several ways in which speech recognition systems currently take advantage of the syntactic structure in language. Bigram and trigram word probability models are incorporated into the recognizer's evaluation mechanism when it outputs the most likely sequence of words. If the speech is constrained to a small set of grammatical constructions, templates may also be used to narrow down the possibilities. Syntactic analysis has also been used to prune out unlikely speech candidates returned by the decoder.

In language modeling experiments related to speech recognition research, Ferrer, Shriberg and Stolcke have independently used Language Models to determine utterance segmentations [10] as part of their research in the effectiveness in utterance endpointing. The language model they use however, is based on direct n-gram modeling of words. In Chapter 3, we present a more generalized method using grammar based word classes which yield generalized and significant results for the HelpingHand corpus.

1.2.3 Cognitive Machines Speech Recognition System

The research presented in the first half of the thesis involves improving the signal level utterance segmenter, an integral component of the spontaneous speech recognition system developed in the Cognitive Machines Group. This section will explain the major components of the overall system to serve to as a backdrop for the research on signal level utterance segmentation.

The recognition system constitutes of 4 main components as shown in Figure 1-1: a front end acoustic analyzer, the utterance segmentation module, and the decoder.

- Acoustic Front-End: The front end takes in audio and computes feature vectors based on the audio every 10 ms. These feature vectors, or frames, are sent

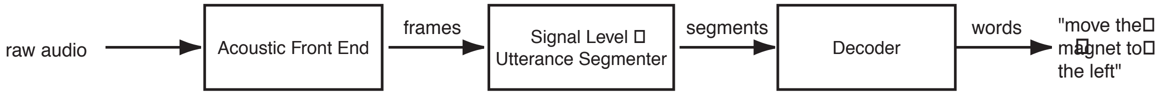


Figure 1-1: Block diagram of the Speech Recognition System

across the network to the signal level utterance segmenter.

- Signal Level Utterance Segmenter: The signal level utterance segmenter uses uniphone models of speech and silence to segment the incoming frames into continuous speech and nonspeech segments. The frames representing speech are sent on to the decoder.

- Decoder: Using the feature vectors in the speech frames sent by the signal level utterance segmenter, the decoder uses acoustic and language model estimates to perform a search on the entire vocabulary space. The decoder outputs the most probable word sequence based on the frames it has received.

The signal level utterance segmenter's primary task is to identify all speech segments from the incoming audio, and to do its best trying to make each speech segment correspond to a utterance segment.

1.3 Problem Description

The goal of this project is to improve speech recognition by creating a module to segment speech into audio segments that represent semantic units. It is a bi-directional approach which integrates two differing viewpoints in the AI community, the top down approach examining high level cues, and a bottom up approach segmenting based on low level features. The lower level component is represented by the signal level utterance segmenter, and the higher level component is represented by syntax and grammatical structures. We attempt to take a small aspect of speech recognition–utterance segmentation–and show the potential effectiveness of multiple cue integration in tackling challenges in speech recognition.

Chapter 2

Utterance Segmentation

2.1 Introduction

The signal level utterance segmenter classifies the audio stream into continuous sections of silence and speech, sending the speech portion of the signal upstream for further processing by the decoder as shown in Figure 1-1.

Accurate extraction of speech elements from the audio signal is crucial, as the performance of the segmenter directly affects the performance of the rest of the speech system. Using an existing frame-based classifier as a base model, this chapter explores design augmentation, detailed analysis, identification and optimization of the parameters which affect utterance segmentation accuracy.

2.2 Original Signal Level Segmenter

Silence and non-silence segmentation begins by modeling the acoustic characteristics of non-silence and silence, and classifying speech according to these models. The acoustic models for speech and silence are trained from the TIMIT Acoustic-Phonetic Continuous Speech Corpora, a standard collection of broadband recordings of 630 speakers in 8 major dialects of American English. For each model, a set of 39 features are extracted every 10 ms with a 20ms Hamming window. The features represent the 12 standard Mel Frequency Cepstral Coefficients (MFCC), the log energy calculated over the window, and the corresponding first and second derivatives calculated using the MFCC values of neighboring frames. Initial and final frames without neighboring frames use derivative information copied over from the nearest vector. Two statistical models, a silence phone, and a speech phone are approximated based on these 39 dimension feature vectors obtained from the TIMIT corpus for each type of phone. These phone models for speech and silence are then used to label incoming audio frames as speech or silence. Frames with a higher probability of being speech are labeled speech, and frames with higher probability of being silence are labeled as silence.

The signal level segmenter divides the audio stream into speech and silence using a four-state finite state machine to smooth out occasional silence frames in speech segments, and speech frames in silence segments. The model is shown in Figure 2-1.

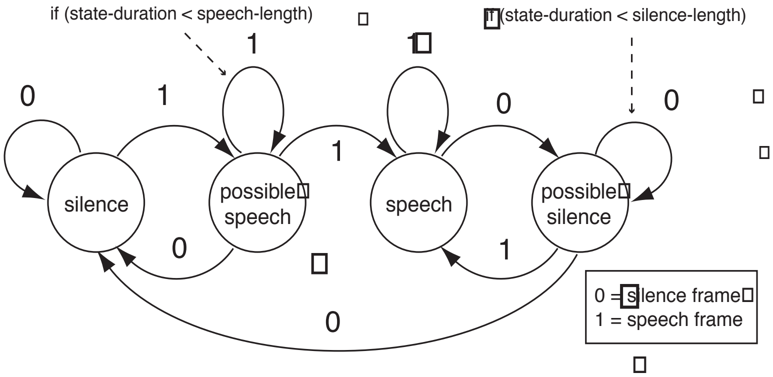


Figure 2-1: Four State Speech Silence Model

The segmenter is initialized in the silence state, and transitions to possible speech, speech, then possible silence based on the input signal. If the state machine is in the speech or silence state, it will stay in that state in the next frame as long as the frame level classifier makes the same classification for the next frame. If a differing classification is given for the the next frame, it switches to the possible-speech or possible-silence state. The possible-speech and possible-silence states insure that a prespecified number of consecutive frames are necessary in order for a transition between speech and silence segments to occur. The segmenter stays in the possible-speech or possible-silence state until a prespecified number of consecutive frames with the opposite prediction has occurred. If this specified number of frames do not occur, the state reverts down to the previous speech or silence state. If the minimum number of consecutive frames with the opposite prediction occurs, we transition into the opposite state. In the speech state, feature vectors of the incoming signal are sent on to other parts of the program, whereas in the silence state, information is not sent on to other parts of the system.

The original segmenter has three parameters which can be modified to change the behavior of the system. The list below illustrates each of the parameters and their function, along with a fourth parameter, EXTRA-SILENCE-LENGTH-BACK, which is subsequently added in the design augmentation phase discussed in Section 2.4.

- EXTRA-SILENCE-LENGTH-FRONT - The number of frames of the audio signal before a speech boundary which is inserted into the audio segment sent to the decoder.

- EXTRA-SILENCE-LENGTH-BACK - The number of frames of the audio signal after an end-speech boundary which is inserted into the audio segment sent to the decoder.

- MAX-POSSIBLE-SPEECH-DURATION (speech-length) - The number of consecutive frames of speech necessary before determining entry into the speech state.

- MAX-POSSIBLE-SILENCE-DURATION (silence-length) - The number of consecutive frames of silence necessary before exiting from a speech state into silence.

MAX-POSSIBLE-SPEECH-DURATION (speech-length) and MAX-POSSIBLE-SILENCE-DURATION (silence-length) are the two parameters determine the state transitions between speech and silence states. The two EXTRA-SILENCE-LENGTH parameters for the front and back attempt to approximate a cache which stores previous audio information and can replay that information for the audio decoder once the system determines that human speech is being processed. Of these four parameters, only two crucially affect segment boundary determination: speech-length and silence-length.

2.3 Limitation of the Original Segmenter

The parameters in the initial version of the utterance segmenter are hand set at values based on qualitative observations by a human experimenter. Although these parameters perform relatively well, there is no formal search method by which the values are obtained, nor is there a formally defined evaluation metric to measure the performance of these hand-obtained parameters. The diversity of speaking styles found in spontaneous speech makes it difficult to be certain that the parameters perform well in all situations. The testing is limited to a few speech characteristics, and may not necessarily be optimal for a wide variety of speaking styles. In order to evaluate and obtain optimal segmentation parameters, formal and automated evaluation metrics must be used to measure utterance segmenter performance.

Before proceeding into the optimization of segmenter parameters, we detail several design augmentations to extend the functionality of the segmenter, and to facilitate our search for optimal parameters.

2.4 Adding New Features

First, we improve the system to do offline segmentations of large audio files in order to reproduce segmentation results from live recordings. To resolve this problem, a TCP/IP style message router is designed to regulate data flow and allow offline segmentation. By using this traffic regulating protocol, we prevent out-of-memory errors while eliminating any idle time in the signal level segmenter.

In addition, we solve several boundary offset errors between the recorded raw audio and the segment transcription. These errors are introduced by the derivative calculations performed when creating the feature vectors. We also insert an additional tuning parameter to pad the back of speech segments to capture any trailing voiced sounds.

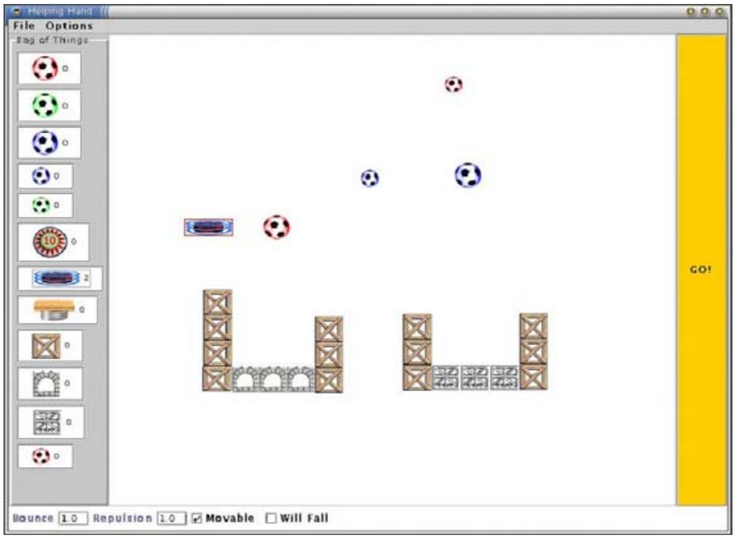
The above modifications resulted in a utterance segmenter with increased functionality. However, there are cases when unvoiced speech segments, particularly at the beginnings of utterances, were not properly identified. To combat this result, we briefly experimented with an alternative frame-level classifier for speech and silence frames, which more accurately captured unvoiced phones. The frame level analysis of this model is provided in Appendix A.

2.5 The HelpingHand Corpus

The HelpingHand corpus is a collection of thirty to sixty minute sessions where a human subject is brought in to interact with the computer to solve a series of puzzles. These puzzles, shown in Figure 2-2 consist of objects that must be moved across the screen into the appropriate configurations. The subject directs the computer through speech, and the program's response is simulated by an operator in a hidden room adjacent to the subject. The computer-directed spontaneous speech corpus was chosen for its relevance to the context of the problem we would like to solve: segmenting speech commands directed to a computer in a conversational manner. This corpus will be used to find optimal parameters for the HelpingHand corpus.

2.6 Defining an Evaluation Metric for Segmentation

This section's goal is to find optimal numbers for these two parameters speech-length and silence-length (defined in Section 2.2) by directly comparing the segmentation performance over a range of setting combinations. By picking pairs of values for the segmenter and scoring the performance based on these values, we can pinpoint the optimal settings for the segmenter for the two variables. First, we construct an evaluation metric to measure the closeness of a particular segmentation to the 'ideal'



Bounce 1.0 Repulsion 1.0 ☑ Movable □ Will Fall

Figure 2-2: Screenshot of a HelpingHand Session.

segmentation boundaries.

2.6.1 Evaluation Metric

The evaluation metric must satisfy several criteria. First, it must detect and categorize different types of errors such as missed segments and over-segmentations. It must not be computationally expensive, because our algorithm must evaluate a score for audio sessions with approximately 360,000 frames. Finally, it must allow for some reasonable amount of error when determining whether a boundary has been correctly identified.

This algorithm examines boundary events in the candidate transcription and classifies the events as one of three categories: correct, false positive and false negative. In order for a boundary to be classified as correct, it must be within some range k of a correct boundary b, be the same type (speech onset or offset), and be closer to b than any other candidate boundary. A false positive is recorded when a boundary in the candidate transcript does not correspond to a boundary in the ideal transcription. A false negative is logged when a boundary in the ideal segmentation is not detected in the candidate segmentation. If there are multiple boundaries that fall within distance

k of a correct boundary in the ideal transcript, the closest one is marked as correct, and all others are recorded as false positives. The maximum number of errors which can occur when comparing a candidate segmentation with the ideal segmentation is the sum of the number of boundaries in both segmentations. The value for k is chosen at 3200 raw audio samples, or 0.2 seconds, based on human judgment. This algorithm allows us to assign a performance metric to the segmentations obtained by the segmenter.

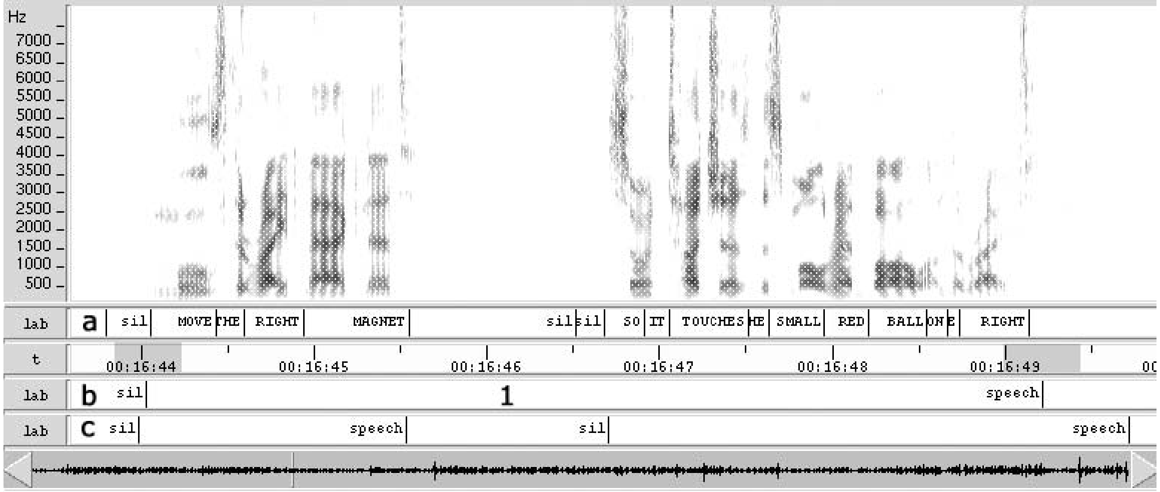


Figure 2-3: Screenshot illustrating scoring algorithm

Figure 2-3 illustrates how the evaluation metric works. The audio segment taken from a session of the HelpingHand corpus illustrates a speaker telling the computer to “move the right magnet....so it touches the small red ball on the right”. The letters (a), (b) and (c) on the bottom left of the diagram display the transcripts corresponding to the audio. (a) represents the word level transcription obtained from force-aligning utterance segments to their transcript. (b) represents the human annotation of utterance boundaries, spanning the entire utterance. (c) represents the machine segmentation of the utterance, where the mid utterance pause between the word “magnet” and “so” causes the signal level segmenter to over segment this utterance into two segments. The gray areas spanning 0.2 seconds on either side of the human segmentations indicate the range which the segmentation from the machine segmenter must be in. In this specific example, the first speech-onset boundary would be classified as correct, the two mistaken boundaries near (1) would be classified as incorrect false positives, and the third boundary, would also be classified as incorrect, as it is close, but not close enough to the human segmentation to counted as a correct segmentation. This particular example shows three false positives and one false negative.

2.7 Collection of Ground-Truth using Manual Segmentation

We first create an ideal segmentation for sessions of the HelpingHand corpus by having a human listener determine utterance endpoints and startings of commands. The human listener annotates portions of an audio session into one of three classifications: speech, silence or misc. Speech segments represent a complete utterance segment as judged by the human evaluator, the silence segments represents portions with little or no noise, and misc segments represent all other noise not pertinent to the HelpingHand task. These include sounds such as coughs, mumbles, background noise and heavy breathing. The definition of a segment is vague in some cases, and even human evaluations can disagree on cases where multiple commands are strung together, when speakers restart, or speak ungrammatically. We attempt to standardize how each of these cases are handled for consistency across the corpus. Yet even with such standardization there are portions which are difficult to unambiguously classify. For the most part however, these borderline cases are few and far between, and most segment endings are largely undisputed. Because the number of unambiguous segments greatly exceeds the number of ambiguous segments, any statistical deviation resulting from the misclassification of these ambiguous segments should be negligible.

2.7.1 Creating the Hand Segmentations

We first create an ideal segmentation for sessions of the HelpingHand corpus by having a human listener determine utterance endpoints and startings of commands. This was done for five sessions of the HelpingHand corpus. The human listener annotates portions of an audio session into one of three classifications: speech, silence or misc. Speech segments represent a complete utterance segment as judged by the human evaluator, the silence segments represents portions with little or no noise, and misc segments represent all other noise not pertinent to the HelpingHand task. These include sounds such as coughs, mumbles, background noise and heavy breathing. This was done for five sessions of the HelpingHand corpus.

2.7.2 Analysis

Analyzing these segmentations, we find an interesting pattern. For the most part, segments of speech which belong together have a strong characteristic of temporal continuity. That is, speech that belongs in a single command are usually closer together than speech which belong in two separate commands.

There were two notable exceptions. One occurs when a user has not fully formulated his thoughts into a clear sentence. In this case, we get significant pauses in the middle of a sentence while the user stops to think about the next actions as in the example in the introduction. These phrases are segmented separately by the segmenter since the number of consecutive silence frames exceeds silence-length and the segmenter loops into a silence state. These situations, as mentioned earlier, represent the bulk of the segmentation errors. At the other end of the spectrum is another kind of error where the segmenter fails to detect a segmentation boundary because the user strings together two commands with little pause between them. This error occurs less frequently than the first case, and is usually associated with the user becoming impatient with the system, or when he already knows what series of commands he wants performed.

From this cursory analysis, several indication points arise. However, humans have no trouble parsing and segmenting the user's commands (as evidenced by the fact that for the HelpingHand recordings, a human interpreter was able to perform all the commands directed by the user.) One important indicator seems to be pause duration. In general this criteria is sufficient to distinguish different sentences/commands/ideas. Another major criteria is grammatical structure. Even when there are large pauses between segments, we see that grammar can link the two together.

2.8 Evaluation of Previous Segmenter

Using the scoring metric, and the manual segmentations prepared in the previous two sections, we perform an initial analysis of the original signal level utterance segmenter parameters, which are hand tweaked to silence-length=30, speech-length=10.

	original
False Negative Boundaries	682
False Positive Boundaries	1976
Total Boundary Error	2658

We have a total error score of 2658, and an uneven distribution between false negatives and false positives, with approximately 75% of the error constituting falsely recognized utterance onsets and offsets. These numbers serve as a standard to measure improvement gains of the subsequent optimizations.

2.9 Optimization of FSM Parameters

Using our scoring algorithm, we obtain segmentations over a range of values for speech-length and silence-length. The total error from each session is summed and graphed across different segmenter settings in Figure 2-4. The total error scoring equation used to generate the graph is shown in Equation 2.1:

$$TotalError(x, y, k) = \sum_{n \in \mathcal{S}} C_{fp}(n, x, y, k) + \sum_{n \in \mathcal{S}} C_{fn}(n, x, y, k) \tag{2.1}$$

where k is the error tolerance boundary, $\mathcal{S} = \{s_1, s_2 ...\}$ the set of sessions over which the error is calculated, x is the value for the speech-length parameter, and y is the value for the silence-length parameter.

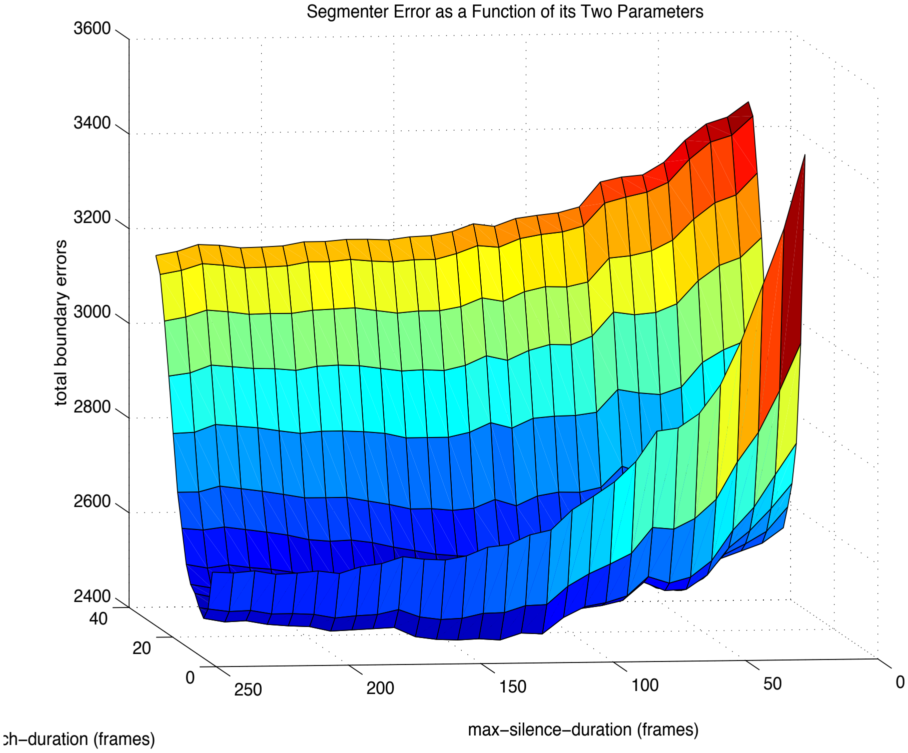


Figure 2-4: Graph representing total error over different parameters.

The graph 2-4 has several notable features. Aside from small local disturbances, there seems to be a nice concave global optimum near the minimum error point where speech-length=9 and silence-length=141. From the half-cylindrical shape of the graph, we see that the more sensitive parameter is speech-length, and as we move away to the outer ranges of this variable, the error score increases significantly. In contrast, segmentation error seems less tied to the silence-length variable. Compared to the previous hand optimized values of 10 and 30 respectively, the speech-length value largely remains the same. However, the optimal value for the silence-length variable more than quadruples. Since the silence-length variable represents the length of time the segmenter waits before signifying the end of the utterance, we see that the new parameters waits significantly longer before concluding that the speaker is done with his utterance. By waiting 1.4 seconds, this new parameter can expect to alleviate the first category of errors generated when the segmenter places utterance boundaries in places where users stop to think mid-sentence. The substantially lengthened silence-length parameter should immediately benefit one application in the Cognitive Machines Group, where the system relies on the signal level segmenter to properly segment the audio into single utterances. Many users of this application have experienced a "breathlessness" during their interactions with the system because the short silence-length variable forced users to speak their commands quickly and with little pause. The new results are summarized in Table 2.1.

	original	optimal score
False Negative Boundaries	682	1371
False Positive Boundaries	1976	1047
Total Boundary Error	2658	2420

Table 2.1: Error Comparison Table for Minimum Error Parameters

The values which we obtained in this section represent the minimum total error over the HelpingHand sessions, which means that there is no distinction between the types of misses being counted, and it is likely that there are a significant number of false negatives along with false positives. False negatives result in mistakenly concatenating two separate utterances together, or totally missing the utterance altogether. Such errors are more detrimental to the system, as these missed segments are irrecoverable. With our new settings, we notice that there are an increased number of occasions in which separate utterances are mistakenly concatenated together. We address these shortcomings in Section 2.11. Before this, we incorporate another filtering step for the segmenter.

2.10 Extending the Segmenter by Incorporating Energy Bounds

Many speech endpoint detection mechanisms use energy values in order to determine endpoints of speech. The segmenter in this thesis takes a different approach and models speech and non-speech uniphones in order to classify frames. A question remains, however, as to whether an additional filter step based upon a threshold energy criterion such as in other speech endpoint systems would improve the segmentation accuracy by eliminating the detection of segmentations falsely classified as speech by our model.

In order to evaluate whether the segmenter would benefit from such a system, we attempt to characterize the energy levels of false positive segments in contrast to correctly identified segments.

2.10.1 Procedure

Using the optimal values obtained above, we would like to characterize all segmentations as either a correct segment, or a segment which was falsely recognized as speech. By examining the distribution of the average log energy values for each segment, we hope to determine whether there is a significant correlation between energy level, and whether a segment has been correctly identified as speech. Unfortunately, not all segments fall into one of these two classes. There are many cases where the segments obtained by the segmenter are partially correct (e.g. the utterance onset is correctly identified but the utterance termination is not correctly identified.) In these cases, there is no clear way classify the segment. Instead, such segmentations are ignored, and we consider two types of segmentations.

- Segments which are completely contained within the boundaries marked by the ideal segmentation.

- Segments which are completely outside the boundaries marked as speech in the ideal segmentation.

There are two drawbacks of limiting our analysis to the above segments. It throws out many samples, creating the risk that the data is too sparse to be useful for analysis. Secondly, it is not clear whether the criteria of looking at all segments which are completely contained within the segment boundaries will not bias the energy results for correctly identified segments higher. However, restricting our analysis to only segments which were totally correctly identified makes our sample set too small for analysis. Our criteria for sample selection is meant to keep the sample set accurate and still reasonably large.

The average log energy of the two types of segments are graphed in a histogram in Figure 2-5 to compare distributions. The graph shows false positive subsegments having a significantly lower average energy than correctly identified segments on average. This suggests that an additional filter based on average energy can be a promising technique. The two distributions are not completely separate, suggesting that any sort of threshold choice must be extremely conservative, or we must be prepared to mistakenly eliminate some valid utterance segments.

In the next section, we describe the energy cutoff feature added into the segmenter, and the results for incorporating the new feature.

2.10.2 Implementing the Energy Thresholding Feature of the Segmenter

The new energy cutoff filter must conform to several properties. Because it is both a real-time and offline system, it is important that it classify a segment as quickly as possible, so that other components downstream can perform further tasks. As such, it is not practical for the segmenter to wait for the entire utterance to end, calculate the average energy of the segment, and make a decision based upon the average. Such a process either overly complicates the message passing mechanism between the segmenter and processes downstream, or it introduces an undesirable delay into the pipeline.

Instead of calculating the average over the entire segment, we examine the average

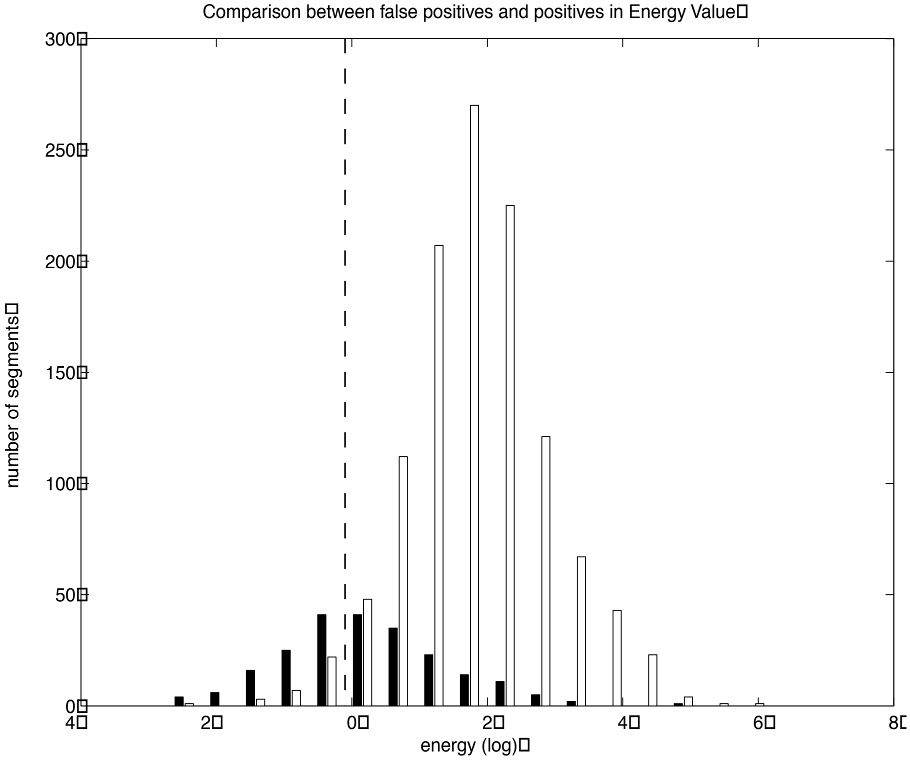


Figure 2-5: Energy difference between mis-segmented and correctly segmented boundaries.

energy of the first few frames of a proposed utterance segment. If the energy of the beginning sequence of frames do not overcome a certain threshold, we never enter into a speech state. If it does, we send the first frames, along with the rest of the frames onto the decoder for speech processing.

A conservative threshold of -0.33, shown as a dotted line in Figure 2-5 is chosen based on the energy histogram obtained above, and the performance of the segmenter with the additional filter is run to obtain new segmentations. Table 2.2 illustrates the improvements in total error in comparison to the segmenter excluding the energy thresholding. Negative numbers show an improvement, and positive numbers show a decrease in segmentation performance due to the additional energy thresholding.

The energy thresholding parameter for the data in Table 2.2 is set to K=-0.33, and the vertical axis represents the silence-length variable, and the horizontal axis represents the speech-length variable. Both numbers are in frames. These numbers illustrate that, while overall, segmentation performance improves, certain sections, particularly where the silence-length parameters are longer, we get a slightly worse performance in comparison to the sections with shorter silence-length parameters. The reason is that the segmenters with a shorter silence-length parameter tend to over-segment the speech into multiple short utterances. In these cases, there is an increased chance that a good portion of these segmentations are false segments which contain no speech. It seems the thresholding is effective at eliminating these sorts of mis-segments. In contrast, with the longer segments, it seems that the algorithm contributes more toward increasing the error by mis-identifying the entry point for an utterance onset.

For the optimal parameters obtained in the previous section, (9 and 141) we have a marginal improvement of 2 less errors (shown in Table 2.3) as compared to the segmenter without energy thresholding. These numbers seem to disappoint our previous analysis of the effectiveness of energy thresholding, but as our next section shows, energy thresholding becomes much more effective for different optimal parameters obtained when a weighting is added to the scoring function in order to insure that we minimize information loss from failing to detect a speech segment

	speech-length (frames)						
	5	6	7	8	9	10	11
25	-68	-42	-26	-22	-24	-24	-12
29	-64	-38	-22	-18	-22	-22	-10
33	-54	-36	-20	-16	-22	-22	-10
37	-46	-28	-16	-16	-22	-22	-10
41	-38	-20	-10	-12	-18	-18	-6
45	-34	-18	-8	-10	-16	-16	-6
49	-32	-16	-6	-10	-16	-16	-6
53	-26	-12	0	-6	-12	-14	-4
57	-24	-12	0	-6	-12	-14	-4
61	-24	-12	0	-6	-12	-14	-4
65	-22	-12	0	-6	-12	-14	-4
69	-22	-10	2	-4	-12	-12	-4
73	-20	-10	2	-4	-12	-12	-4
77	-18	-10	2	-4	-12	-12	-4
81	-16	-8	2	-4	-12	-12	-4
85	-12	-6	2	-4	-12	-12	-4
89	-10	-4	2	-4	-10	-10	-2
93	-10	-4	2	-4	-10	-10	-2
97	-8	-2	2	-4	-10	-10	-2
101	-6	-2	2	-4	-10	-10	-2
105	-8	0	2	-4	-10	-10	-2
109	-10	-2	0	-6	-10	-10	-2
113	-8	0	2	-4	-8	-8	-2
117	-8	0	2	-4	-6	-6	-2
121	-6	2	4	-2	-4	-4	-2
125	-6	2	4	-2	-4	-2	0
129	-4	4	6	0	-2	-2	0
133	-4	4	6	0	-2	-2	0
137	-4	4	6	0	-2	-2	0
141	-4	4	6	0	-2	-2	0
145	-4	4	6	0	-2	-2	0
149	-4	4	6	0	0	0	2
153	0	8	10	2	2	2	4
157	0	8	10	2	2	2	4
161	0	8	10	2	2	2	4
165	0	8	10	2	2	2	4
169	0	8	10	2	2	2	4
173	0	8	10	2	2	2	4
177	0	8	10	2	2	2	4

Table 2.2: Table representing the difference in error between the original and energy thresholded segments.

	original	optimal score	optimal+energy
False Negative Boundaries	682	1370	1371
False Positive Boundaries	1976	1050	1047
Total Boundary Error	2658	2420	2418

Table 2.3: Error Comparison Table for Energy Thresholding

altogether.

2.11 Weighting the Evaluation Function

Our optimal segmenter parameters were obtained using a simple error function that treated both false positive boundaries and false negative boundaries with the same importance. In reality however, it is more important for the segmenter to catch all the existing speech segments at the cost of over-segmentation or mis-identifying noise as speech. If there are segments of speech which do not get identified (false negatives), this speech information is irrevocably lost to all systems downstream. In contrast, over-segmented boundaries can always be analyzed downstream and discarded or combined accordingly. Recalling that the total error surface obtained before was the sum of two separate surfaces, we can change the optimal location by valuing one error higher than the other.

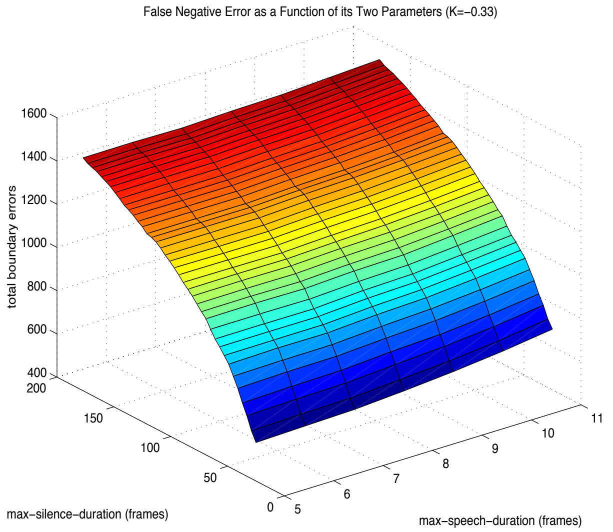


Figure 2-6: Graph of false negative errors

Figures 2-7 and 2-6 illustrates the two error surfaces in isolation. As expected, false negative errors increase as both parameters are increased, and the false positives

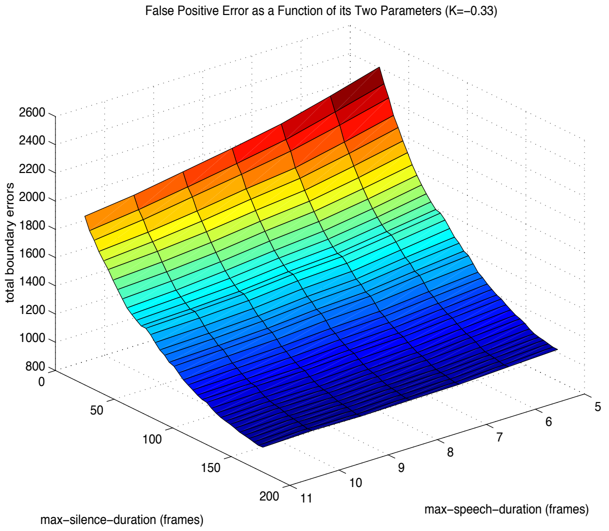


Figure 2-7: Graph of false positive errors

decrease as both parameters are decreased. From these graphs, it is expected that as we penalize false negatives, our values, particularly for silence-length, should decrease.

To penalize false negatives more than false positives, a cost function is assigned to the false negative errors. Weighted total error is now described in the following function.

$$TotalError(x, y) = \sum_{n \in \mathcal{S}} w_1 C_{fp}(n, x, y) + \sum_{n \in \mathcal{S}} w_2 C_{fn}(n, x, y) \tag{2.2}$$

This function is identical to before, except that now a weighting w_1 and w_2 assign costs to the two types of errors. Since we are only concerned with the cost ratio relative to the two errors and not the absolute cost, we can re-normalize the equation and rewrite the equation as a function of one weighting parameter $W = w_1/w_2$ on the false negative.

$$TotalError(x, y) = \sum_{n \in \mathcal{S}} C_{fp}(n, x, y) + W \sum_{n \in \mathcal{S}} C_{fn}(n, x, y) \tag{2.3}$$

Here, W represents the number of false positives which equal to the cost of a false negative. The higher W is, the larger the penalty associated with having false negative errors.

The optimal obtained in previous section is equivalent to evaluating our new function with $W = 1$. We obtain a range of numbers which represent the optimal over a variety of different weightings. The parameters obtained are shown in the table below.

W	speech-length	silence-length
1.0	9	141
1.2	9	121
1.3	10	61
1.4	10	61
1.5	10	29

Table 2.4: Optimal parameters for different weights

Table 2.4 is essentially the data points for a curve representing the "Optimal Parameter Path" of the speech silence segmenter as we choose different penalties (in this case from 1-1.5) for missed speech boundaries. From the data in Table 2.4 we observe that a optimum parameter setting of 10 and 61 exists over a significant range of the weightings.

The weighting of $W = 1.3$ is chosen to yield the parameters (speech-length = 10, silence-length = 61). As guessed from our analysis of Table 2.4 this set of values are stable across a wide range of penalty weighting, and yield the results shown in Table 2.5.

	original	optimal	opt+enrgy	wtd opt+enrgy
False Negative Boundaries	682	1370	1371	1052
False Positive Boundaries	1976	1050	1047	1377
Total Boundary Error	2658	2420	2418	2429

Table 2.5: Error Comparison Table for Weighted Optimal

2.12 Performance Summary

Instead of pinpointing a single set of optimal segmentation parameters for the signal level utterance segmenter, we have discovered range of optimal settings which can be varied depending on the behavior that is desired from this system. At the aggressive end, the utterance segmenter has a short silence-length parameter, causing it to easily exit speech states. As a result, brief pauses are classified as utterance endpoints, resulting in segmenter behavior ideal for applications with short utterances and a small lag time between each utterance. At the other end of the spectrum, the large value of the silence-length parameter causes the utterance segmenter to allow long pauses to occur without signaling an endpoint. This setting may be ideal for applications where the task is complex, forcing the user to pause mid-utterance in order to formulate his next thoughts, and where the delay time between each utterance is long. At the most aggressive end of this tuning range, we discovered parameters which were very similar to the original human set parameters for the segmenter. Our unweighted, optimal scoring parameters however, were at the other end of the tuning spectrum, with a silence-length parameter more than four times longer than the original.

silence-length	141
speech-length	9
Energy Threshold	-0.33
Error improvement	238
Percentage Improvement	9%

Table 2.6: Minimal error parameters

These scores, however, did not reflect the fact that it is important to weight false negatives higher than false positives. With a new weighted scoring system and a higher energy threshold which yielded significant gains.

Our final settings, which has a silence-length roughly twice as long as the original, show a 8.6% improvement in boundary detection over the original signal level segmenter, and exhibits favorable performance both over the HelpingHand corpus, and in the nine speech system.

silence-length	141
speech-length	9
Energy Threshold	1.00
Error improvement	229
Percentage Improvement	8.6%

Table 2.7: Final parameters based on weighted scoring

This section presented improvements to the front end module for utterance segmentation of an incoming audio signal at the signal level. Starting with a four-state segmentation model, we stabilize the behavior of the segmenter and extend its functionality. In addition, we use this basic model as a departure point to explore a variety of additional improvements including energy thresholding, a three-model classifier, and average subtraction techniques. Each of these variations exhibited improvements in performance. Most importantly however, this section presents a detailed characterization of each parameter of the segmenter. As a product of this detailed analysis, we obtain an optimal set of tuning values for the segmenter which allows any user to easily readjust the behavior of this component as the application tasks change. These parameters can also change as additional higher level components are added to aid in the utterance segmentation task. The next chapter presents one such higher level system which utilizes word classes to predict utterance endings for the HelpingHand task.

Chapter 3

Higher Level Cues

3.1 Introduction

This section deals with higher level aspects of utterances and the exploration of linguistic structure to aid in utterance endpointing. The previous chapter outlined the design and analysis of a signal level utterance segmenter. The front end system described in the previous chapter functions as a filter to extract possible speech sections from silence portions of the incoming audio stream. The primary method for utterance segmentation at this level is based on extracting and classifying features from the audio signal which are then combined at a slightly higher level to identify continuous sections of non-speech and speech. Pause and energy parameters are used to fine-tune the sensitivity and tolerance of the segmentations.

At this level, such analysis does not capture the grammatical content of the utterance, which is important to determining utterance endpoints. As mentioned in the HelpingHand corpus analysis in Chapter Two, two types of segmentation errors are exceptions to the strong temporal continuity of words which belong in the same utterance. This chapter is aimed at addressing these segmentation errors by examining the use of grammar as a high level indicator to utterance endpointing.

Although it is intuitively appealing to use grammar to segment utterances, it is not clear to what extent the sequence of words alone contributes to utterance endpointing. Before attempting to use grammar to assist in utterance endpointing, we perform an informal experiment to qualitatively assess the contribution of grammar in determining utterance endpoints.

3.2 Experiment

Our signal level silence detector effectively divides the audio stream into continuous silence and non-silence segments. Our task is to take these segmentations and group them together so that segments which belong in the same utterance are together, and those of different utterances are in different groups. In order to isolate the the effectiveness of the higher level cues, we assume an ideal decoder and use hand transcriptions in place of speech decoder output when evaluating the performance of our indicators. First, to test how important grammar is in this task, we ask human subjects to perform the same task based on grammatical information alone.

The helpinghand corpus contains transcribed audio segments which were obtained by the Silence-Non-Silence classifier of the previous chapter. The parameters for the segmenter were chosen to aggressively segment utterances into smaller pieces, in order to increase the difficulty of the task. We ask human subjects to indicate utterance endpoints by grouping together audio segments which belong to the same utterance segment by examining only the transcript. As the subjects do not have access to any additional information provided by the audio, the strength of syntactic and semantic indicators alone in utterance segmentation can be assessed by their performance on this task.

In preparation of this experiment, the transcriptions for the non-silence segments are concatenated into a single file, with each transcription on a separate line. These transcriptions are given to human evaluators who are asked to mark off utterance segmentations based on these transcriptions alone. The questionnaire can be found in Appendix B, and the results of this experiment are outlined in table below.

The average error represents the average number of total (false positives and false negatives) missed boundaries between each of the eight individuals surveyed. The break-count represents the number of boundaries in the correct transcription, and the

average error	9.75
σ	5.07
break-count	211
accuracy	95.4%

Table 3.1: Experimental results from the transcription based utterance segmentation task.

accuracy entry loosely represents the average percentage of time a user's boundary markings corresponded with the true boundary markings.

The numbers shown in Table 3.1 indicate that humans can determine utterance segmentation boundaries quite accurately without audio signal information beyond the identification of the words which were spoken. This suggests that the sort of information used by human subjects to determine these segments are present in semantic and syntactic information. Although the semantic content of the words undoubtedly plays a large part in a human's decision of an utterance endpoint, we believe a portion of this information can be captured in the form of grammatical structure.

Grammar groups words into word-classes that represent the word's functional role in the sentence. For example, verbs represent action in a sentence, and nouns can be objects which are acted upon, or perform the action. Similarly, determiners, prepositions and adverbs all have their distinct role in communication. These sequence of these grammatical roles may provide a general template for the structure of an utterance, which can be used to determine when utterances begin and end.

3.3 Finding domain specific endpointing cues using n-gram grammar

This section presents the analysis of n-gram modeling for part-of-speech word classes as a technique to capture regularities in the grammatical structure of an utterance. Based on these regularities, we hope to pinpoint reliable indicators of utterance endpoints for conversations in the HelpingHand corpus. This sort of analysis can be quite powerful, as the modeling is not constrained to just bigrams, and because multiple signals can be easily combined to evaluate utterance endpoint probabilities.

Standard n-gram modeling is a statistical method of predicting future observations and assumes a process exhibits the following property:

$$P(o[n]|o[n-1], o[n-2]...) \approx P(o[n]|o[n-1], o[n-2]...o[n-N]) \tag{3.1}$$

In most cases, we use this model to model word sequences by making an assumption that the last N observations exerts most of the influence in determining $o[n]$ and that observations $o[n-N]$ and before contribute only negligible amount in determining $o[n]$.

3.3.1 Procedure

In order to obtain the grammar classes of our words, we use a rule-based part of speech tagger [2]. A rule-based part of speech tagger uses a training corpus which consists of a collection of sentences with each word annotated with its correct parts of speech tag. A rule-based part of speech tagger creates rules according to this training set, and can be used subsequently to determine the part of speech of words in an untagged sentence. Using a rule-based tagger with lexical rules trained on the Brown corpus, we tag correctly segmented transcriptions of the HelpingHand corpus. We then obtain measurements for the following four parameters for a correctly segmented utterance $U = \{w_1...w_n\}$ where U is a substring of the entire stream of words representing a HelpingHand transcript, and UB signifies an utterance break at time n:

$$P(UB|WC(w_{n-1}))$$
$$P(UB|WC(w_{n-1}), W(w_{n-2}))$$
$$P(UB|WC(w_{n+1}))$$
$$P(UB|WC(w_{n+1}), WC(w_{n+2}))$$

In an most n-gram language models, only observations before time n can be used to predict observations at time n. This is because the tasks which utilize this model often use these n-gram models to predict the likelihood of the nth observation, before the system has access to observations in the future. In utterance segmentation however, we are not required to produce boundary decisions under such harsh time constraints. Therefore, it is advantageous to extend the n-gram model to look at words which come both before and after the nth observation in order to making boundary decisions. The last two measurements listed in the above table are a relaxation of the n-gram model which takes advantage of the weaker time restrictions of the utterance segmenter task.

3.3.2 Results

Despite the fact that the speakers in the HelpingHand corpus were given complete freedom in their method of interaction with the computer, there were significant grammatical patterns which arose from the data from the class based n-gram analysis. Below, we present the analysis, observations, and relevant data from the n-gram probability calculations. Complete results for each calculation can be obtained in the Appendix section.

The results of the unigram counts over the entire HelpingHand corpus already exhibit one advantage of word class based clustering. Since the HelpingHand corpus is based upon spontaneous speech interactions based on a largely unconstrained vocabulary set, many words appear relatively few times, making n-gram analysis on the words themselves difficult due to data sparseness. While the HelpingHand corpus contains 523 unique words [9], 193 of these words only occur one or two times [12]. Since the probability state-space grows exponentially with N, the number of observations used to calculate the n-gram, word-based n-gram analysis will quickly run into over fitting and sparseness issues. In contrast, grouping these words by word class has reduced the number of distinct classes to 27, which increases the density of the data, therefore increasing the chances of locating statistically significant correlations in the data.

The probability values obtained for $P(UB|WC(w_{n+1}))$ represent a modified n-gram technique examining the probability of a utterance endpoint based on the first observed word class after the utterance endpoint. The bigram model probabilities

Part of Speech	Count	$P(UB\|WC(w_{n+1}))$
CC	30	0.211268
CD	1	0.00684932
DT	35	0.0120151
IN	11	0.0066305
JJ	36	0.0144289
JJR	26	0.337662
JJS	1	0.0204082
MD	3	0.428571
NN	115	0.0429265
NNS	4	0.015873
RB	38	0.076
RBR	8	0.103896
TO	9	0.0144231
UH	4	1
VB	2693	0.901875
VBD	1	0.0588235
VBP	1	0.0833333
VBZ	4	0.0366972

Table 3.2: Unigram Part of Speech Probabilities

give results indicating that in over 90% of the cases, a verb word class signifies the beginning of a new utterance. The set of tags used to signify the part of speech of each word is the Penn Treebank Tag Set [8].

This metric exhibits the reliability of a verb word class signifying the beginning of an utterance, which, in retrospect is not suprising given the command oriented nature of the HelpingHand task. In order to evaluate the effectiveness of this criteria, we must obtain a second metric showing the percentage of utterance breaks which begin with a verb. This second metric shows us that out of 3020 utterances, 2693 of them or 89.2% began with a verb. This makes verbs a particularly powerful indicator of an utterance break especially given the freedom which users were given in interacting with the HelpingHand simulation.

The probability values for the trigram model for utterance beginnings,

$$P(UB|WC(w_{n+1}), WC(w_{n+2})) \tag{3.2}$$

verifies our bigram findings of the importance of verb classes in beginning an utterance. Almost all trigram pairs with a verb at the n+1 position have high associated probabilities in the mid-eighties in comparison to the other trigrams. Most notably, the *UtteranceBreak*/Verb/Determiner trigram cause by utterances of the form “move the..” is the most frequently occurring trigram, accounting for more than half of the utterance beginnings with a word length greater than one. Additional notable features include the *UtteranceBreak*/Noun/Verb combination which yields a 70% accuracy rate, occurring when utterances beginning with the form “ok, [move/place/get] the...” when the user acknowledges the computer’s actions, and gives it the next command. These utterance patterns are all well distributed across the majority of speakers. Although the trigram model has many other high probability trigrams which delineate the beginnings of an utterance and can be intuitively explained from a grammatical perspective, the strongest indicator which explains the majority of utterance breakpoints seems to have been captured by the *UtteranceBreak*/Verb bigram in the previous bigram model.

The probability values obtained for

$$P(UB|WC(w_{n-1})) \tag{3.3}$$

and

$$P(UB|WC(w_{n-1}), W(w_{n-2})) \tag{3.4}$$

represent the traditional n-gram modeling technique using one and two previous observed word classes in the transcript to predict an upcoming utterance break at time n. We readjust the probability values for the bigram model $P(UB|WC(w_{n-1}))$ to discount utterances which are one word long (such as “go” or “stop”) as these are overlap with a portion of the information calculated for $P(UB|WC(w_{n+1}))$ discussed above. The same is done to the trigram model $P(UB|WC(w_{n-1}), W(w_{n-2}))$ to avoid overlaps with the trigram analysis of the utterance beginnings.

While both results show reasonable density distributions, no particular bigram or trigram clearly distinguishes itself as a strong indicator of utterance segmentation as in the previous case. Some potential values which may be combined with other indications to yield stronger signals are

Part of Speech	Tag	Count	$P(UB \mid WC(w_{n-1}))$
Adverb/*UtteranceBreak*	RBR	38	0.493506
Noun/*UtteranceBreak*	NN	742	0.276969
Adjective/Adverb/*UtteranceBreak*	JJ/RBR/.	17	0.607143
Noun/Adverb/*UtteranceBreak*	NN/RBR/.	17	0.53125

Table 3.3: Potential candidate results from standard bigram/trigram analysis

Comparative Adverbs end sentences in cases where the user is tweaking a magnet position, such as “move the magnet higher”. Singular noun endings occur most frequently when users are specifying a magnet position relative to a landmarks, such as “place the magnet above the red ball”, and “move it above the square”. As these probability values indicate, however, these values in isolation do not convincingly indicate an utterance endpoint, although in conjunction with other signals, they may become more useful.

In the trigram model conditioned on the word classes of the last two words of an utterance, we also fail to find as strong a correlation between word classes of utterance endings and utterance endings. This suggests that even though no explicit restrictions were placed on the interaction, the task itself prompted speakers to adopt a similar linguistic style.

3.4 Incorporating Pause Duration as a Higher Level Cue

In this section, pause duration is examined as another high level cue aiding in grouping together segmented utterances produced by the signal level segmenter.

The analysis in the previous chapter extensively evaluated pause durations while optimizing the segmenter parameters for the signal level segmenter. By taking advantage of the strong temporal continuity of words associated with the same utterance, the signal level segmenter was able to cluster much of the speech into correct utterance groupings. In order to minimize two distinct utterances from mistakenly being concatenated together, we introduced a weighted scoring function which causes the segmenter to hedge its bets toward over-segmentation. In dealing with complex utterances where the user pauses mid-sentence in order to construct the remainder of his command, the signal level segmenter will conservatively group these two word clusters separately. Because of the over-segmenting tendency of the signal level segmenter, we hypothesize that there may still be a strong temporal correlation which can be used to distinguish segments from the segmenter which belong to the same utterance, from those which belong to distinct utterances.

3.4.1 Data Preparation

In order to evaluate pause duration between utterance segments and examine the correlation between pause and utterance endpoints, we must align utterance segments from the segmenter with the hand segmentations and use this alignment to obtain pause duration distributions of utterance endpoints, and mistaken segmentations.

We combine three information sources to obtain the measurements. A file representing word level segmentations as obtained by the segmenter, a file representing the ideal word level segmentations obtained by referring to the hand segmentations from the previous chapter, and a word level transcription file used to derive pause length between utterance segments.

In order to obtain the first two files, the same five sessions of the helpinghand corpus from the previous analysis are segmented into short raw audio files as determined by the signal level segmenter. These raw audio files for each session are hand transcribed, and the file level transcriptions are force-aligned to the audio. We cannot directly force align an entire session to its transcription since small errors propagate to throw off the alignments for long lengths of audio. These force alignments are then mapped back into the entire session recording to obtain word transcriptions aligned to the audio for the entire session. As described above, this session level transcription is used to determine pause length between utterance segments.

We then refer back to the hand segmentations from the previous chapter to obtain correct unaligned word level transcriptions for the correct segmentations, and combine these three files to obtain the pause length following each utterance segmentation.

3.4.2 Data and Analysis

Utterance Endpoint Segments		
Pause (secs)	No. Correct	No. Incorrect
1	10	42
1.5	12	58
2	12	24
2.5	20	19
3	7	13
5	127	19
10	498	7
10+	464	1

Table 3.4: Pause duration histograms for Utterance Endpoints and Mid-Utterance Segments.

The data from Table 3.4 contains two histograms classifying pause into one of ten bins. The numbers in column two and three represent the frequency that segments in each category contain a subsequent pause falling in the specified range. The second column contains information on segments whose endings correspond to utterance endpoints, and the third column contains information on segments whose endings are not utterance endpoints. These distributions exhibit a clear trend showing that pause durations following utterance endpoints are much longer than pause durations in the middle of an utterance. This data suggests that the longer the system waits for the next utterance segment, the more confident it can be that the last segment corresponds to an utterance segment. Such information can be used to determine an upper bound on pause length to determine utterance endings. The same probabilities are shown in Bayesian form in Table 3.5.

This data suggests that the longer the system waits for the next utterance segment, the more confident it can be that the last segment corresponds to an utterance

$P(UB\|\ t = 1)$	0.192308
$P(UB\|\ t = 1.5)$	0.171429
$P(UB\|\ t = 2)$	0.333333
$P(UB\|\ t = 2.5)$	0.512821
$P(UB\|\ t = 3)$	0.35
$P(UB\|\ t = 5)$	0.869863
$P(UB\|\ t = 10)$	0.986139
$P(UB\|\ t > 10)$	0.997849

Table 3.5: Probability of an utterance endpoint conditioned on pause duration

segment. Using this information, we can set an upper bound on the waiting time for the next utterance segment. If the segmenter does not send another utterance within this time frame, it can conclude that that an utterance has ended.

3.4.3 Discussion

In the previous section, we performed a traditional n-gram analysis using past word observations to predict an upcoming utterance endpoint. The resulting probabilities however, did not show any strong terminating word classes which could be used to predict utterance endpoints. This can partially be attributed to the fact that there are a notable number of cases where false starts occur in the HelpingHand corpus. In the particular grammatical cues using only past observations would not predict the occurrence of an utterance endpoint in cases where the speaker trails off to reformulate a statement after beginning the first few words. A upper bound on the pause length would allow the system to predict endings that may be missed with traditional n-gram methods.

3.5 Combining Multiple Higher Level Cues

The discussion in the previous section naturally leads us to a final method of utterance endpoint detection combining pause and linguistic structure. In the first analysis using a modified n-gram language model based on word classes, the verb part of speech distinguished itself as a strong indicator for utterance beginnings. Independently, the pause analysis in the second section presented a method of analysis to obtain an upper bound beyond which the system could confidently conclude that an utterance endpoint has occurred.

Aside from these two strong indicators, there are several word class indicators which are significant, but not strong enough to be used to declare an utterance endpoint. In the previous section, we used pause duration to find a hard upper bound for utterance endpointing. The data in Table 3.4 and Table 3.5 indicate that this boundary is not discrete, but a gradient where the confidence of a breakpoint increases as the pause length increases. In this section, we attempt to combine both elements to use a combination of signals to strengthen our decision process on utterance endpointing.

3.5.1 Data and Analysis

The rule based part of speech tagger used for earlier sessions is used again to tag the utterance level transcripts and combine grammatical information with the pause data obtained in the same manner described above.

We obtain the results for

$$\begin{array}{lll} P(UB|\ WC(w_{n-1}), t \in \beta_i) \quad \text{where} \quad t \in & \beta_0 = \text{if } 0 \leq t < 1.5 \\ P(UB|\ WC(w_{n+1}), t \in \beta_i) & \beta_1 = \text{if } 1.5 < t \leq 3 \\ & \beta_2 = \text{if } 3 \leq t \leq \infty \end{array}$$

Where t_u is the pause length after the last word of the previous utterance. The ranges for β_i are hand chosen based on the pause distribution histogram in Table 3.4 such that the first and last bins primarily fall in the pause range of incorrectly or correctly segmented utterances, and the middle bin contains a mix of both types of boundaries. Trigrams in conjunction with pause are not calculated as our limited data set encounters sparseness issues.

Table 3.6 is the output representing pause used in conjunction with the word class of the last word before the pause. The middle column represents the total number of examples of which were encountered, and and the right column represents the

TAG	β_i	COUNT	$P(UB \mid WC(LAST), \beta_i)$
DT	β_0	9	0.111111
DT	β_1	7	0
DT	β_2	7	0.857143
IN	β_0	17	0.0588235
IN	β_1	7	0.142857
IN	β_2	56	0.964286
JJ	β_0	15	0.0666667
JJ	β_2	109	0.990826
NN	β_0	50	0.06
NN	β_1	29	0.206897
NN	β_2	294	0.945578
NNS	β_2	22	0.954545
RB	β_2	48	1
RBR	β_2	23	1
VB	β_0	23	0.695652
VB	β_1	34	0.794118
VB	β_2	540	0.998148

Table 3.6: Probability of an endpoint given the last word class and pause duration

probability that these indicators signal an utterance break. We filter out entries with less than 5 samples, as the sample set is too small to draw accurate conclusions.

The trends for each bin do reinforce our previous analysis of pause duration. Independent of word class, β_2 generally has the highest probability of being an utterance breakpoint followed by β_1. A good example of this the (VB,β_2) parameter, which shows that almost all verbs preceding a significant duration of silence are sole utterances. This connects with our qualitative analysis of the HelpingHand corpus, which show many single word verb commands such as “execute” and “stop”. The additional insight that we obtain from the incorporating the pause duration is that, as we hypothesized, shorter segments–although still high–have a lower probability of being an utterance endpoint.

A good example illustrating the power of our combined analysis of high level cues are the noun (NN) endings in combination with pause duration. Recall that in normal bigram and trigram analysis earlier in the chapter Noun-class endings provided only a 27.8% certainty of an utterance endpoint (Table 3.3). By incorporating pause duration however, we see that when pause lengths are over three seconds long, there is a 94.6% chance of an utterance segment whereas if it is less than 1.5 seconds there is only a 0.06% chance of an utterance ending. Since multiple nouns are relatively frequent in utterances in the HelpingHand corpus it is not surprising that the presence of a noun is not a strong indicator of an utterance break. This suggests that many noun ending segments which are not utterance endpoints

"move the left magnet....to the left just a tiny bit"

can be accurately distinguished from

"move the right magnet"

just by waiting a few seconds. By combining both pause and word class information, we are able to discover a strong indicator out of two previously weak signals for utterance endpointing.

TAG	Pause Bin	COUNT	$P(UB\mid WC(FRONT), \beta_i)$
CC	β_0	17	0.588235
CD	β_0	6	0
DT	β_0	41	0.146341
IN	β_0	59	0.0338983
JJ	β_0	16	0.3125
NN	β_0	59	0.864407
NN	β_2	6	1
RB	β_0	33	0.30303
TO	β_0	22	0.0909091
VB	β_0	508	0.988189
VB	β_1	30	0.966667
VB	β_2	512	1

Table 3.7: Probability of an endpoint given the first word class of the utterance and pause duration

Table 3.7 shows the output representing pause used in conjunction with the word class of the last word before the pause. The format is identical to Table 3.6 Again, we see the same trends as in the pause only indicator, showing that independent of the word class beginning the next utterance, an increase in pause duration increases the probability of an utterance endpoint with one exception. The strong verb correspondence which was identified in the first section is independent of pause duration between segments, and reliably signals an utterance endpoint for all pause lengths.

Finally, we obtain new scores based on these combined grammatical indicators listed in Table 3.6 and Table 3.7 to augment the segmentation performance on the five sessions of the HelpingHand corpus examined in Chapter Two. The entries in the two tables are triggered when it observes an occurrence of the rule, and the probability of the rule is higher than an activation threshold K, or lower than $1 - K$, where $0 < K < 0.5$. Rules with probabilities higher than 0.5 are considered "segmenting rules", or rules which indicate the next speech section should not be combined with the previous one. Rules with probabilities lower than 0.5 are considered "combining rules", or rules which indicate the next speech section should be combined with the previous one. At the end of each speech section determined by the signal level segmenter, we tally up the votes from "segmenting rules" and "combining rules" to decide whether to segment or combine two speech segments. Ties are broken by taking the side with the higher probability, and nothing is done when zero rules are triggered.

K	$\Delta falsepositives$	$\Delta falsenegatives$	$\Delta totalerror$
0.6	-266	14	-252
0.7	-270	12	-258
0.8	-268	8	-260
0.9	-226	6	-220
1.0	-26	0	-26

Table 3.8: Error Count Changes Due to Grammar and Pause Rules

Table 3.8 illustrate the improvement in error as the activation threshold K is varies from .6 to 1. We see a steady improvement in false negative errors as our activation threshold K becomes more stringent. At the same time, as K increases, we also observe a decrease in the improvement of false positives error bounds due to the fact that less rules are triggered. This is the same sort of trade off encountered in Chapter Two, where we weighted the value of an additional false negative error in comparison to one less false positive error. Our total performance improvement is summarized in Table 3.9, which indicates a 10.7% improvement in segmentation performance based on the higher level cues for $K = 0.8$.

	orig	opt	opt+enrgy	wtd opt+enrgy	everything
False Negative Boundaries	682	1370	1371	1052	1060
False Positive Boundaries	1976	1050	1047	1377	1109
Total Boundary Error	2658	2420	2418	2429	2169

Table 3.9: Error Comparison Table for Higher Level Cues

3.6 Conclusion

In this section we examined the incorporation of higher level cues into the utterance segmentation task. We hypothesized that higher level indicators in the form of linguistic information and pause duration could aid in distinguishing utterance endpoints and segments which are part of a longer utterance. Although humans use semantics to accurately determine true utterance segments, we hypothesized that there were a variety of other cues which also provide strong indication of utterance endpoints.

Modified n-gram based language models were used based on the argument that syntax carries a large amount of information regarding the semantics. We generalized the syntactic n-gram model a step further to show that the grammatical role of the words can be effective in capturing the semantics of statements through the functional roles of the words. By using n-gram analysis to search for regularities in these functional roles, we were able to isolate a word class which provided a strong indication of an utterance endpoint.

Despite the fact that the speakers in the HelpingHand corpus were given no restrictions on how they verbally interacted with the simulation, it is interesting to see that they all automatically fell into similar patterns of speech. It is likely that such similarities arose because the speakers adapted their speech and speaking style (within the grammatical constraints of the English language) to a format which they determined was optimal for communicating with the computer and for the particular task of directing it to perform tasks. As one of these features, speakers consistently placed the verb at the start of the utterance.

We hypothesize that the command-oriented nature of the HelpingHand task prompted all speakers to use a discourse style which focused on to putting the most functionally significant part of speech at the front of the utterance to insure that the meaning was best communicated. As a result, we were able take advantage of grammatical structure to identify utterance endpoints. Indeed, the traditional usage of active and passive voice seem follow the same trend. It is often noted that active voice is always preferable to passive voice because it makes the sentence sound stronger. One reason the sentence seems stronger may be that in active voice, the location of the words more central to the meaning of the sentence are placed closer to the front. Dialog cues such as these are effectively captured using our grammar based n-gram analysis.

In summary, word class based n-grams are shown to be effective in capturing statistically significant relationships between the grammar classes and utterance endpoints. Pause duration analysis is another indicator which performed well at the higher level. Here we found that pause duration was positively correlated with the probability of an utterance endpoint. Lastly, these two variables were crossed to yield several new endpoint indicators which, independently, were too weak to be used for endpoint detection. By incorporating these higher level cues into the segmentation task, we observed a 10.7% improvement in segmentation accuracy.

Chapter 4

Conclusion

As speech recognition systems become more common in conversational settings, it must tackle the associated challenges of spontaneous speech. In particular, systems must be able to accurately identify when a user is done speaking. Utterance endpointing in spontaneous speech is challenging because users often speak in an irregular manner, pausing mid-sentence, or failing to pause between successive commands. This presents problems for simple pause duration metrics used for utterance endpointing in many recognition systems.

This thesis presented a combined high level and low level approach to utterance endpoint detection in spontaneous speech. The first half of this research describes the development of a procedure to set optimal parameters for the signal level utterance segmenter, and the second half of this research outlines the use of grammar and pause duration as higher level cues to aid in utterance endpoint detection.

In the first section, we obtain a range of optimal parameters for the signal level segmenter, and show an 8.6% improvement in boundary detection from the original, hand set values to the final parameters. The experiments were done over the Helping-Hand corpus, and the final optimal was obtained by adding energy filtering and using a weighted scoring function. The obtained set of optimal values lets us easily adjust the behavior of the segmenter depending on: (1) the nature of the application and (2) the accuracy of additional utterance endpointing systems at higher levels. Although the missed boundaries count is higher for our new optimal values, these new values reflect the weighting chosen in Section 2.11, which weights our reduction in overall error slightly higher than for our original parameters.

Future work for the signal level utterance segmenter involves increasing the accuracy of the segmenter by incorporating a three phone model to do frame level classification, as outlined in Appendix A. Our initial experiments on the three phone model show a 2.94% improvement in classification accuracy for speech frames, suggesting that the incorporation of a three phone model is a promising strategy to improve the accuracy of the signal level utterance segmenter.

In the second section, we examined the effectiveness of higher level cues, pause duration and grammar sequence, in the utterance endpointing task. For the HelpingHand corpus, we found that although there were no restrictions placed on the users' interaction with the computer, the task was constrained enough that regular grammatical patterns emerged from the dialogue. The advantages to grammatical analysis, compared to normal n-grams is that it is class based, minimizing the observation set and eliminating many sparseness problems encountered in word based n-gram modeling. Additionally, as long as the part of speech is known, grammar based n-gram analysis allows us to make accurate endpointing predictions even if the word has not been encountered in the training phase. The results of our class-based n-gram analysis showed that Verbs were a strong indicator of utterance startpoints, with over 90% accuracy in predicting utterance startings. Next, we performed an analysis of using pause duration to set an upper threshold in clustering segments coming from the signal level segmenter. Our results showed that the probability of two segments belonging to the same utterance decreased as the time interval between them grew longer. We successfully combined pause and grammar to yield several new parameters, such as the Noun word class, which, in conjunction with pause duration became a reliable indicator for an utterance endpoint. Finally, by incorporating these higher level cues into the segmentation task, we obtained an additional 10.7% improvement in segmentation accuracy.

Future work in this section involves the exploration of a variety of more complex modelling methods, including a variation of the Hidden Markov Model to predict clusterings of groups of utterances. In such a model, transition weights would be trained to the transcription data from application domain, and the state distribution probabilities–corresponding to part of speech classes–would be trained on a larger, text based corpus. Additionally, it may be useful to explore the effectiveness of additional higher level cues to facilitate in utterance endpointing. Finally, the high level indicators which we have identified as useful predictors for utterance endpoints can be incorporated to assist in utterance endpointing in real time conversational systems.

Appendix A

An Alternative Frame Level Classifier using a Three Phone Model

This section explores the implementation and design of a model intended to obviate this sort of workaround by reconstructing an utterance segmentation model to more correctly recognize unvoiced sounds as speech. A new utterance segmentation model is modified to include three instead of two phone classes. Previously, we had non-silence and silence phone models. The speech model was trained on all phones in the speech stream, and silence model was trained on the silences. This study further subdivides the non-silence model into two models, unvoiced and voiced phones. Voiced phones are defined as sounds where the vocal tract vibrates, and unvoiced sounds are defined as those where the vocal tract do not. The TIMIT database is used to train these new phone classes by mapping voiced and unvoiced phonemes. The back of Appendix A lists the phone mappings between the three states and the TIMIT phones. We extract the uniphone audio frames from the TIMIT database and create a uniphone Gaussian mixture model (GMM) [6] of a phone consisting of 12 Gaussian and the 39 features outlined above.

One Gaussian of 39 features is defined as follows:

$$\theta(\chi) = \frac{1}{\mathcal{S}\sqrt{2\pi}} e^{-(\chi-\mathcal{M})^2/2\mathcal{S}^2} \tag{A.1}$$

where

$$\mathcal{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_i \end{bmatrix} \quad \mathcal{M} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_i \end{bmatrix} \quad \mathcal{S} = \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_i \end{bmatrix} \quad 1 \leq i \leq 39 \tag{A.2}$$

Twelve such Gaussians are combined into a GMM to simulate the feature distribution of voiced, unvoiced, and silence phones such that the probability of a particular observation x arising from the particular distribution, can be approximated by the sum probabilities of the 12 Gaussians in the following manner.

$$p(x = silence) = \sum_{i=0}^{M} \alpha_i p_i(x|\theta_i) \tag{A.3}$$

Where θ_i represents the ith Gaussian distribution involved in modeling the particular phone, and $\sum_{i=0}^{M} \alpha_i = 1$. The Expectation Maximization (EM) [1] algorithm for finding maximum likelihood mixture densities parameters are used over some iterations to find numbers for $\mathcal{M}$ and $\mathcal{S}$ which will do a decent job of approximation.

Unfortunately, it turns out that our sample size from the TIMIT database, particularly for voiced phones exceeds the amount of memory needed to process the data. As a result, we break the sample into 10 groups of about 15,000 samples each, and we average the operation to utilize the entire dataset.

A first attempt was made by averaging the Gaussian distributions of the 10 datasets so that we would obtain a new phone model with 12 Gaussians representing the average Gaussian of each of the 10 models. the parameters M', and S' for each of the 12 Gaussians in the average phone model Θ' are obtained as follows.

$$\mathcal{M}'_i = \sum_{j=0}^{N} \alpha_{ji} \frac{\mathcal{M}_{ji}}{N} \tag{A.4}$$

Where j refers to the jth training subgroup, and i refers to the ith Gaussian mixture, and α_{ji} is the weighting of the ith mixture in the jth subgroup. A parallel procedure is done to obtain values for $\mathcal{S}'_i$.

Based on the distribution within each subgroup however, it was subsequently discovered that this sort of averaging procedure may not give accurate results as the EM algorithm fits the 12 Gaussians relative to the data in each subgroup. Therefore unless the underlying distribution of each of the subgroups is roughly similar to the ideal distribution, our averaging procedure is not guaranteed to obtain favorable results. An alternative linear combination method is shown below.

$$p_{avg}(x) = \sum_{j=0}^{N} \frac{\beta_j}{\sum_{j=0}^{N} \beta_j} p_j(x) \tag{A.5}$$

Here $p(x)$ represents the probability that a frame vector x is a phone that came out of this phone model and β_j represents the number of samples used to generate the GMM for subgroup j. We are taking each of the Gaussian mixtures trained by the subgroups and linearly combining the models by weighting the probabilities returned by those subgroups by the fraction of the data used to train the mixture j. Since the fraction term $\frac{\beta_j}{\sum_{j=0}^{N} \beta_j}$ sums to 1, our new probability density $p_{avg}(x)$ remains valid A-1.

A.0.1 Results

Several tests are performed on the TIMIT corpus to see how well the three state model does in comparison to the two state model. Our results are expressed in a confusion matrix as follows:

The Matrix A.0.1 represents the original speech silence model classifying non-silence and silence frames in the TIMIT corpus, and Matrix A.0.1 represents the voiced-unvoiced-silence model classifying frames in the TIMIT corpus. The rows

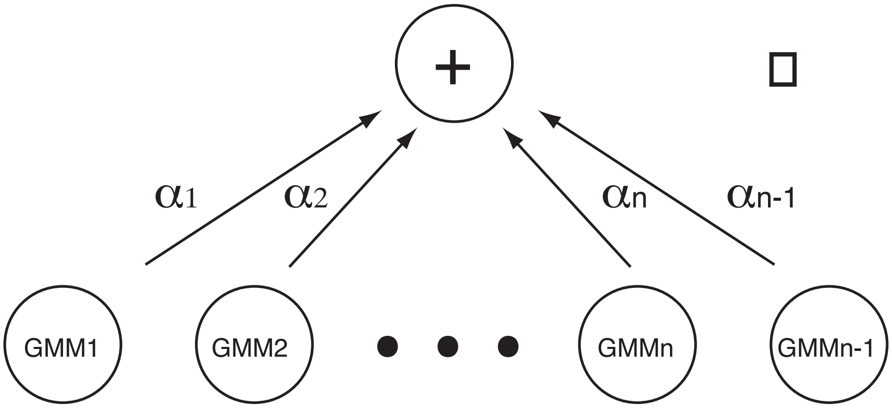


Figure A-1: Weighted recombination of GMM models

$$\begin{array}{r|rr|} & \textit{sil} & \textit{voiced} \\ \textit{silence} & 349748 & 277305 \\ \textit{voiced} & 146803 & 1154776 \end{array}$$

Table A.1: Classification Matrix for the Original Two State Segmenter.

$$\begin{array}{r|rrr|} & \textit{silence} & \textit{voiced} & \textit{unvoiced} \\ \textit{silence} & 332378 & 187044 & 40084 \\ \textit{voiced} & 61648 & 833784 & 62349 \\ \textit{unvoiced} & 102525 & 115961 & 192859 \end{array}$$

Table A.2: Classification Matrix for the Three Phone Segmenter.

represent the model's hypothesis, and the columns represent the correct classification based on the real solution obtained by mapping the phonetic transcript of the sentences into the corresponding phone models. The number of correctly classified frames lie on the diagonal of the matrix, and the off diagonals represent classification errors. (For example, Matrix A.0.1 shows us that 187044 voiced frames were mistakenly classified as silence.)

There is an improvement in the number of voiced/unvoiced segments recognized accurately, up by 50177 samples compared to the first model, but there is a corresponding decrease in recognition accuracy of silence segments, down by 17370 samples. This is an increase in recognition accuracy of about 3.5% in voiced frames, but a decrease in recognition accuracy of unvoiced frames by about 3.49%

Frame Realignment

We also present another optimization where we realign the frames to begin and end in the middle of the window instead of the beginning of the window. With a 20ms window and a 10ms step-size, the acoustic features which were extracted were being mapped to the beginning of the 20ms window instead of being centered on the window. Figure A-2 illustrates the modification.

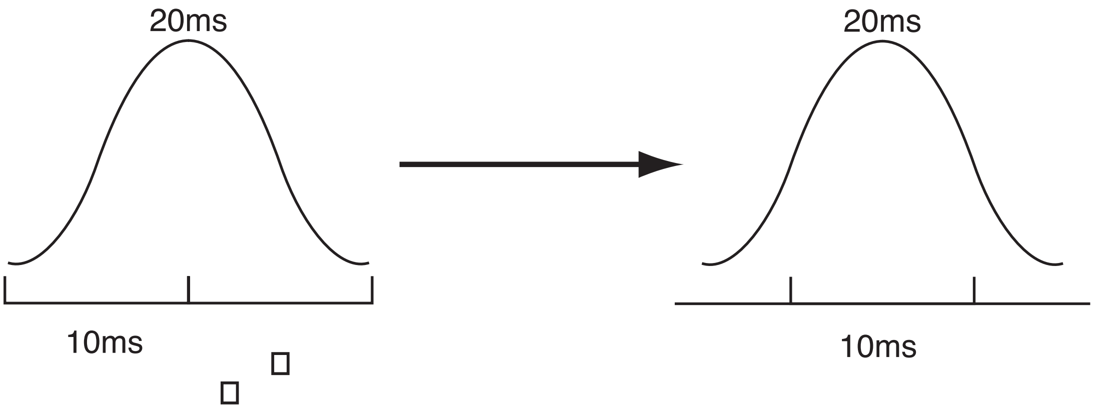


Figure A-2: Frame-centering performed to maximize Hamming window area under the current frame.

This alignment correction is made, and the resulting improvements in recognition accuracy are shown in the confusion matrix below.

$$\begin{array}{rc} & \begin{array}{cc} \quad \textit{sil} & \quad \textit{voiced} \end{array} \\ \begin{array}{r} \textit{silence} \\ \textit{voiced} \end{array} & \begin{bmatrix} 365018 & 262035 \\ 131678 & 1169901 \end{bmatrix} \end{array}$$

Table A.3: Classification Matrix for realigned Two Phone Classifier

$$\begin{array}{rc} & \begin{array}{ccc} \textit{silence} & \textit{voiced} & \textit{unvoiced} \end{array} \\ \begin{array}{r} \textit{silence} \\ \textit{voiced} \\ \textit{unvoiced} \end{array} & \begin{bmatrix} 347895 & 173654 & 37957 \\ 54604 & 851856 & 51321 \\ 94197 & 110972 & 206176 \end{bmatrix} \end{array}$$

Table A.4: Classification Matrix for Realigned Three Phone Model

For the two phone model, this change improves our frame based recognition accuracy for silence and non-silence frames by 3.1% and 1% respectively. In the three phone model, our changes improve the recognition by 3.1% and 2.2%.

We finalize our results by iterating the BaumWelch algorithm for our 3-state models until convergence, yielding tighter bounds and better results shown in Table A.0.1

$$\begin{array}{rc} & \begin{array}{ccc} \textit{silence} & \textit{voiced} & \textit{unvoiced} \end{array} \\ \begin{array}{r} \textit{silence} \\ \textit{voiced} \\ \textit{unvoiced} \end{array} & \begin{bmatrix} 346902 & 170865 & 37355 \\ 55499 & 854068 & 51711 \\ 94295 & 111549 & 206388 \end{bmatrix} \end{array}$$

Table A.5: Final Results

This section outlined a more accurate frame classifier for the signal level utterance segmenter, which uses three instead of two phone classes to determine speech and non-speech frames. In conjunction with the frame realignment modification, the results show a 2.94% improvement in speech frame recognition but a 0.814% decrease in the in the classification accuracy of silence at the frame level. This decrease in classification in silence segments can be attributed to some silence segments being misclassified as unvoiced, and therefore speech frames. Overall however, these percentages show an improvement in frame classification accuracy and can potentially translate to an improvement at the segment level for the signal level utterance segmenter. Future work would include retraining the GMM models with a more efficient algorithm in order to increase classification speed to real-time. This three state model can then be incorporated to improve the performance of the signal level segmenter.

A.1 Phone mappings for the Three Phone Model

The tables indicate the phonetic mapping between the three phone frame classifier and the set of phones used in the TIMIT corpus.

Silence Phones
sp
sil

Unvoiced Phones
hh
k
p
t
th
f
s
sh
ch

Voiced Phones	
aa	ih
ae	iy
ah	jh
ao	l
aw	m
ay	n
b	ng
d	ow
dh	oy
eh	r
er	uh
ey	uw
g	v
w	z
zh	

Appendix B

Grammar Evaluation Survey

Directions: The following is an unformatted transcription of a computer user instructing the computer to perform a series of tasks. Each numbered line represents what the computer thinks is a single sentence/command. Help the computer out by correcting the computer's mistakes. Separate lines that have more than one phrase/command on it, and combine lines which together represent one command. Use / to separate the commands. Once the computer knows where the phrase ends and starts are, it can interpret the phrase without a problem. It just needs help finding the starts and ends of the user's thoughts. This task is very subjective and there is no real "right" answer. Just do the best you can, and mark the boundaries in the way you think best separates the speech.

Example:

Original	Corrected
1. HELLO	1. HELLO /
2. PLEASE GET ME	2. PLEASE GET ME
3. THE RED BALL	3. THE RED BALL /
4. OK	4. OK /
5. GO	5. GO /
6. STOP MOVE THE BALL TO THE RIGHT	6. STOP / MOVE THE BALL TO THE RIGHT /
7. OK	7. OK /
8. END	8. END /

In the above case, lines 2 and 3 are put together since combined, they are one command. The / at the end of line 1 signals the beginning of the phrase PLEASE GET ME THE RED BALL as well as the end of the phrase HELLO. The / at the end of line 3 signifies the end of the phrase spanning lines 2 and 3. On line 6, we have two phrases which were accidentally combined into one line. The / between STOP and MOVE provides the appropriate break to segment the line into two phrases.

1. OK CAN I DO THE CAN I DO IT MULTIPLE TIMES OR WITH THE ONE GAME
2. CLICK ON GO OR I SAY GO
3. OK
4. AND I CAN SAY GO OR EXECUTE AND THAT WILL WORK
5. OK
6. CAN I JUST START TELLING IT
7. CAN I TELL IT TO SAY PLEASE IS THAT OK OK PLEASE PLACE A MAGNET
8. UNDER

9. THE SMALL BLUE BALL
10. PLEASE MOVE THE MAGNET UNDER THE SMALL BLUE BALL SLIGHTLY TO THE LEFT
11. PLEASE MOVE THE MAGNET UNDER THE SMALL BLUE BALL
12. SLIGHTLY TO THE RIGHT BUT NOT COMPLETELY UNDERNEATH
13. THE SMALL BLUE BALL
14. PLEASE PLACE
15. A NEW MAGNET
16. ON THE RIGHT OF THE SMALL RED BALL
17. PLEASE
18. PLACE
19. A MAGNET DIRECTLY ABOVE THE LARGE RED BALL
20. AND SLIGHTLY ABOVE THE LEVEL OF THE SMALL RED BALL
21. PLEASE
22. MOVE THAT MAGNET SO +HAT IT IS
23. AT THE SAME HEIGHT BUT DIRECTLY ABOVE THE LARGE RED BALL
24. PLEASE MOVE THE THAT MAGNET TO THE LEFT
25. PLEASE MOVE THAT MAGNET TO THE LEFT AGAIN
26. PLEASE MOVE THAT MAGNET UP SLIGHTLY
27. PLEASE MOVE THE MAGNET THAT IS CURRENTLY BELOW THE SMALL BLUE BALL
28. SLIGHTLY ABOVE THE SMALL BLUE BALL
29. PLEASE MOVE THAT MAGNET A VERY SMALL DISTANCE TO THE LEFT
30. PLEASE MOVE THE MAGNET
31. ABOVE THE LARGE RED BALL
32. SMALL A VERY SMALL DISTANCE TO THE LEFT
33. OK
34. I WAS JUST TRYING TO GET IT SET UP
35. PLEASE MOVE THAT MAGNET BACK TO THE RIGHT
36. GO
37. STOP
38. PLEASE MOVE THE MAGNET ABOVE THE SMALL BLUE BALL TO THE RIGHT
39. PLEASE MOVE
40. THE MAGNET
41. ABOVE THE SMALL BLUE BALL
42. SO THAT IT IS UNDER THE SMALL BLUE BALL
43. PLEASE MOVE THAT MAGNET BACK UP SO THAT IT IS DIRECTLY UNDER THE SMALL BLUE BALL
44. PLEASE MOVE THE MAGNET AT THE TOP
45. OF THE SCREEN
46. SO THAT IT IS
47. ABOVE THE SMALL BLUE BALL
48. PLEASE MOVE THE MAGNET AT THE TOP OF THE SCREEN DOWN SLIGHTLY
49. PLEASE MOVE THAT MAGNET DOWN SLIGHTLY MORE
50. PLEASE MOVE THAT MAGNET DOWN JUST A SMALL BIT MORE
51. PLEASE MOVE THAT MAGNET SLIGHTLY TO THE RIGHT
52. PLEASE MOVE THAT MAGNET SLIGHTLY TO THE RIGHT AGAIN

53. GO

54. STOP

55. PLEASE MOVE THE MAGNET NEXT TO THE SMALL RED BALL DOWN

56. PLEASE MOVE THE MAGNET PLEASE MOVE THE MAGNET THAT IS ABOVE

57. THE LARGE BLUE BALL SLIGHTLY TO THE RIGHT

58. PLEASE MOVE IT BACK VERY VERY SLIGHTLY TO THE LEFT

59. PLEASE MOVE IT AGAIN VERY VERY SLIGHTLY TO THE LEFT

60. AGAIN VERY VERY SLIGHTLY TO THE LEFT

61. PLEASE MOVE THE MAGNET

62. THAT IS NEAREST TO

63. THE SMALL RED BALL TO THE RIGHT

64. PLEASE MOVE THAT MAGNET UP SLIGHTLY

65. PLEASE MOVE THAT MAGNET SLIGHTLY TO THE LEFT

66. GO

67. STOP

68. PLEASE MOVE THE MAGNET

69. ABOVE THE LARGE BLUE BALL

70. SO THAT IT IS ABOVE THE LARGE RED BALL

71. PLEASE MOVE THAT MAGNET UP

72. PLEASE MOVE THAT MAGNET DOWN VERY SLIGHTLY

73. PLEASE MOVE THAT MAGNET UP

74. EVEN MORE SLIGHTLY

75. PLEASE MOVE THAT MAGNET DOWN AND TO THE LEFT

76. GO

77. STOP

78. PLEASE MOVE THE MAGNET NEXT TO THE SMALL RED BALL

79. SO THAT IT IS ABOVE THE LARGE BLUE BALL

80. PLEASE MOVE THE MAGNET THAT IS NEXT TO THE SMALL RED BALL SO THAT IT IS ABOVE THE LARGE BLUE BALL

81. DIRECTLY ABOVE THE LARGE BLUE BALL

82. PLEASE MOVE THAT MAGNET UP VERY VERY SLIGHTLY

83. PLEASE MOVE THE MAGNET UNDER THE SMALL BLUE BALL SO THAT IT IS DIRECTLY TO THE LEFT OF THE SMALL BLUE BALL

84. PLEASE MOVE THAT MAGNET BACK TO WHERE IT WAS BEFORE

85. GO

86. STOP

87. PLEASE MOVE THE MAGNET ABOVE THE SMALL ABOVE THE LARGE BLUE BALL UP SLIGHTLY

88. GO

89. STOP

90. STOP

91. STOP

92. PLEASE MOVE

93. THE MAGNET ABOVE THE LARGE BLUE BALL UP SLIGHTLY

94. GO

95. STOP

96. PLEASE MOVE THE MAGNET ABOVE THE SMALL BLUE BALL UP SLIGHTLY

97. PLEASE MOVE THE MAGNET ABOVE THE LARGE BLUE BALL UP SLIGHTLY

98. GO

99. STOP

100. STOP

101. PLEASE MOVE THE MAGNET

102. UNDER THE SMALL BLUE BALL SO THAT IT IS IN THE SAME POSITION BUT ABOVE THE SMALL BLUE BALL

103. GO

104. STOP

105. PLEASE MOVE THE MAGNET

106. THAT IT IS ABOVE THE SMALL BLUE BALL

107. SO THAT IT IS

108. NEXT TO THE SMALL BLUE BALL

109. ON THE LEFT

110. GO

111. STOP

112. PLEASE MOVE THE MAGNET NEXT TO THE SMALL RED BALL SO THAT IT IS AS CLOSE AS POSSIBLE TO THE SMALL RED BALL

113. PLEASE MOVE THE MAGNET

114. PLEASE MOVE THE LEFT MOST MAGNET SLIGHTLY TO THE LEFT

115. GO

116. GO

117. STOP

118. GO

119. PLEASE MOVE ON TO THE NEXT PUZZLE

120. PLEASE MOVE ON TO THE NEXT PUZZLE

121. PLEASE PLACE A MAGNET

122. DIRECTLY UNDER

123. THE VERTICAL BLOCKS THAT DO NOT HAVE A HORIZONTAL BOTTOM

124. PLEASE MOVE THIS MAGNET DOWN

125. PLEASE MOVE THIS MAGNET UP SLIGHTLY

126. PLEASE PLACE A MAGNET AS FAR AS POSSIBLE TO THE LEFT OF THE SCREEN BETWEEN THE BLUE BALL AND THE CURRENT MAGNET

127. PLEASE MOVE THE MAGNET AT THE BOTTOM OF THE SCREEN TO THE RIGHT

128. PLEASE MOVE THE MAGNET TO THE LEFT OF THE SCREEN DOWN SLIGHTLY

129. PLEASE MOVE THE MAGNET TO THE LEFT OF THE SCREEN DOWN SLIGHTLY

130. PLEASE MOVE THE MAGNET TO THE LEFT OF THE SCREEN DOWN SLIGHTLY AGAIN

131. PLEASE PLACE A MAGNET ON EQUAL LEVEL WITH THE BLUE BALL

132. BUT ON THE OTHER SIDE OF THE GRAY BLOCKS NEAREST TO THE BLUE BALL

133. PLEASE MOVE THAT MAGNET SO THAT IT IS ON TOP

134. OF THE GRAY BLOCKS WHICH SURROUND THE BLUE BALL

135. PLEASE MOVE THAT MAGNET TO THE RIGHT

136. SLIGHTLY

137. GO

138. STOP

139. PLEASE MOVE THE MAGNET ON THE LEFT OF THE SCREEN UP

140. PLEASE THE MAGNET ON THE LEFT OF THE SCREEN UP AGAIN

141. GO

142. PLEASE MOVE THE MAGNET ON TOP OF THE GRAY BLOCKS TO THE FARTHEST POSSIBLE LEFT OF THE SCREEN

143. PLEASE TAKE THE MAGNET THAT IS CLOSEST

144. UNDER THE BLUE BALL

145. AND PLACE IT

146. UNDER

147. THE TARGET

148. AND UNDER THE GRAY BLOCKS UNDER THE TARGET

149. MOVE THAT MAGNET UP CLOSER TO THE GRAY BLOCKS

150. MOVE THE MAGNET AT THE BOTTOM OF THE SCREEN VERY VERY SLIGHTLY TO THE LEFT

151. MOVE IT AGAIN VERY VERY SLIGHTLY TO THE LEFT

152. GO

153. MOVE THE MAGNET AT THE BOTTOM OF THE SCREEN TO THE LEFT

154. STOP

155. MOVE THE MAGNET AT THE BOTTOM OF THE SCREEN TO THE LEFT AGAIN

156. MOVE THE MAGNET

157. CURRENTLY UNDER THE TARGET DOWN

158. GO

159. MOVE THE MAGNET

160. FARTHEST TO THE RIGHT

161. DOWN

162. AND TO THE MIDDLE

163. GO

164. STOP

165. MOVE THE MAGNET AT THE BOTTOM OF THE SCREEN DOWN

166. GO

167. STOP

168. PUT THE MAGNETS

169. PUT THE TWO MAGNETS THAT ARE FARTHEST FROM THE BLUE BALL BACK INTO THE BAG OF THINGS

170. PUT THE SECOND MAGNET REMOVED BACK ONTO THE SCREEN

171. PUT THAT MAGNET BACK WHERE IT WAS BEFORE IT WAS TAKEN AWAY

172. MOVE THAT MAGNET TO THE LEFT

173. MOVE THAT MAGNET TO THE LEFT AGAIN

174. MOVE THAT MAGNET TO THE LEFT AGAIN

175. GO

176. MOVE THAT MAGNET SO THAT IT IS DIRECTLY UNDER THE LARGE BLUE BALL

177. GO

178. MOVE THAT MAGNET

179. MOVE THE MAGNET THAT IS CURRENTLY

180. ABOVE

181. THE BLOCKS

182. SO THAT IT IS IMMEDIATELY TO THE LEFT OF THE BLUE BALL

183. MOVE THE MAGNET AT THE BOTTOM OF THE SCREEN

184. VERY SLIGHTLY TO THE RIGHT

185. GO

186. STOP

187. MOVE THE MAGNET TO THE LEFT OF THE BALL SLIGHTLY UPWARD

188. GO

189. MOVE THE MAGNET ON THE LEFT OF THE SCREEN UP VERY SLIGHTLY

190. GO

191. THE THIRD MAGNET

192. PLACE IT ON THE BOTTOM OF THE SCREEN IN THE VERY MIDDLE

193. MOVE

194. THAT MAGNET DOWN

195. MOVE THAT MAGNET DOWN

196. PLEASE MOVE THE MAGNET IN THE MIDDLE OF THE SCREEN AT THE BOTTOM

197. DOWN SLIGHTLY

198. PLEASE MOVE THE MAGNET

199. THAT IS IN THE MIDDLE OF THE SCREEN

200. THE BOTTOM DOWN

201. PLEASE MOVE THE MAGNET THAT AT IS AT THE BOTTOM OF THE SCREEN ON THE LEFT UP SLIGHTLY

202. GO

203. STOP

204. PLEASE MOVE THE MAGNET SLIGHTLY TO THE LEFT

205. GO

206. STOP

207. GO

208. STOP

209. PLEASE MOVE THE MAGNET ON THE BOTTOM OF THE SCREEN ON THE LEFT SLIGHTLY TO THE LEFT

210. GO

211. GO

212. NO STOP

213. GO

214. PLEASE MOVE THE MAGNET AT THE VERY VERY BOTTOM OF THE SCREEN SLIGHTLY TO THE LEFT

215. PLEASE MOVE THE MAGNET AT THE BOTTOM OF THE MIDDLE SCREEN SLIGHTLY TO THE LEFT

216. GO

217. PLEASE MOVE THE MAGNET AT THE VERY BOTTOM OF THE SCREEN SLIGHTLY TO THE LEFT

218. GO

219. STOP

220. GO

221. PLEASE MOVE THE MAGNET ON THE BOTTOM LEFT OF THE SCREEN UP SLIGHTLY
222. PLEASE MOVE THE MAGNET ON THE BOTTOM LEFT OF THE SCREEN UP OK GO
223. GO
224. STOP
225. STOP
226. GO
227. STOP
228. GO
229. PLEASE
230. PLEASE MOVE THE MAGNET
231. THAT IS
232. IN THE TOP LEFT
233. DO NOT MOVE THAT MAGNET
234. PLEASE MOVE THE MAGNET THAT IS AT THE VERY BOTTOM OF THE SCREEN UP SLIGHTLY
235. STOP STOP
236. PLEASE MOVE THE MAGNET THAT IS THE VERY BOTTOM OF THE SCREEN UP SLIGHTLY
237. PLEASE MOVE THE MAGNET THAT IS AT THE VERY BOTTOM OF THE SCREEN UP SLIGHTLY
238. GO
239. STOP
240. GO
241. PLEASE MOVE TO THE NEXT PUZZLE
242. PLEASE PLACE A MAGNET DIRECTLY UNDER THE MIDDLE TARGET
243. PLEASE PLACE A MAGNET DIRECTLY UNDER THE MIDDLE TARGET
244. PLEASE MOVE THAT MAGNET UP SLIGHTLY
245. GO
246. STOP
247. GO
248. STOP
249. GO
250. PLEASE PLACE A MAGNET
251. DIRECTLY ABOVE THE NO STOP STOP
252. PLACE A MAGNET DIRECTLY ABOVE THE PINK BALL
253. GO
254. STOP
255. PLEASE RETURN TO THE FIRST PUZZLE
256. PLEASE PLACE A MAGNET
257. NEXT TO ON THE RIGHT AND SLIGHTLY ABOVE THE SMALL RED BALL
258. PLEASE PLACE A MAGNET
259. DIRECTLY ON THE LEFT OF THE SMALL BLUE BALL
260. PLEASE PLACE A MAGNET
261. ON THE FAR LEFT OF THE SCREEN
262. AT THE SMALL AT THE SAME LEVEL AS THE BLUE BALLS
263. GO
264. GO
265. STOP

Bibliography

[1] Jeff A. Blimes. A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, International Computer Science Institute, 1998.

[2] Eric Brill. Some advances in transformation-based part of speech tagging. In *National Conference on Artificial Intelligence*, pages 722–727, 1994.

[3] V. Zue et al. Jupiter: A telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, 2000.

[4] B. Mak J. Junqua and B. Reaves. A robust algorithm for word boundary detection in presence of noise. *IEEE Transactions on Speech and Audio Processing*, 1994.

[5] N. Kanwisher K. O'Craven, P. Downing. fMRI evidence for objects as the units of attentional selection. *Nature*, 1999.

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

[7] A.E. Rosenberg L.F. Larnel, L.R. Rabiner and J.G. Wilpon. An improved endpoint detector for isolated word recognition. *IEEE Transactions in Acoustics, Speech, Signal Processing*, 1981.

[8] Beatrice Santorini Mitchell P. Marcus and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1994.

[9] Niloy Mukherjee. A technical report on speech recognition research in the cognitive machines group. Technical report, MIT, 2002.

[10] E. Shriberg and A. Stolcke. Harnessing speech prosody for human-computer interaction. Presentation at the NASA Intelligent Systems Workshop, February 2002.

[11] Shimon Ullman. *High Level Vision: Object Recognition and Visual Cognition.* MIT Press, 1996.

[12] Benjamin Yoder. Spontaneous speech recognition using hmms. Master's thesis, Massachusetts Institute of Technology, 2001.