Prosody-Based Automatic Detection of Punctuation and Interruption Events in the ICSI Meeting Recorder Corpus

Don Baron

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II.**

Approval for the Report and Comprehensive Examination:

Committee:

Professor Nelson Morgan
Research Advisor

May 26, 2002

* * * * * * *

Dr. Elizabeth Shriberg
Second Reader

(Date)

Acknowledgements

Throughout the course of this work, I have been lucky enough to work with some extremely bright and generous people at the International Computer Science Institute, without whom this project (and many others) would not be possible. In no particular order, many thanks go out to Thilo Pfau, for the work on his speech/non speech detector; Jane Edwards, for her tireless work towards transcription perfection; and Adam Janin, who helped me tremendously in learning Perl and the meeting recorder infrastructure. Barbara Peskin and Chuck Wooters have been excellent project leaders, and Morgan's navigation helped this ship set sail. Annotators Sonali Bhagat, Ashley Krupski, and Raj Dhillon have provided both pristine dialog act annotations and an endlessly good time in the office. Last, but certainly not least, Liz Shriberg and Andreas Stolcke deserve my deepest gratitude for their patience, time, and constant willingness to help me with this project and report. Without their guidance and support my databases would be empty, my models would perform at chance, and my experience at ICSI would certainly not be the same.

Table of Contents

1. Introduction

1.1 The importance of meetings

In recent years, speech researchers have taken a greater interest in the automatic processing of natural multi−person meetings. Meetings constitute a ubiquitous form of human communication, and present unique research challenges (A. Waibel et al., 1998, N. Morgan et al., 2001). While better word recognition is an important goal in much of this work, interest is also shifting toward higher−level tasks, such as information extraction and summarization.

For such tasks to succeed, information currently not in speech recognition output, such as punctuation, disfluencies and overlap markings must be available. Figure 1.1 illustrates the importance of these markings. The first line of the figure show a potential word stream, as derived from a speech recognizer from a far field microphone, while the last lines contain the same words but includes punctuation and turn taking markings. Readability and understanding increase dramatically in the latter case, showing that while extremely important, words alone do not paint the whole picture in conversation understanding.

Without Structure:

```
did you know that yeah i went to the beach oh today really was it crowded no
```

With Structure:

```
A:                        I went to the beach today                    No.
B:                Yeah.                       Oh really?  Was it  Crowded?
C: Did you know that?
```

Figure 1.1 A word stream with and without punctuation and turn taking annotations. Overlapped words are in boldface and clearly indicate where speakers interact with one another.

1.2 The importance of prosody

In order to develop successful automatic classifiers for these tasks, features associated with such events and that can be extracted automatically must be utilized. While words themselves play a role in indicating certain punctuation and dialog events,

many of the cues used to predict semantic and pragmatic structures are characteristics beyond word identities alone. *Prosody*, or the timing, pitch, and energy patterns of speech, has been observed to be related to such events as punctuation and turn−taking (E. Couper−Kuhlen and M. Selting, 1996). Furthermore, since prosodic features are, by definition, independent of word identity, one can expect that they may offer robustness to automatic speech recognition errors for machine processing of meetings.

1.3 Overview and scope of this project

In this study prosodic features such as pitch, speaking rate, energy, and pause durations are automatically extracted and modeled for the purpose of classifying a variety of punctuation and dialog events. In past work (E. Shriberg, et al., 2000, E. Shriberg, A. Stolcke, D. Baron 2001, J. Buckow, et al., 1999), prosodic features have been shown to be extremely useful in punctuation, disfluency, and interruption event classification. This report focuses on extending the use of prosody to the domain of natural meetings using a collection recorded at the International Computer Science Institute (ICSI). This corpus presents new challenges, because speakers are familiar with one another, have access to other cues such as gesture, are not typically constrained to one topic as they are in corpora such as Broadcast News or Switchboard, and because of the high degree of speaker overlap and presence of multiple speakers.

This study uses automatically derived prosodic features, including stylized pitch, pause durations and energy statistics, based on both forced alignments and recognized words, to build a prosodic classifier for various events of interest. An analysis of performance degradations in the ASR−based feature set is included and provides some useful observations regarding the feasibility of a fully automatic system in the presence of word errors. The value of "online" classifiers, which have no access to future features and would therefore be used in real time systems, is assessed and compared to the case of the full feature set. This comparison is relevant for ongoing research (Y. Matsusaka, et al., 2001) where robotic conversational agents participate in meetings and interact with human participants. In order for such a machine to function well, it must master the

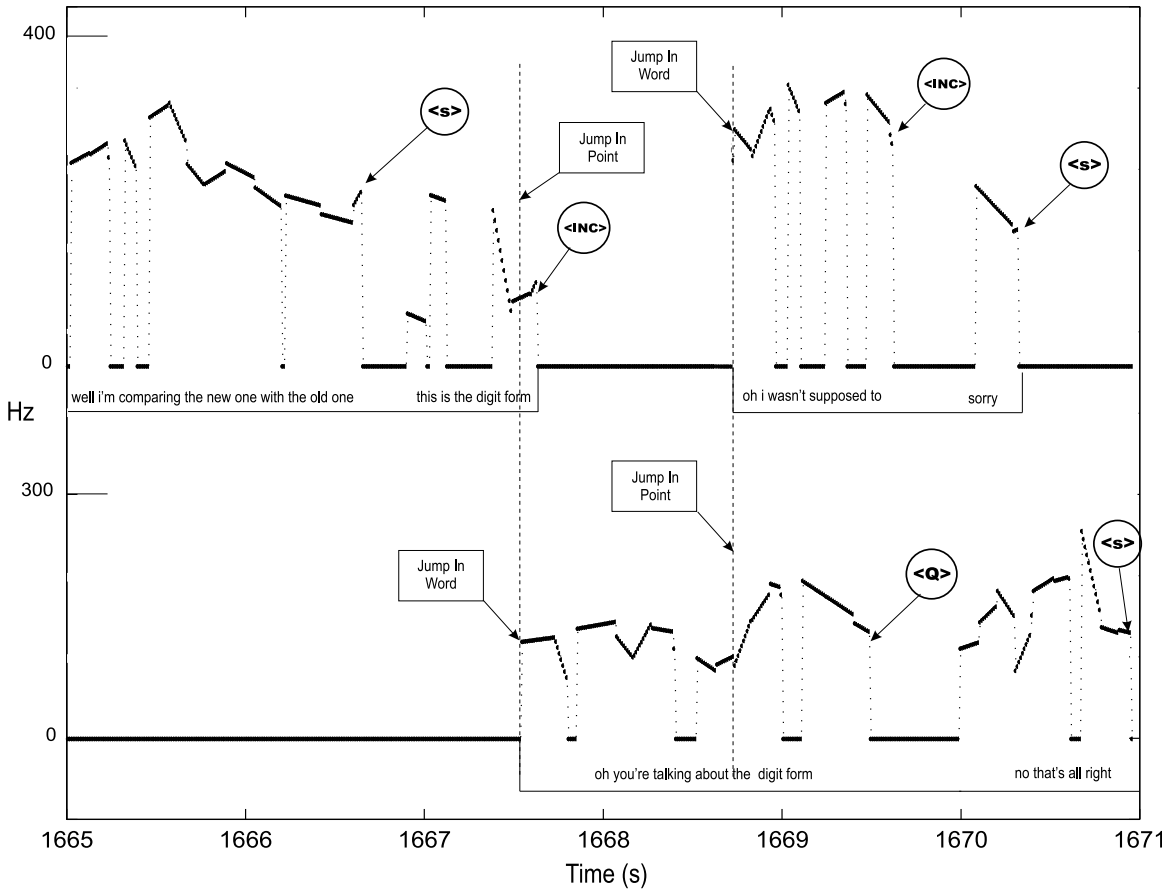


Figure 1.2: Excerpt from a meeting illustrating overlaps, punctuation boundaries, jump−in points, jump−in words, and stylized F0 contours for two speakers (female = top, male=bottom). Spurts, or regions of speech with no more than 0.5s of silence, are delimited by vertical lines enclosing words. Circled tags mark punctuation events, including <S> (sentence boundary), <INC> (incomplete sentence) and <Q> questions. Overlap events are indicated by square boxes. A jump−in point for one speaker corresponds to a jump−in word for the other and vice versa. Note than jump−in words are always spurt−initial, as defined in the text.

prediction of pragmatic and semantic events.

Where applicable, results are compared to a language model classifier, which provides a measure of the usefulness of words alone in our classification tasks. Finally, the performance of a combined prosodic and language classifier is assessed. These event classification systems allow for feature analysis across tasks that provide important insights on the usefulness of various cues in both human and machine prediction.

1.4 Tasks

This section gives a high−level description of the five classification tasks

explored in this study. (The reader is referred to Section 4.1 for a more detailed description of the tasks.) Figure 1.2 shows a stretch of speech and is referred to below in the descriptions of the tasks. Table 1.1 offers a brief description of the tasks, including the number of event classes and whether the classes are structural units (such as sentences and disfluencies) or dialog and speaker interaction related. Punctuation based classifiers greatly increase the readability of a word stream, while the dialog and interaction tasks provide useful information about meeting flow and participant behavior.

	Name	**Description**	**# classes**
Punctuation	s−ns	sentence/non−sentence	2
	s−di−n	sentence/disfluency/neither	3
Punctuation and Dialog Act	s−q	Question/declarative sentence	2
Interaction	Jump−In Points	Word boundary a point of interruption for a bg speaker?	2
	Jump−In Words	First word of spurt in silence or someone else's speech?	2

Table 1.1: Description of 5 tasks discussed in this chapter, along with number of classes for each task

1.4.1 Task 1

Task 1 aims to distinguish sentence boundaries from fluent (non−sentence end) word boundaries (E. Shriberg, et al., 2000, Shriberg et al. 2001), In order to simplify the class groupings, disfluencies, fluent boundaries, and incomplete sentences are all grouped into one class, despite the fact that there are inherent prosodic differences among these groups. Similarly, question ends and declarative sentence ends are also grouped together. The latter two events are denoted by the circled <S> and <Q> markings, respectively, in Figure 1.1. The following example illustrates the word boundary labels for this task.

do	you	−	i	know	.	what	was	that	?
<ns>	<ns>	<ns>	<s>	<ns>	<ns>	<s>			

For Task , 1 the effects of using recognized words are examined, as is performance degradation when the classifier has access only to past features (features occurring before the label boundary).

1.4.2 Task 2

Task 2 is also a punctuation classification task, but where each word boundary is classified as a sentence−end, disfluency, or fluent boundary. This is similar to Task 1, but disfluencies and incomplete sentences are considered in a separate class, as shown in the figure below:

```
do    you   −     i     know   .   what   was     that  ?
  <n>       <d>       <n>      <s>       <n>    <n>       <s>
```

The introduction of a separate disfluency class is of interest since there are different inherent prosodic characteristics between tokens in this class and those in the <n> class. As in Task 1, the role of word errors, along with the "online" feature set performance is examined. In addition, speaker specific data sets are used to train models, to examine how smaller, but more prosodically and lexically consistent data sets fare, as compared to the full speaker data set.

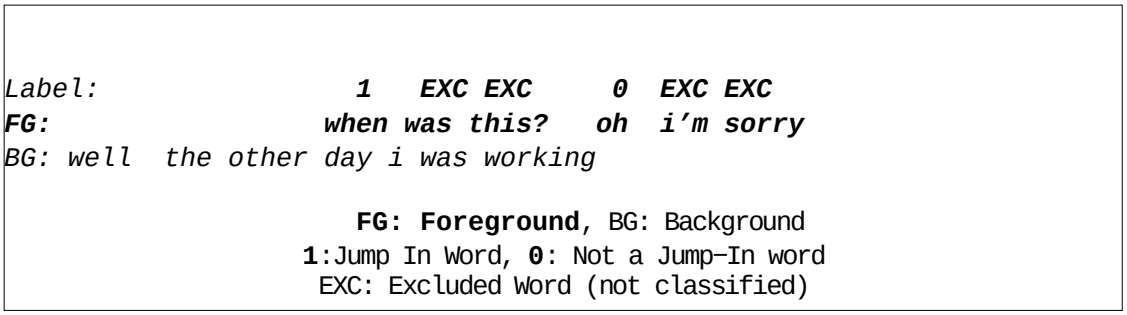
1.4.3 Task 3

In Task 3, sentence ends are classified as either question or declarative sentence ends as shown below:

```
do    you   −     i     know   .   what   was     that  ?
  EXC       EXC       EXC      <P>       EXC    EXC       <Q>

          EXC = Excluded, Q = Question, P = Period
```

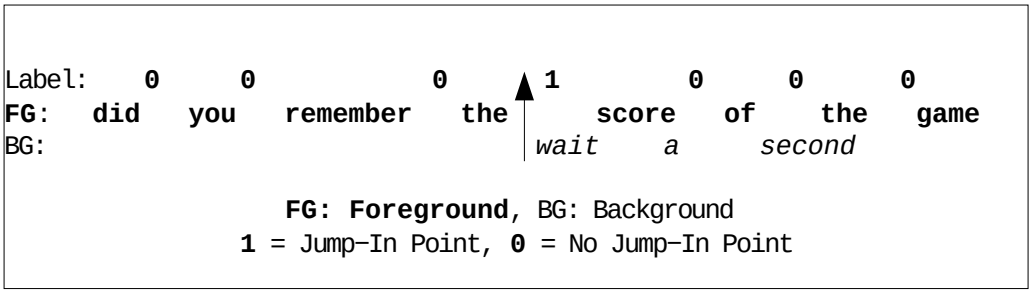
Note that since only sentence ends are considered, language model performance is not



included in this case, since the LM used in this report requires the inclusion of all datapoints (word boundaries). The goal here is to find out whether there are inherent prosodic differences between question and sentence ends.

1.4.4 Task 4

Task 4 and 5 are more exploratory and deal with interactive conversational events such as overlap and speaker interruption. In Task 4 the following question is asked: "given the prosodic features of a foreground speaker, can classifiers predict where a background speaker will interrupt?" Points of interruption, as seen in Figure 1.1, are called "Jump−In Points" and an example is shown below.



Note the inherent uncertainty involved in this classification task: decision trees attempt to predict prosodically advantageous points of interruptions with no knowledge at all about whether the background speaker wanted to interrupt at other places but did not.

1.4.5 Task 5

Finally, Task 5 analyzes spurt initial words (the events corresponding to Jump−In Points), where a spurt is defined as a region of speech with no more than 0.5s of silence. These spurt initial words are classified as starting in silence or starting in another speaker's speech. The latter case is considered a Jump−In Word and is illustrated below:

```
Label:                1   EXC EXC     0  EXC EXC
FG:                 when was this?   oh  i'm sorry
BG: well  the other day i was working

                      FG: Foreground, BG: Background
                    1:Jump In Word, 0: Not a Jump−In word
                     EXC: Excluded Word (not classified)
```

As in Task 3, since not all datapoints are included in this task, a language model is not used in determining a word−only baseline metric.

1.5 Outline of Chapters

This report is organized as follows: Chapter 2 discusses the steps that were required in building and organizing the prosodic and lexical databases used in the tasks described above. Specifically, automatic speech presegmentation, human annotation, pitch extraction and stylization, and feature descriptions are included, along with a discussion of the prosodic and language model classifiers. Chapter 3 presents ASR results and observations on the nature of overlaps in the Meeting Recorder corpus. Chapter 4 reports the results and discussions of the five tasks, with comparisons across tasks and conditions. Finally, Chapter 5 concludes the study and offers some ideas for future work.

2. Methods

2.1 Meeting recordings

The tasks and experiments described in this thesis use the ICSI Meeting Recorder corpus (Morgan, et al., 2001) as the data set. This corpus currently contains meetings recorded at ICSI and may in the future include meetings from other sites including the University of Washington, Columbia University and SRI.

Meetings were recorded from February 2000 and continue to be recorded as this report is being written. The target of 100 hours of recording should be completed by summer, 2002. ICSI meetings were conducted in an on−site conference room, which was equipped with various types of microphones, including close talking head−mounted mics, table−top microphones for far−field work, and inexpensive microphones attached to a mock PDA. Considering the number of open research questions within the meeting context, the different microphone types allow for analysis in both near−field and far−field domains.

As described in (ICSI MR web page, 2001), the audio signals were fed to an A/D that was connected to a workstation in the back of the room. This workstation displayed microphone levels on the screen and allowed the meeting operator to adjust gains where appropriate, providing one setting per channel at meeting onset.

Before their first session, meeting participants were required to fill out a form that asks a number of questions, including speaker name, gender, degree of nativeness in American English, and age. These statistics were committed to a centralized database to allow partitioning of speakers by these various features. Once a speaker is in the database, he or she has a unique speaker tag which is embedded in the file names of utterance level audio segments. As explained below, statistics like gender, name, and the native/non−native distinction, are extremely important because many experiments, including ASR, interruption prediction, and punctuation modeling, are correlated with some or all of these factors.

Each meeting generates a "KEY" file which contains information about speaker participants, microphone gains, dates and times of recording, and any other information

deemed relevant (i.e., poor recording on a channel).

2.2 Data and segmentation

2.2.1 Data and train/test partition

This research focusses on three meeting types, as they have the most data. These meeting types are the *Bmr*, *Bro* and *Bed* meeting sets. Table 2.1 describes the amount of data used in this study for each meeting type. The *Bmr* set is composed of 13 meetings about the Meeting Recorder project itself. These recordings usually involve between four to eight speakers, most of whom are native speakers and familiar with one another, creating a speaking environment rich in overlaps, interruptions, disfluencies and other conversational events of interest. The *Bro* meetings are recorded discussions on recognizer front end issues and tend to have more of a seminar style, with one active speaker holding the floor for a majority of time. The *Bed* meetings, which are comprised of various topics in ICSI's Artificial Intelligence group, fall somewhere in between the *Bmr* and *Bro* types, in terms of speaker distribution. These contrasting styles provide for a different landscape in terms of speaker interaction and interruptions. A more detailed discussion on the pervasiveness of overlaps across meeting types is found in Chapter 3.

	Meeting Type			
	Bed	Bmr	Bro	Total
Number of Meetings	7	13	12	32
Total Speech Duration	7.0 h	13.7 h	11.2 h	31.9 h
Transcribed Words	67,546	145,150	94,261	306,957
Speech Spurts	8,254	15,414	11,821	35,989

Table 2.1 Data Collected. Speech spurts are regions of speech interrupted by pauses of no greater than 500 ms.

The full data set was partitioned into train and test sets, where the test set had approximately 18% of the total meeting time. In partitioning, one must be careful not to

include the same data in both sets. To avoid this problem, individual meetings were not allowed to be split up across the train/test sets, because if overlap speech existed, for example, background speech in the train set may be in the foreground of a train test, and the prosodic models trees may be inadvertently trained on this test case. While meetings were not split across the test and train boundary, some speakers appear in both sets. This unavoidable since so many speakers appear in multiple meetings. This is also a perfectly reasonable scenario in a real−word application, where meetings will involve a mix of recurring and unknown participants.

2.2.2 Data presegmentation

Once a meeting was recorded, a speech/non−speech detector (T. Pfau, et al., 2001) segmented each channel into regions of interest. This presegmentation technique employs a hidden Markov model (HMM) that has two main frame−level states, one for speech and one for non−speech frames along with a number of intermediate states that impose speech to non−speech (and vice versa) time constraints. In addition to this structure, the presegmenter also does additional post−processing of numerous features (energy, zerocrossings, loudness) that accounts for significant detection of a speaker's cross talk on a different channel − an important issue when recording multiple speakers in close proximity of one another. The result is "segments" which are short , ideally utterance−level waveforms that are surrounded by some user−defined amount of silence. In our experiments this value was set to 500ms, which is the same value used in Switchboard acoustic segmentation.

After segments have been defined, the meeting was either handed off to human transcribers for creation of a manual transcript, or run through an automatic speech recognizer (ASR) for an automatic version of the transcript. Originally, it was intended that the segment boundaries over which ASR was to be run would not be hand−adjusted at all, thereby providing a purely automatic database. After numerous experiments, however, it was decided that non−annotated segmentations were not accurate enough to feed to the speech recognizer, and that ASR segments would also be hand adjusted so as to maximize word recognition (and thereby prosodic feature) performance. In many ways, such a system is preferred because the purpose of this research is to assess the

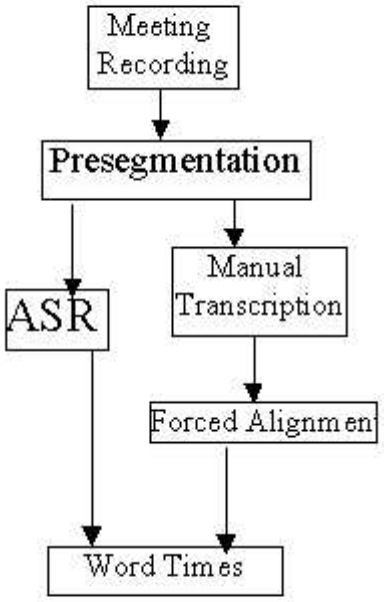


Figure 2.1: Word time extrapolation. Meetings are recorded and then either manual or automatic transcripts are generated. In the manual case, forced alignments are used to create word times.

usefulness of prosodic features, not the segmentation. There is reason to believe that an accurate presegmentation would be available in the future. Thus, by using hand−adjusted (assumed to be ground truth) segment boundaries, poor performance in classification tasks can be isolated to feature extraction and/or processing, rather than to some other step which is not an essential part of this research.

Figure 2.1 illustrates how word times are extrapolated in both the ASR and the manual transcription cases.

2.2.3 Transcripts

In order to utilize prosodic features at the word level, reliable word boundaries are necessary for delineating where words start and end. In the ASR databases, word

times were retrieved from the backtraces of recognition results, but for manual transcripts, forced alignments were used to generate these word boundaries. Once listeners had completed creating transcripts, a number of steps were taken to prepare the forced alignment. First, transcripts were separated into channel specific reference files. The transcripts themselves have the following form:

```
<Sync chan="3" time="498.566"/>
N: Yeah, then we're *completely gone. That's
–
<Sync chan="3" time="500.694"/>
{VOC laugh}
<Sync chan="4" time="497.290"/>
J: uh, no problem. I mean, I'm not saying
accents. I'm say– I'm saying fluency. Well,
yeah.
<Sync chan="3" time="502.540"/>
..
<Sync chan="4" time="502.540"/>
..
<Sync chan="1" time="505.198"/>
A: Well, I think that, um ..
```

Since the above form includes all channels in one file, some processing is required in order to make word reference files on a per channel basis. Transcript parsing yielded files which are much more amenable to our processing steps:

```
Bmr006_me013–s1–w1–3_0486240_0494059          [laugh]  Oh<EXC> You're not  talking about
                                              foreign language at all<PER> You're just
                                              talking about <D>  [laugh]

Bmr006_me013–s1–w1–3_0494059_0498566          EMPTY

Bmr006_me013–s1–w1–3_0498566_0502540          Yeah<COM> then we're completely gone<PER>
                                              That's <D>  [laugh]

Bmr006_me013–s1–w1–3_0502540_0502920          EMPTY

Bmr006_me013–s1–w1–3_0502920_0505209          The <D> the habits are already burnt
                                              in<PER> But <D>
```

```
Bmr006_me013-s1-w1-3_0505209_0512320          EMPTY

Bmr006_me013-s1-w1-3_0512320_0512826          Yeah<PER>
```

Example waveid:

Bmr006_mn005-s1-w1-2_4232466_4233142

Bmr | 006 | m | n | 005 | s1 | w1 | 2 | 4232466 | 4233142

Example	Category	Values	Notes
Bmr	Meeting Type	B**	meetings recorded at ICSI
		W**	meetings recorded at UW
		S**	meetings recorded at SRI
		I**	meetings recorded at IBM
		N**	meetings recorded at NIST
006	Meeting Number	001-999	
m	Gender	m/f	male/female
n	Native American English Speaker?	e/n	english/non-native
005	Speaker Number	001-999	
s1	Microphone Type	s1	Sony handheld mic WRT-807A
		s2	Sony headset mic ECM-310BMP
		c1	Crown headset mic CM 311 A/E
		l1	Sony lapel (lavalier) ECM-77BMP
		p1	Plantronics monaural headset mic
		a1	Andrea monaural headset mic NC-50
w1	Connection Type	w1/j1	wired jack
		s1/s2	Wireless Sony transmitter/receivers
2	Channel Number	0,1,2,3,4,5,8,9,A,B	
4232466	Start Time in ms	0000000-9999999	
4233142	End Time in ms	0000000-9999999	

Table 2.2 Segment filename explanation

This file is a two column field, with the left column indicating the segmented filename and the right column corresponding to the words in that filename. Table 2.2 describes the different fields within the segment file names. These filename attributes allow for unique filenames across different meetings across different possible sites and contain important channel−specific information such as speaker, microphone type, and whether the speaker is a native American English speaker.

2.3 Word Alignments

2.3.1 Forced alignments

All the reference files for a particular meeting were concatenated together and each line sent to the recognizer, which attempted to find word boundaries for that segment's transcription, within the audio file provided. The forced alignments were created using the alignment option of the SRI Hub 5 recognizer (A. Stolcke, et al., 2000), which employs vocal−tract length normalization and feature normalization on a per−channel basis. The alignment attempts to match words from the transcript to the acoustic signal by finding the most likely sequence and duration of phones in each word, based on the SRI dictionary. Any words which were not present in the dictionary (i.e., technical terms, proper nouns, foreign words) are reported as "out of vocabulary", were entered manually into the dictionary, and forced alignments were recalculated.

2.3.2 Recognition−based alignments

A much more difficult but interesting task is to extrapolate these time boundaries from automatic recognition results and compute features from there. As mentioned the use of hand−adjusted segment boundaries, makes this step not completely automatic, but it is assumed that perfect segmentations will be available in the future. For recognition, the same recognizer (SRI March 2000 Hub 5) that was used in alignments was used here, with the addition of phone−loop speaker adaptation on each conversation channel, excluding areas with crosstalk. In addition, a bigram language model (LM) of about 30,000 words was used. The LM was trained on Switchboard, CallHome, and Broadcast

News data. These three corpora provide a wide array of speech contexts: Broadcast News is read news programming, which is a very different style to spontaneous conversations, but Switchboard and CallHome provide a large body of phone conversations that are extremely helpful in modeling word usage and turn taking endemic to multi−party conversations such as meetings.

Note that the recognizer was not tuned specifically to the meeting corpus. This was mainly to allow comparisons across these varying corpora. Certain front end features, such as downsampling to 8KHz (to telephone bandwidths from the original 16KHz sampling rate) remain in the recognizer, though preliminary experiments showed negligible performance loss due to this downsampling. ASR results are reported in Chapter 3.

2.3.3 Alignment files

Word boundaries were automatically generated when ASR was performed on a meeting−channel side. The alignments (either from ASR or the forced alignments) were further processed into a more useful database file. These alignment files take the following form:

```
Bmr006 c3 habits 3 8 503.43 0.38 hh:8_ae:8_b:6_ax:6_t:4_s:6
Bmr006 c3 are 4 8 503.81 0.09 er:9
Bmr006 c3 already 5 8 503.9 0.22 ao:3_l:3_r:3_eh:3_dx:3_iy:
Bmr006 c3 burnt 6 8 504.12 0.3 b:9_er:11_n:3_t:7
Bmr006 c3 in 7 8 504.42 0.21 ih:11_n:10
```

The first two fields indicate the conversation (meeting and channel), then the word is listed, along with the word position in its segment, the total number of words in a segment, the start time of the word (in seconds), and the word duration. The latter two features described are used to line up pitch values to pitch features. The last field shown above is a concatenation of all phones in a particular word, along with the frame length of that particular phone. Phone features were later separated into vowel/non−vowel phones and vowel phones were then used to give a first order approximation to speaking

rate.

2.4 F0 extraction and stylization

One main goal of this work is to automatically extract robust and useful prosodic features which will be used in classification experiments. To accomplish this goal, pitch features must be extracted from the audio files and then lined up with the word times. Then word−level feature values can be calculated, based on the time boundaries provided.

In creating pitch contours for our data set, several important steps were taken. First, it was necessary to extract raw F0 values from the audio signal. A number of techniques are available for pitch detection (B. Secrest and G. Doddington, 1983, W. Hess, 1983), but the ESPS `get_f0` package (Entropic, 1993) was used in this work for convenience. This package, based on (D. Talkin, 1995) uses the normalized cross correlation function (NCCF) which is more robust to fast F0 variations than a simple autocorrelation function (ACF). The NCCF function is defined as:

$$\phi_{i,k} = \frac{\sum_{j=m}^{m+n-1} s_j s_{j+k}}{\sqrt{e_m e_{m+k}}}$$

Where *m* is the sample number for frame *i* and *n* is the length of the analysis window. The normalization factor *e* is defined as:

$$e_j = \sum_{l=j}^{j+n-1} s_l^2$$

Both ACF and NCCF correlate adjacent samples within a potential vocal frequency range as they look for potential periodicity, but the former is more able to discern rapidly changing F0 values, and is therefore used in this study.

Before this correlation function is computed at each frame, `get_f0` first downsamples the audio data signal and finds areas of high correlation in this coarse

version of the signal. Then more careful analysis is employed near the areas of interest, producing a final high resolution NCCF signal. Once this is completed, dynamic programming is used to determine the best F0 value and voicing state (voiced/unvoiced) based on various combinations of local and contextual artifacts. F0 features were extracted from each speech segment and are saved out for further processing. Alternatively, pitch values over an entire channel could have been computed, but this would be significantly more data than required, because many speakers have relatively few areas of speech on their channels.

2.4.1 Pitch normalizations

Although this pitch estimation technique is fairly robust, it is, like other pitch trackers, prone to octave errors. Along with this potential source for error, `get_f0` does not provide the user with any sense of relative pitch rise for a particular speaker. If, for example, a female speaker has a pitch value at some frame of say, 500 Hertz, this has a different meaning than if a male speaker had the same frame level pitch value. Pitch normalization is necessary to account for both of these concerns.

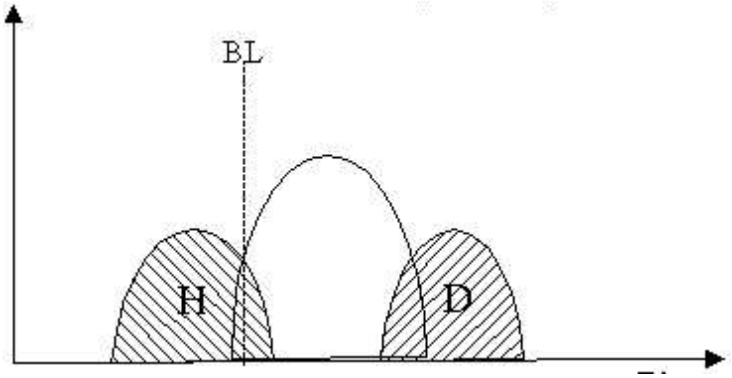


Figure 2.2: Log−Tied Model for pitch detection. Area H represents observations of pitch halving, while area D represents pitch doubling events. Line BL indicate the Baseline for correct pitch detection.

The first goal of this step was to identify the lowest, or baseline, pitch value for a particular speaker operating in normal (rather than halving or doubling) mode. To determine this baseline pitch value, all of the F0 values from a speaker's channel are accumulated and fitted to a log−tied normal (LTM) model, as shown in Figure 2.2.

This model presupposes that there are three distinct means in the F0 distribution. The center mean represents the most common non−halved, non−doubled pitch value, with a normal distribution in log domain. Area H in Figure 2.2 represents pitch values which have been halved as a result of tracking error from the `get_f0` routine or vocal fry (a "creaky" voice condition when only the front part of the speakers' vocal cords are vibrating) , whereas area D represents the doubled region. All real world pitch trackers are subject to a certain amount of inherent doubling/halving, despite attempts at removing such errors via post processing and dynamic programming. Nonetheless, it is important to account for errors, and this LTM model uses an expectation maximization (EM) algorithm to find the correct mean and extrapolates means under the halving (log(u/2)) and doubling (log(2u)) regimes**.**

Once this EM algorithm is completed, LTM parameters such as mean and variance, were returned to the user. From these values, an important baseline metric that is crucial in understanding relative pitch rises and falls is inferred. In Figure 2.2, the pitch value BL indicates the point at which the probability of an accurate pitch is equal to the probability of halving. In our processing and feature extraction, this was taken to be the lowest non−halved pitch value, or baseline F0 value −− the point from which all pitch rises and falls are to be measured.

2.4.2 Pitch stylization

After the distribution of pitch values has been modeled, the frame level F0 values are to be stylized to remove microintonations along with noisy pitch values and errors from halving/doubling as done in (K. Sonmez, 1998). In addition, a linearization of the data would be ideal, because line fits would provide a suitable and tractable method of interpreting tonal contours, shapes, and ultimately, slopes. Each segment was run through the pitch tracker, and frame level raw pitch features were again determined. These features were then compared to the LTM model and halving/doubling posterior probabilities are computed. Those frames which had halving or doubling posteriors greater than the posterior probability of "normal" speech were automatically excluded from prosodic feature calculation, as they are unreliable due to either poor pitch tracking or the aforementioned vocal fry phenomenon.

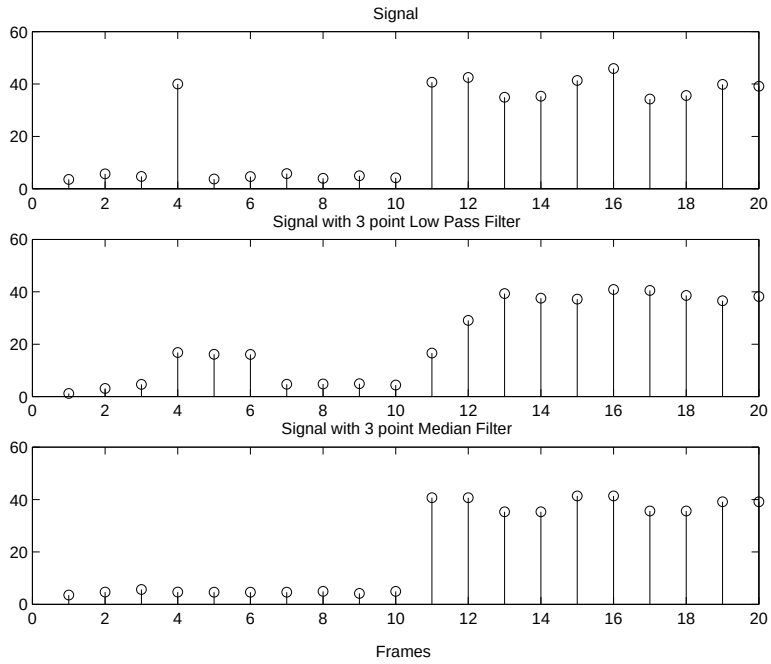


Figure 2.3: A comparison between low pass filtering (LPF) and median filtering. Median filtering attains smoothing while preserving edges. This operation also removes outlying frames; sample 4 above is removed after median filtering, but averaged in the case of the LPF.

Following the computation of halving/doubling posteriors, median filtering was applied to the raw pitch values. Median filtering is a non−linear filtering step which is similar to low pass filtering (LPF) in the sense that slight variations in the signal are removed, but unlike a low pass filter, median filtering still preserves the integrity of "edges", or sudden transitions to a new level in the signal. The median filter looks at five samples and rather than averaging them like an LPF system, chooses the value which falls in between the other two. Five samples were chosen as this value allowed for a nice tradeoff between smoothing and edge preservation. A comparison between the result of low pass filtering and median filtering can be seen in Figure 2.3. Because median filtering requires three samples for successful processing, the first and last two frame values in a segment's pitch file were discarded, but this loss is insignificant since segments generally tend to be significantly larger than these 40ms of pitch values near the segment boundaries.

A piecewise linear fit (PWL) algorithm based on (K. Sonmez, 1998) was used to create line estimates for the median−filtered F0 values. As mentioned above, these estimates are extremely useful for error correction and in quantifying tonal trends. On a

particular voiced region, the PWL algorithm attempted to fit lines by minimizing the following mean square error (MSE) criteria:

$$(x_k^*, y_k^*)_{k=0}^{K} = \arg \min_{(x_k, y_k)_{k=0}^{K}} \frac{1}{T} \sum_{t=1}^{T} (f_0(t) - g(t))^2$$

where f(t) is the linearized pitch estimate and g(t) are the raw F0 values, and Xk and Yk are the node locations. After the algorithm picked these best fit nodes, the resultant pitch contour is the summation across all of these nodes for the voiced region in question:

$$g(x) = \sum_{k=1}^{K} (a_k x + b_k) 1_{[x_{k-1} < x \leq x_k]}$$

Figure 2.4 shows a stylized pitch contour and its original raw F0 pattern.

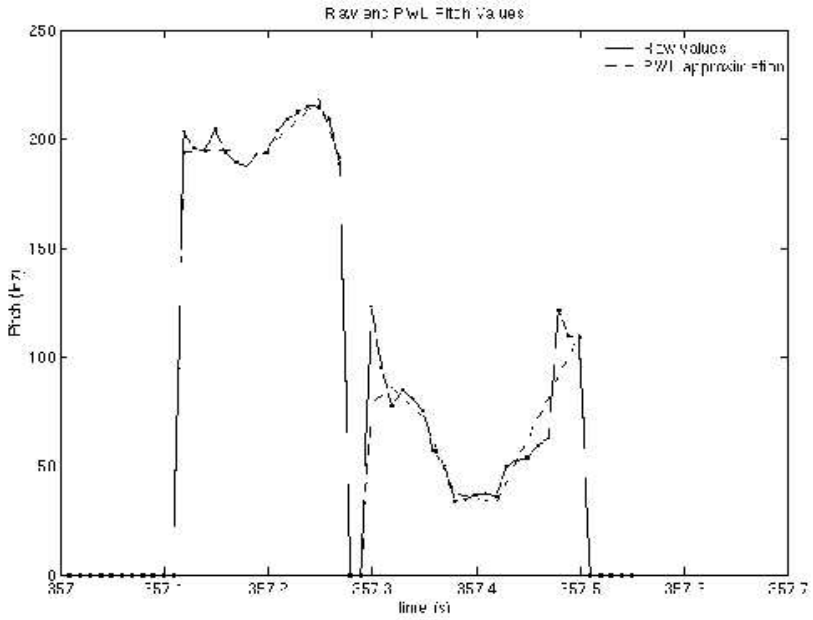


Figure 2.4: Raw and stylized pitch contours for a stretch of speech

2.5 Language model

In order to determine the relative gain of using prosodic features in the various classification tasks, it is important to provide a baseline performance metric over which one can assess the value of the system. One way to tell how a prosodic feature based system performs is to compare it to a system which only uses language model (LM) features. Such a system would consider certain events, such as punctuation, as hidden events , while transcribed (assumed true) or ASR gotten words are treated as observations. Figure 2.5 illustrates this idea.

The purpose of the LM is, as mentioned above, to determine the usefulness of the prosodic features alone. In other words, this language model will be used for a classification task to see how certain events are tied to the words themselves. For example, consider, a simple two class punctuation task, where the only class choices are period/no period. In this case the LM will train on a large amount of words and will most likely never come across the word "I" followed by a period, since it is very rare for this word to end a well−formed sentence. Conversely, a forward observation (when allowed) of the word "I" is a strong indication that a sentence boundary exists before this observation, since "I" is frequently used to start sentences. After learning this, the LM will output a very low posterior probability for a sentence−ending punctuation mark (i.e., "." , "?", etc.) if the word "I" is seen as a test case, and performance should be relative high, based on word knowledge alone. But other words are not as obvious, and it is these cases for which the prosodic modeling is useful.

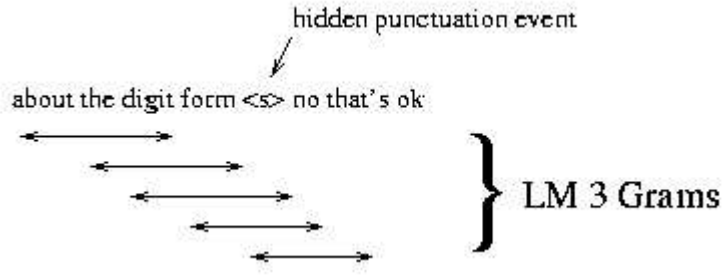


Figure 2.5: Words are observations to the LM which attempts to predict hidden events such as punctuation

The LM for punctuation is a hidden−event N−gram model of the type used in (Shriberg, et al., 2000)**.** Word and boundary type sequences are modeled by a backoff

trigram model, trained in a supervised fashion from annotated training data. (Higher−order N−grams did not perform better, due to lack of sufficient training data.) In testing, the N−gram is interpreted as a hidden Markov model in which the boundary types are treated as hidden states, and the words as observations. The forward−backward algorithm for HMMs is used to recover the best boundary types as well as their posterior probabilities.

The hidden−event LM (A. Stolcke, et al., 1998) used in predicting sentence boundaries and disfluencies comes in two variants. In one case, all events, including the unmarked word boundary type, are modeled by special tags occurring between words. The second variant does not represent unmarked boundaries by tags, and models them implicitly by the absence of a tag. The latter type of LM captures more words in the scope of an N−gram, and was found to work better in experiments were both past and future words were used to predict events (forward−backward computation). However, in "past−only" prediction, the LM that represents the unmarked case with an explicit tag was found to give better results; the "implicit unmarked" LM never predicts a marked event (sentence boundary or disfluency) in that case. This could be a subtle side−effect of the way probabilities are smoothed by back−off in the LM, and needs further investigation.

The LM was used in Tasks 1,2, and 4, as these are the tasks that do not exclude any data. Tasks 3,5 only look at a non−contiguous subset of words, and were therefore not compatible with our LM which expects a continuous stream of words.

2.6 Prosodic models

2.6.1 Features for prosodic classifiers

The processing steps discussed in Section 2.6 yielded stylized pitch values for all the regions of interest on a particular channel. In order to extract useful features for each word, it was necessary to line up the word boundaries with this frame level pitch file, and pull out the appropriate values for each word. Figure 2.6 graphically explains this step of lining up word boundaries with computed pitch values. A script systematically selected

all the frames that fell within a word's start and end times and stored these pitch values in a large database, along with the word and time boundaries. Once frame level pitch samples had been separated into the appropriate word bins, pitch values are considered on a word by word basis, thereby creating a word−level feature granularity.

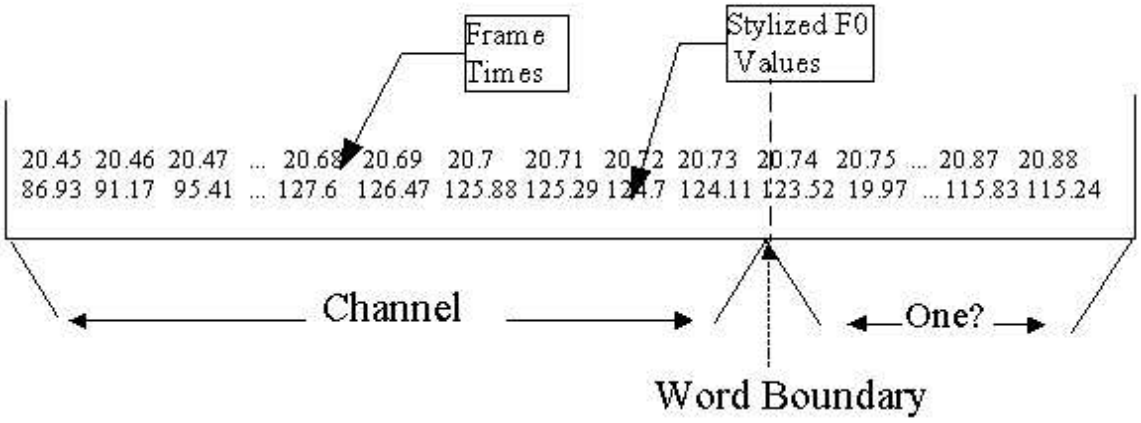


Figure 2.6: Stylized F0 values (in Hz) are matched up with word boundaries. In the case illustrated here, the words "Channel" and "One" are treated as time bins and are given allocated their frame−level pitch values (stylized and raw) according to their time boundaries

Feature extraction regions

Along with features for the current word, the prosodic classifier also had access to previous and future word features, along with prior and future boundary features. A toy example is shown below, where the word "you" precedes the current word "bet" and the period following "bet" resides in the forward boundary:

you		bet	.	I
P_word	P_boundary	C_word	F_boundary	F_word

Tasks 1,2,3 and 4 examine the effectiveness of an on−line classifier (which has only previous feature information) and only use the features to the left and including the current word. Feature names include a prefix (F_, C_, P_) to indicate relative position to the current word, a base (i.e., PAU_DIR), and a possible suffix (_R, _Z, _N) to indicate

the normalization method, where necessary.

What follows is a description of the methods and rationale behind the extraction of a number of different types of features. Before calculating these features one intuitively has an idea of the relative importance of the features, either from prior experiments or theoretical studies, but ultimately the decision tree classifier (to be described below) will use a number of iterative algorithms to decide on how useful these features are in various classification tasks. With this idea in mind, many features are computed, some of which may be considered less useful than others, and allow the learning algorithm to sort through the possible combinations, rather than betting against any particular features *a priori.*

2.6.2 Pause and vowel duration features

Pause and vowel duration features are both extremely useful sets of features. Pauses indicate breaks in prosodic continuity that may indicate boundaries between sentences. Vowel durations are also important as they provide a way to measure speaking rate, which is very useful since speakers' words generally start fast and words near semantic boundaries, such as sentence ends, are often drawn out. As Chapter 4 will show, these feature sets are crucial in various tasks.

Vowel durations (VOWEL) were taken from forced alignment or ASR outputs and normalized by Switchboard vowel statistics. Vowel durations were chosen over phone durations because they are more directly contribute to speaking rate than consonants and they are likely to be more robust against mistakes from ASR. Vowel and vowel−triphone (TRIVOWEL) durations were normalized against Switchboard statistics because there was much more data in this corpus, although future MR statistics can be computed, and results can be expected to improve. For this project we use raw vowel duration, a normalized version based on ratio of value to mean (_N), and a z−score (_Z), which also incorporates second order statistics in normalization.

As mentioned above, pause duration features (PAU_DIR) can be very good indicators of events such as sentence ends. If, for example, F_PAU_DIR, the time between the current word and the next word is very large, there is a good chance that this

boundary should have a period, question mark, or other sentence−ending label. Because of this reason, this feature was used extremely often when forward features are available. As Chapter 4 will show, this feature was also very important in predicting times at which the current speaker is interrupted; there is often a large delay between the current word and next word when this event occurs, especially if the speaker decides to abandon his sentence.

2.6.3 F0 features

F0 feature were calculated over a variety of different frame ranges for words, windows, and segments. These features provide the decision trees with a plethora of tonal information for each word. Pitch movements are also very good indicators of events such as sentence ends, as speakers generally start utterances in a high pitch range and drop low as their words come to an end. These features are also used to examine if speakers create certain prosodically favorable situations for other speakers to jump in (Task 3) and whether those who do jump in altered their prosodic word characteristics from the case when starting an utterance in silence (Task 4).

Local range features:

These features computed the minimum, maximum, mean (MIN, MAX, MEAN) and last (LAST) F0 values for each word position, excluding values which are unvoiced (no F0) or halved or doubled. They were then normalized by the baseline F0 values computed in the LTM model using a linear difference (DIFF) , log difference (LOGDIFF), and log ratio (LOGRATIO). Prosodically, these numbers provide a picture of the overall position of the word with respect to the computed baseline. In order to provide some measure of robustness of against poor word boundaries or short words which may not have data samples over which to calculate the above F0 features, also calculated are MAX/MEAN/MIN values for a window of F0 values which begins at the last frame of a word and stretches backwards N frames, where N={10,20,50,80,100}. If the last F0 value was undefined, the last F0 value from the window was chosen. This window can stretch past the current word and into the previous word if no good values

are available. Figure 2.7 indicates the regions over which these two sets of features operate. Note that in this case, LAST is not defined for the word, and the last value for the 20 frame window is below the baseline (assumed to be halved), so a longer window is necessary (50, 80 or 100) to capture the last F0 value.

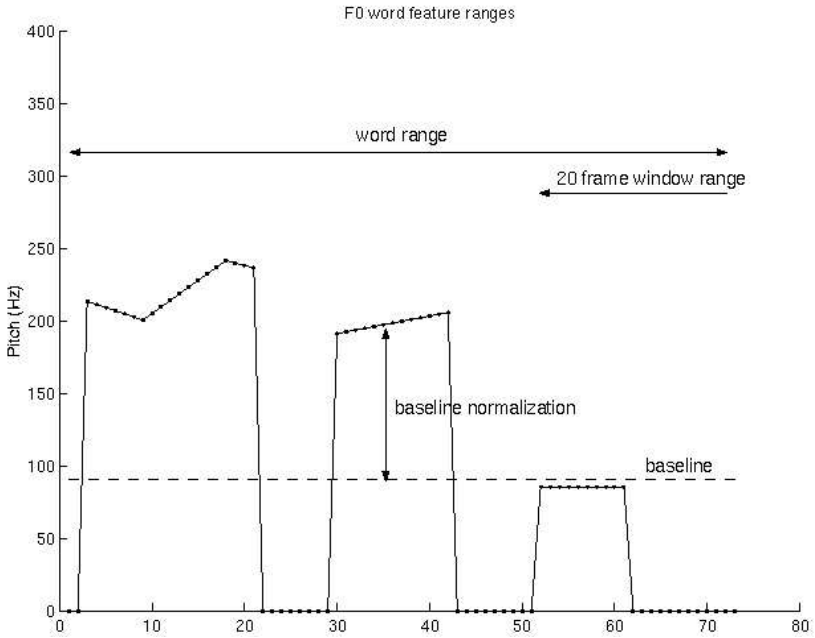


Figure 2.7: F0 feature ranges and baseline normalizations for the word "congratulations". Note that the the region between frames 50 and60 is below the baseline and will be considered as halved pitch data, and therefore discarded. Because of this, the 20 frame window shown above will not have any useful values, and a larger window is necessary to encapsulate the prosodic information towards the end of this word.

The stylized raw pitch values lend themselves to direct pitch slope calculations, since the fitting algorithm uses line fits. The last slope in a word (LAST_SLOPE) and in the window (LAST_SLOPE_WINDOW) are used as features that indicate prosodic change locally.

Global range features

In order to provide for a supra−word context by which one could measure pitch movement in an utterance, segment−level pitch statistics similar to those computed

between word boundaries were also computed. Namely, MIN/MEAN/MAX F0 values were computed over an entire segment and then compared to the local range features discussed above. These segment level F0 feature values provided a context which is more local than the channel−wide baseline values. In other words, prosodic differences between local values such as the mean word F0 value and the mean segment F0 can give an indication of how far a word strays from the average pitch in a segment. Segmental F0 values were themselves normalized by baseline F0 values in the same manner as the word feature in discussed above.

Local pitch movements

In addition to the slope features, which quantify interword change, it is also useful to measure the amount of change between words. For this purpose we calculated the differences (log difference and log ratio) of multiple features across word boundaries. A multitude of differences across word boundaries can be computed. The main features which were of interest for this project include the difference between the current word's last slope or F0 and the next word's first slope or F0 (CF_DIFF_LASTSLOPE_F−FIRSTSLOPE or CF_DIFF_LASTPWLWORD_F−FIRSTPWLWORD) or the difference between the current word's first slope or F0 and the previous word's last slope or F0 (PC_F0K_DIFF_P−LASTSLOPE_C−FIRSTSLOPE or PC_F0K_DIFF_P−LASTPWLWORD_C−FIRSTPWLWORD).

Distance features

Distance features are related to those from the global range set in that they give an indication of how a speaker's prosodic information is changing with respect to the segment min or max. Instead of normalizing the current word's F0 values by those of the segment, distance features measure the time between the start of the current word and the min (C_DIST_SEGPWLMINLOC_WORDSTART) and max (C_DIST_SEGPWLMAXLOC_WORDSTART) of the segment. These time differences help describe how close the current work are to the prosodic peaks and valleys of the

segment.

Octave error features

As mentioned above, the pitch tracker does not always perform within the normal pitch range of a speaker. All pitch trackers are sensitive to noise along with harmonic errors that result in doubling or halving of the true pitch. In addition, speakers may go into vocal fry (i.e., "creaky voice"), a phenomenon which may correlate with our boundaries of interest. For these reasons, the percentage of frames which are in halving and doubling mode for a given word were included in our feature set.

2.6.4 Speaking rate features

Speaking rate is an important facet of prosodic analysis. Certainly speakers experience rate changes over their utterances, usually starting quickly and drawing out their utterance's final words. Because of this, it is important to have some measure of speaking rate. In this study, we measured how fast a word is uttered by counting the numbers of vowels from the start of a spurt to the end of the current word and dividing that number by the total time from the beginning of the utterance to the end of the word, excluding pauses. These vowel averages were then reset at spurt onsets.

2.6.5 Energy features

Similarly, speakers tend to start utterances loudly and taper off with time. Energy features were used to encapsulate a speaker's loudness. The `get_f0` package computes frame level RMS energy values, and from these values the min and max RMS values over an entire word were computed. Two sets of energy features were computed −− those over the voiced frames and also minima and maxima for all the frames. These values were normalized by channel−wide RMS means that account for microphone gain, inherent speaker loudness, or other variables. Normalizations were computed in three ways: raw, ratio to mean for that channel, and z−score based on the mean and standard

deviation over that channel. The set of energy features used in this study were derived from close talking microphones, but it is not completely clear how nearby speakers affected local energy statistics. Preprocessing of the RMS values via an energy separation technique could improve the quality of these features significantly, but that aspect was not investigated here.

2.6.6 Nonprosodic features

Punctuation features (target classes):

These features are our target classes, and thus "hidden" in testing, but their role in model training makes reliable annotation of these events critical. Because of this, hand labelers carefully annotated the entire collection of 32 meeting transcripts accordingly. Table 2.3 describes the punctuation marks added or edited by the annotators.

Punctuation Mark	**Meaning**	**Example**
.	End of complete sentence	Yeah, and use that**.**
==	End of incomplete sentence	Yeah but ==
Q	End of complete question	What are we collecting here?
Q==	End of incomplete question	Is that what you're ?==
−	Disfluency	Yeah, I'm − I'm not quite sure what I'm talking about.
D==	Disfluency ends incomplete sentence	We were th− D==

Table 2.3: Punctuation added and edited by annotators(S. Bhagat et al, 2002)

In the case of automatically derived words, punctuation from real transcripts was merged with words obtained from ASR. This was done by first aligning hypothesized and reference words using a distance metric based on phonetic similarity, a method that can deal with fairly high word error rates. Event labels for sentence boundaries and disfluencies were then transferred to corresponding locations in the hypothesized word

string, and served as reference labels for event scoring in ASR output.

2.6.7 Overlap features

Overlap features were provided to trees in tasks that are not trying to predict overlap−related classes themselves. Overlaps were marked automatically via a number of processing steps and then recorded in our database. In order to mark overlaps completely, a round robin approach was taken, where each speaker was considered the foreground speaker against the background speech of all the rest of the speakers, as in Figure 2.8. A

	Turn 1	Turn 2	Turn 3
Speaker 1	Foreground	Background	Background
Speaker 2	Background	Foreground	Background
Speaker 3	Background	Background	Foreground

Figure 2.8 Overlap processing in the three speaker case. Each speaker has a turn to be the foreground speaker while the rest of the speakers are considered background speech

script systematically considered each word of a particular speaker and then searched through the rest of the speakers to check if another word was spoken on a different channel at some time over the duration of foreground (current) speaker's word. If any other speaker(s) talked during a word, the overlap is noted in an intermediate transcript, and the IN_OVERLAP feature for the interrupting word was set to one. In addition to this overlap feature, other features indicate when overlaps begin and end as well as how many speakers are involved in the overlap. These features were used to train classification trees for several overlap prediction tasks, which will be discussed in more detail in Chapter 4. For calculating and marking overlaps it was useful to create speech segments that are based on acoustic information alone, and are therefore independent of punctuation events and true word knowledge. With this in mind, overlap rates are also calculated on a per *spurt* basis, where a spurt is defined as a region of speech that contains pauses of no more than 500ms within its boundaries Chapter 3 discusses in more detail the nature and pervasiveness of overlaps in the MR corpus and how it compares to other corpora.

2.6.8 Lexical features

The prosodic features described above were all computed from raw pitch and energy features, along with alignments from ASR. Along with these measurements, lexical features were also collected. An important goal in the inclusion of these lexical features was to automatically isolate and label certain words that belong to conversationally important categories, such as backchannels, coordinated conjunctions, filled pauses, and discourse markers. Table 2.4 lists lexical categories and member words. This set of features was calculated from the transcripts (ASR or manually created) and was useful in assessing the importance of these categories in our experiments. For example, backchannels such as "uhhuh" were good indicators of overlap, since backchannels are, by definition used as conversational feedback . Words were labeled as one of these features via a simple heuristic designed to capture the most frequent cases. Because this method uses a word lookup table, it is not always correct. Certain words, such as *right*, may be used in contexts other than backchannelling, but overlabelling is a small charge compared to the manual effort it would take to mark instances of true backchannels. Future work could involve manual annotations of these events.

Lexical Categories	Entries
Backchannel*	*yeah, okay, right, uh, oh, uhhuh*
Discourse Marker	*i don't know, i think, i mean, you know*, so**
Filled Pause	*uh, um, mm*
Repeat	*i i i, the the the, i i , the the* , etc.
Coordinated Conjunction*	*and, but, because*

*Table 2.4 Lexical Categories and their member words. Words or categories denoted with a * must follow a sentence boundary*

2.6.9 *Contextual features*

Finally, a number of features based on a speaker's identity were recorded as

they may correlate with events of interest. For instance, a speaker's name could be a useful feature since some speakers are more likely to interrupt while others are more likely to stay quiet. Similarly, a sociolinguist would argue that gender and cultural background play prominent roles in overlap and interruption modeling and therefore gender and a speaker's English proficiency are recorded as well, despite that in order for these statistics to be significantly useful, a large amount of speech from males and females along with both American and non−American speakers would be required. In addition, a speaker's degree of American English[1] proficiency often very strongly affects recognition performance (see more detailed results in Chapter 3). Therefore word boundaries, which were derived either directly from ASR results, or from alignments which were trained on American English speakers are affected by recognition results. If word boundaries are unreliable, then pitch features will also be error−prone, since they strictly rely on the ability of placing pitch values in word bins. Therefore American English speaking proficiency is in some way a useful metric to determine the integrity of these features.

2.6.10 Prosodic trees

Once all the aforementioned features were tabulated, decision tree classifiers, were used to train models for varying tasks, as they were in related previous studies (Shriberg, et al., 2000). The trees considered a number of observed features X and predicted a an event E , by outputting the posterior probability $P(E|X)$.

The IND software package (W. Buntine and R. Caruana, 1992), which uses CART−style decision trees (L. Breiman, et al., 1984) was used in our experiments. The package analyzed the training data and conducted a variety of tests using the given feature sets, which in our case are the variety of features discussed above. In finding useful features, IND uses cost complexity pruning, a measure of the resubstitution error estimate of a feature set, further penalized by the size of the tree. The result was a

1 The distinction between English and American English proficiency is not a trivial one −− as ASR results indicate. Since these features are self−reported via form questions when a speaker first speaks at a meeting, some speakers reported that they are native speakers, while their proficiency lies in British or Indian English, and recognition results strongly indicated this disparity.

decision tree with class probabilities at decision nodes and decisions on class membership at the leaf values, as show in Figure 2.9. IND is extremely versatile in that it can deal with missing data, and smoothing and pruning algorithms are available in order to avoid overfitting on training data. Finally, the ability to easily add and manipulate feature sets along with the readability of decision trees, make this probabilistic classifier particularly attractive for use in this study.

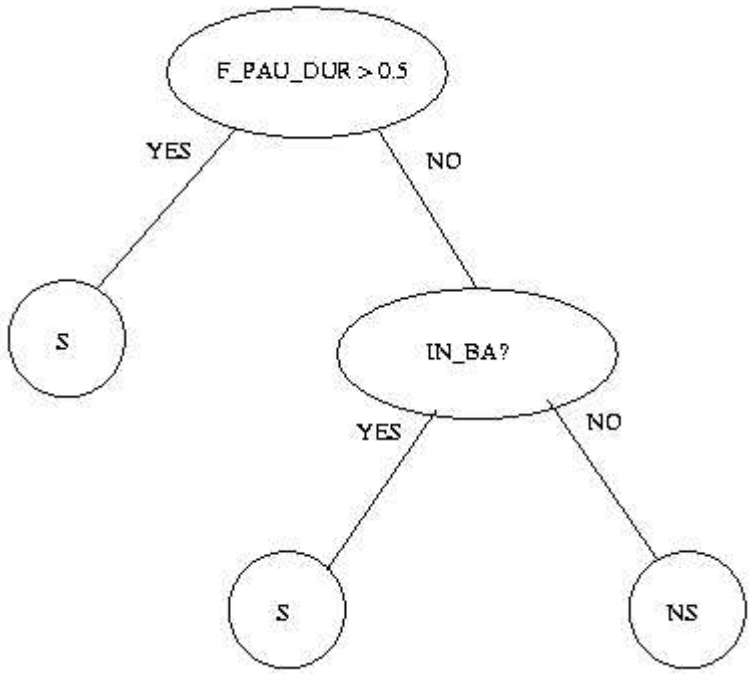


Figure 2.9: A toy example of a decision tree. Words with F_PAU_DIR > 0.5 s get or Backchannels are classified as sentence boundaries.

For many classification tasks, it was useful to downsample the data for a number of reasons. First, some tasks were highly skewed towards one event. A good example of this case is a two−class sentence/non−sentence classification task, where posterior probabilities give the likelihood that the observed word is followed by a sentence−ending punctuation mark. From preliminary observations, it was noticed that about 90% of words were not followed by punctuation marks, so training on raw distributions would have skewed posteriors strongly to the "non−sentence" class. Downsampling also served as a type of normalization across speaking variations encountered across different

meeting and speakers. In interruption prediction tasks, for example, certain speakers were more likely to interrupt and to be interrupted, and certain meeting types such as the "Bmr" set discussed above had more interruptions than other meeting types. Thus downsampling serves as a type of normalization across these different factors.

2.7 Model combination

We also combined the predictions of a decision tree with the Language Model HMM to construct a combined classifier, by converting the decision tree probabilities into additional HMM observation likelihoods Downsampling is relevant in this process as well, since this operation allows for direct integration of the prosodic model with the LM, where the class posteriors are directly proportional to the likelihoods. This is shown as follows (E. Shriberg, R. Bates, A. Stolcke, 1997):

$$
\begin{aligned}
P(D|W,X) &= \frac{P(D|X)P(W|D,X)}{P(W|X)} \\
&\approx \frac{P(D|X)P(W|D)}{P(W|X)} \qquad (1) \\
&= \frac{P(D|X)P(D|W)P(W)}{P(W|X)P(D)} \\
&\propto \frac{P(D|X)P(D|W)}{P(D)} \qquad (2)
\end{aligned}
$$

Where P(D) is the probability of a hidden boundary event (i.e., a disfluency or punctuation mark) , W are observed words and X is the feature array.

Step (1) above is true, because it is assumed that word probabilities are independent of their prosodic features conditioned on the event, since none of these features depend on word identity. The proportionality in (2) holds because when considering posteriors for event D, one may drop all the terms which are independent of this event. Finally we may drop P(D) from Eq. 2 since all the event probabilities are equal in the downsampled case, and the proportionality is met.

In experiments where the combination models were used, the LM was used as an HMM but likelihoods were also computed for the event states using the prosodic decision

trees. They then were factored into the computation of event posteriors. As results in Chapter 4 will show, this model combination generally does better than either model on its own.

3. ASR results and observations on overlap

This chapter discusses the results of Automatic Speech Recognition (ASR) on the Meeting Recorder (MR) corpus, along with a number observations on the number and nature of overlaps in the corpus. While the primary goal of this project does not include building, tuning, or spending a considerable amount of research time on the advancement of a MR word recognizer, ASR performance has tremendous implicit importance on the effectiveness of our various classification tasks, since word time boundaries are derived either from forced alignments, or from automatic transcripts. Similarly, overlaps are undeniably important as far as meeting understanding is concerned; a good model of overlap and turn taking are of crucial importance if machine participation or summarization is realized.

3.1 ASR results

As stated above, since the main purpose of this research is to automatically extract prosodic features from meetings and subsequently use these features in numerous classification tasks, much effort was not put into optimizing a word recognizer. Also, because there was not enough meeting data to train recognition models, we default to the Hub−5 training set. Specifically, the word recognizer used in this project was a stripped down version of SRI's large−vocabulary conversational speech recognizer used in the March 2000 Hub−5 evaluation. As stated above**,** the system performs vocal−tract length normalization, feature normalization, and speaker adaptation, using all the speech collected on each channel (i.e., all speech from one speaker per meetings, excluding crosstalk). The acoustic model consisted of gender−dependent, bottom−up clustered (genonic) Gaussian mixtures (V. Digalakis, et al., 1996). The Gaussian means are adapted by a linear transform so as to maximize the likelihood of a phone−loop model, an approach that is fast and does not require recognition prior to adaptation. The adapted models are combined with a bigram language model for decoding. Also, the acoustical models and the Language Model used in these experiments were identical to those used in the Hub−5 domain. Most notably, the front−end assumes a telephone channel and

downsamples the 16KHz signal to 8KHz accordingly. The language model contains about 30,000 words and is trained on a combination of Switchboard, CallHome English, and Broadcast News data.

Preliminary results of ASR performance were noted in (N. Morgan, et al., 2001), but since then recognition experiments have changed significantly. First, the results previously reported were only conducted on 8 meetings but now our data collection has expanded significantly and 32 meetings are available. Second, and more importantly, the segmentation techniques employed in previous recognition experiments used time−synchronous boundaries across channels, as opposed to current segmentations, which used different boundaries for each speaker. This change in segmentation techniques is critically important since, as Figure 3.1 shows, old segmentations created situations where speech was surrounded by a large amount of silence, especially for short utterances such as backchannels. This silence has the potential to cause a large amount of insertions, especially when a nearby speaker is talking and/or when the speaker is wearing the lapel microphone, which is poorly localized. Current segmentations used the HMM presegmentation technique discussed in Chapter 2, and were then modified by human transcribers where appropriate. Table 3.1 compares the effect of this segmentation type change for a particular speaker, who used the a lapel microphone in multiple meetings.

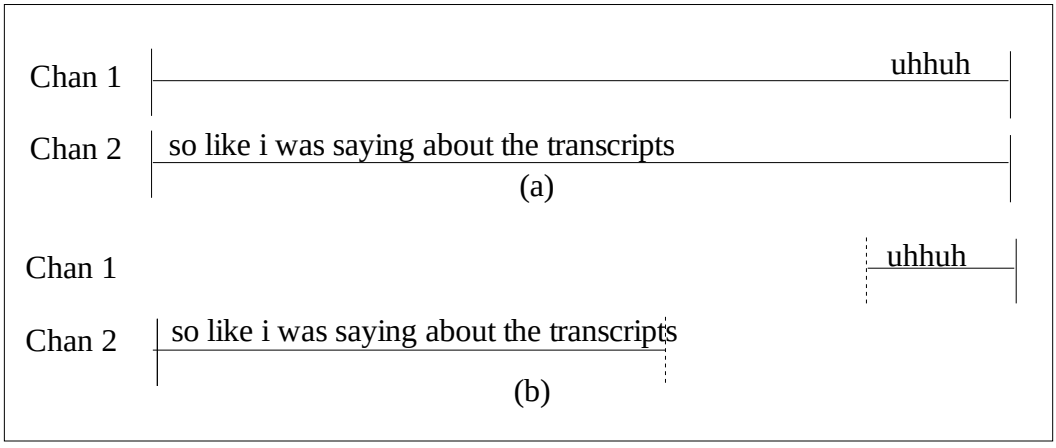


Figure 3.1: Differences in segmentation types. (a) shows time−synchronous segmentations across channels . Note than Channel 2 speech can be heard in silence of Channel 1, causing insertions and that this problem is avoided in segmentation (b)

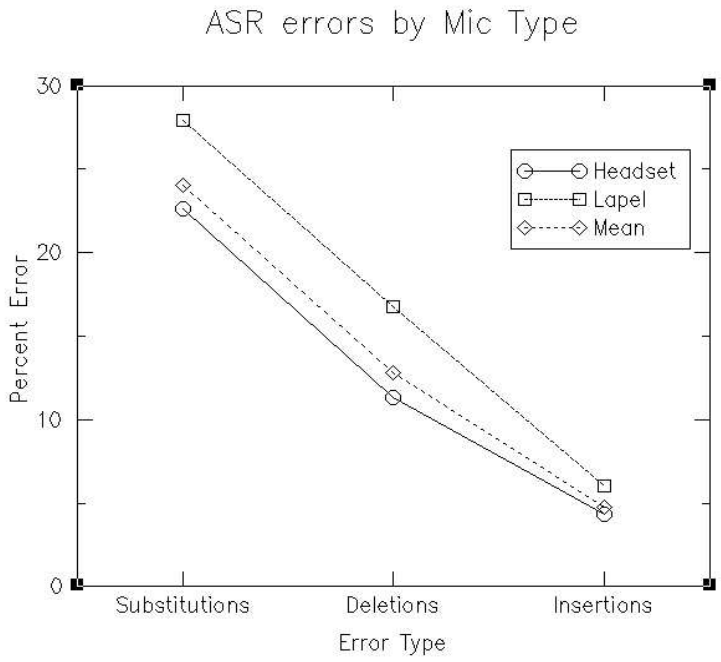


Figure 3.2: Error rate comparisons of lapel/Headset for one speaker

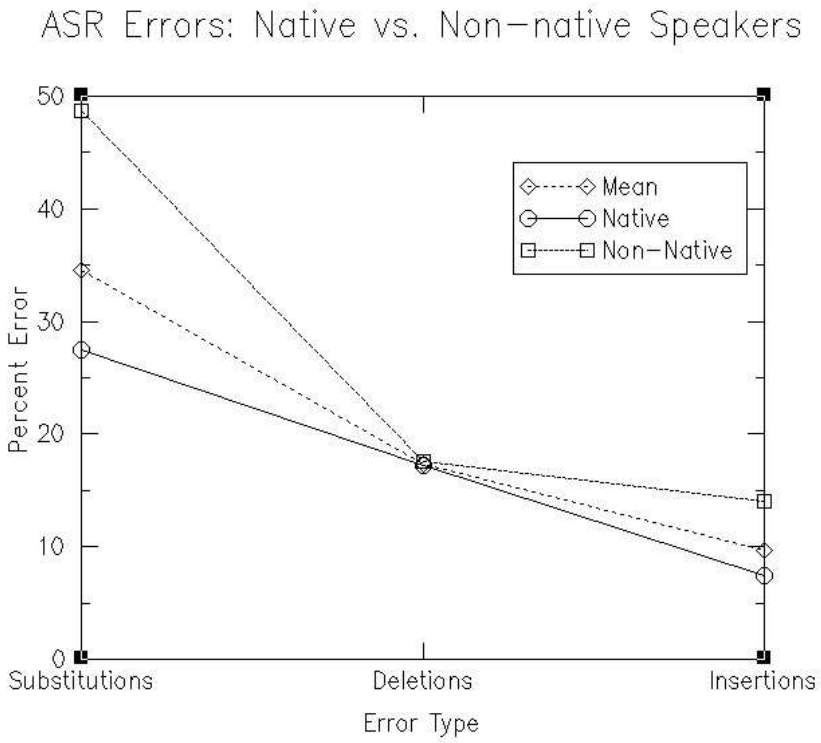


Figure 3.3: ASR rates for native and non−native speakers of more than 20 words per meeting.

From the table, it is clear that using channel−specific segment boundaries strongly decreases the rate of insertions and thereby brings down word error rates as well. Tighter segment boundaries are therefore used for our recognition and classification experiments, since best possible automatically generated word−time boundaries are desired.

Meeting	Number of Words	Substitutions	Deletions	Insertions	WER
Bmr006−c0	1132	33.1	9.3	57.7	100.1
	1175	**27.5**	**17.5**	**6.6**	**51.6**
Bmr007−c0	500	35.8	31.0	45.0	94.7
	517	**30.8**	**19.5**	**8.9**	**59.2**
Bro004−c0	701	29.4	6.7	30.7	66.8
	742	**28.7**	**8.9**	**5.8**	**43.4**

Table 3.1: Comparison of ASR results for channel synchronous and asynchronous segment boundaries for a speaker wearing the lapel microphone over multiple meetings. New (non−synchronous) results are in bold.

Table 3.1 indicates how insertion rates are extremely high for speakers using the lapel microphone with the same segmentation boundaries across channels. Even with better segmentations, however, performance on the lapel microphone still suffers tremendously. For this reason, this microphone type was not used in favor of the close talking microphones in later meetings. Figure 3.2 shows ASR results for a common speaker who used both types of microphones in multiple meetings. Lapel performance is clearly poorer than the mean over all meetings for this speaker. Interestingly, in this case the better segmentations help insertion error rates, but overall lapel performance is still significantly worse than that of close−talking microphone.

Tighter segmentations are extremely helpful, since they block out other speech in long regions of silence, but many other factors affect recognition performance across speakers. In particular, ASR quality degrades substantially when it encounters non−native American English Speakers, primarily because acoustic models are not available for the multitude of variants of American English accents. True to its name, ICSI's meetings considered in this work included participants from England, Spain, Germany, Finland, India, the Czech Republic, Israel, and Belgium, providing a truly complicated

scenario where training on the variety of international accents is simply infeasible. Figure 3.3 presents different error rates compared across speakers with native and non−native American English background.

Despite the fact that no models were trained on meeting speaker data, performance for this system was still relatively high. For native speakers, the overall WER was 45.2 % , representing about a 5% relative increase over a comparable recognition system on Hub−5 telephone conversations. This affirms our use of this system, and is particularly impressive considering no meeting speech was used to train the models, and neither the front end nor the language model were adjusted for this data or the use of a close talking microphone instead of a telephone. To some extent this may occur because speech in meetings is not too dissimilar to telephone speech, in that at a very high level, the pronunciation and language in multi−party conversations does not stray too far from speaking patterns in two−party telephone sessions. This is very interesting considering the familiarity and use of gesture between meeting participants.

3.2 Observations on overlap

Observations on the amount and nature of the overlaps in the MR corpus are reported in this section. Perhaps the most defining aspect of the corpus is the abundance of speaker overlap and turn−taking across the meetings. Indeed, the combination of a casual atmosphere, face to face contact, and speakers' familiarity with one another combines for a setting rich in speaker overlap. In quantifying the relative importance of overlaps in the MR corpus as compared to other corpora, Table 3.2, based on a similar table in (E. Shriberg, A. Stolcke, D. Baron., 2001, *Observations*)**,** shows percentage of spurts in overlaps, both including and excluding backchannels for different meeting types, along with the Call Home and Switchboard corpora. Backchannels are excluded as these are not indicative of turn−taking interruptions. Spurts, as defined in Chapter 2, are units of speech that uninterrupted by pauses longer than 500ms.

In terms of overlap, the disparity across meeting types is significant. Some meetings, such as those belonging to the Bmr group have quite a lot of overlapped words and spurts, while other meetings, such as the Bro group, are not quite as overlap

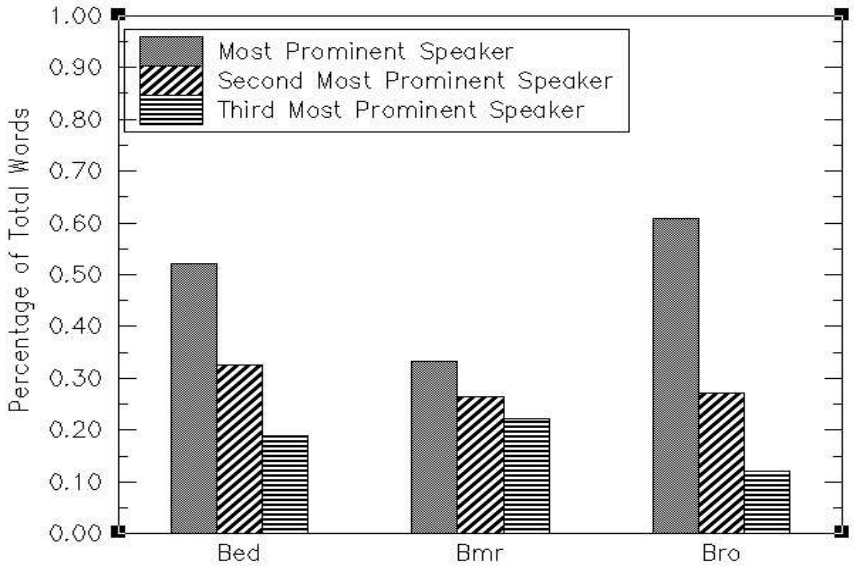


Figure 3.4 Average distribution of words across top 3 speakers in each meeting type

intensive. Bed meetings lie somewhere in between. As compared to the phone conversation corpora, Bmr meetings have more interruptions per word and per spurt than CallHome or Switchboard, when including backchannels in calculations. These results make sense since the meetings are conducted in a casual manner, with familiar participants.

	Meetings			Phone Conversation	
Backchannels	Bed	Bmr	Bro	CallHome	Switchboard
Included					
words	13.5	21.7	10.5	11.7	12.0
spurts	*44.1*	*63.2*	*36.5*	*53.0*	*54.4*
Excluded					
words	8.43	16.4	6.12	7.9	7.8
spurts	*28.9*	*31.3*	*22.5*	*38.8*	*38.9*

Table 3.2: Relative frequencies of overlapped speech in different corpora. Values are given in percentages of total number of words (in plainface) and total number of spurts (in italics)

Qualitatively, overlap rate variability across meeting type is fairly clear, as Bro meetings tend to be more of a seminar style, where one speaker generally leads the discussion and clearly has more control over the meeting than any other of the

participants. In contrast, Bmr meetings tend to be more democratic in nature, without any clear meeting leader. Figure 3.4 shows percentage of total words, spoken by the top three speakers in each meeting type. The numbers do not necessarily reflect any particular speaker's importance across meeting types, but rather show the distribution of words across the three main speakers.

From the figure it is clear that, on average, one speaker generally dominates the meetings in Bro, while the distribution is fairly even in the Bmr meetings. Bed meetings fall somewhere in between; while not quite as one−speaker dominated as the Bro meetings, on average, one speaker has over half the words in any given meeting. These statistics do not give any explicit information regarding the nature or amount of overlap in the meetings, but it is useful to know which meetings have more active speakers contributing as this inherently affects the potential for overlap.

4. Results

In this Chapter, a number of experiments conducted on the Meeting Recorder corpus are examined. The first set of experiments are punctuation oriented and involve detection of disfluencies and sentence punctuation. The second set of tasks involve dialog phenomena. These involve the prediction of interruptions and turn taking in various contexts.

Where appropriate, a language model is used to provide a baseline metric that allows us to establish the added value of prosodic features beyond word knowledge alone. Language models are only included in experiments where all the words are available. Some experiments, by their definition, exclude many words and therefore do not lend themselves easily to LM training, which requires a stream of contiguous words in order to operate correctly. Similarly, the role of omitting features that look forward in time (either forward boundary or forward word features) is examined, since real−time applications will not have the benefit of future features. Of particular interest is the comparison of the LM and the prosodic−feature based decision tree *vis a vis* this backward−only model, and the potential degradation because of this feature reduction in each case.

The experiments discussed in this section are evaluated using two metrics: accuracy and efficiency. Accuracy is defined as the percentage of cases in which the class with the highest posterior probability is correct. This value is a simple percentage which counts correct descisions and is most often used here to compare variations of the same experiments, as the prior distributions are the same. Efficiency, on the other hand, is a measure of the reduction in class entropy achieved by the classifier relative to the prior distribution (the raw distribution of the classes in data). Formally the efficiency is defined as follows:

$$\frac{H(p_0)-H(p)}{H(p_0)}$$

Where p_0 denotes the prior distribution, p the estimated posterior distribution, and H is the entropy. The latter metric is particularly useful because it normalizes the reduction in

entropy by the entropy of the prior distributions, which allows us to directly compare the results of multiple experiments regardless of prior class distributions, and by extension, regardless of the inherent difficulty of task. An efficiency of 1.0 (100%) implies a perfect classifier, whereas a zero efficiency characterizes a classifier that does no better than chance (i.e., posteriors are equal to priors).

4.1 Task descriptions

Results for a number of tasks are reported. Some of these experiments are extensions of earlier work (M. Mast, et al., 1996, P. Heeman et al., 1997, A. Stolcke, et al., 1998, Shriberg, et al., 2000) that concentrated on monologue corpora such as Broadcast News, or telephone conversations from Switchboard or CallHome. These tasks can be grouped together as punctuation classification or disfluency modeling experiments. The extension of these tasks to the meeting domain is an important goal; as research on meeting analysis progresses beyond word recognition and towards higher level understanding, punctuation classification becomes a necessity. The remaining tasks involve the prediction of dialog events. These tasks use prosodic and lexical cues to discriminate turn−taking and interruption events, and are absolutely critical for high level understanding tasks.

	Name	Description	# classes	All Words?	LM?	ASR?
Punctuation	s−ns	sentence/non−sentence	2	X	X	X
	s−di−n	sentence/disfluency/neither	3	X	X	X
Punctuation and Dialog Act	s−q	Question/declarative sentence	2	last word in sentence		X
Interaction	Jump−In Points	Word boundary a point of interruption for a bg speaker?	2	X	X	
	Jump−In Words	First word of spurt in silence or someone else's speech?	2	first word in spurt		

Table 4.1.1: Description of 5 tasks discussed in this chapter, along with data inclusion, number of classes and ASR/LM invocation

Five tasks are examined. Table 4.1.1 describes the experiments in terms of their

data usage (is every word examined?), whether ASR experiments are reported, and if an LM is invoked. From the table, it is seen that LM experiments are only run on experiments that contain all the words. Another observation from Table 4.1 is that ASR experiments were not conducted on dialog experiments. The difficulty in these cases is determining how to score and label true interruptions in the ASR domain. For punctuation one can simply merge punctuation from manual transcripts to ASR words. In the case of interruptions, finding true cases of overlap are not as easy because it would require true knowledge about interruptions and simply comparing word times for one speaker's ASR words to the other speakers (as done in the manual overlap calculation discussed in Chapter 2), is not sufficient, since word errors may cause phantom overlaps, or miss overlaps altogether in the case of word deletions.

4.1.1 Task 1: Predicting sentence boundaries (s−ns)

This is a two−way classification task where fluent boundaries are distinguished from sentence ends. Disfluencies and incomplete sentences are included in the fluent boundary class, despite possible inherent prosodic differences between these groups. Similarly, questions are included in the sentence class. These simplifications are particularly useful for segmentation applications where finer details such as disfluencies are not as important as sentence demarcation. The prior of the majority class is also considered the "chance" performance, i.e., the performance if the majority class were chosen at each word boundary. Also clear from the table is the amount of data lost when downsampling is performed. As mentioned in Chapter 2, downsampling evens out distributions by creating N classes, all with the same number of tokens as the smallest minority class. While useful in leveling class priors, much of the data in the larger classes is lost. As discussed in the task results show, downsampled results are reported, which are underestimates of true performance. Table 4.1.2 sums up the prior distributions of the classes.

	Class Priors		Class Tokens		Downsampled Priors	Downsampled Tokens	
	Train	Test	Train	Test		Train	Test
No sentence End							
True Words	**89.12**	90.91	224815	49828	0.50	27446	4980
ASR	**90.12**	*91.25*	*215495*	*47515*	*0.50*	*23631*	*4554*
Sentence End							
True Words	10.88	9.09	27446	4980	0.50	27446	4980
ASR	*9.88*	*8.75*	*23631*	*4554*	*0.50*	*23631*	*4554*

Table 4.1.2 Priors for Task 2. ASR values are in italics, chance in bold.

4.1.2 Task 2: Predicting disfluencies and sentence boundaries (s−di−n)

This task is a three−way classification of all word boundaries as either complete sentence ends, incomplete sentence/disfluent boundaries, and fluent sentence−internal word boundaries. Because the minority class sizes are so small, incomplete sentences and disfluencies are grouped together as one class, despite intrinsic prosodic differences between these two cases. Similarly, questions are grouped with declarative sentence ends. These oversimplifications may result in a loss, as there are theoretical prosodic differences within the individual classes, but since there are so few question, for example, further dissecting the "sentence" class would skew prior class probabilities even more. Though this problem can be partially solved by downsampling, this operation throws away data to match the priors of the smallest minority class, and could result in even greater performance degradation.

Table 4.1.3 shows the priors for each of the classes in the test and train sets, along with the number of tokens considered in each class, and the number of tokens in each class for the ASR and true word cases. From the table it is clear that the fluent boundary class is the most common of all classes, with the *a prior* probability of around 80%.

4.1.3 Task 3: Distinguishing declarative sentences from questions (s−q)

For Task 3, the data is altered quite significantly from the aforementioned punctuation tasks. Namely, the following question is asked: given the knowledge that the

	Class Priors		Class Tokens		Downsampled Priors	Downsampled Tokens	
	Train	Test	Train	Test		Train	Test
Fluent Boundary							
True Words	**78.49**	80.74	197993	44252	0.33	26822	4980
ASR	**80.82**	*80.70*	*193265*	*42020*	*0.33*	*22230*	*4554*
Disfluency							
True Words	10.63	10.17	26822	5576	0.33	26822	4980
ASR	*9.29*	*10.55*	*22230*	*5495*	*0.33*	*22230*	*4554*
Sentence End							
True Words	10.88	9.09	27446	4980	0.33	26822	4980
ASR	*9.88*	*8.75*	*23631*	*4554*	*0.33*	*22230*	*4554*

Table 4.1.3: Class distributions for Task 1, in ASR/manual words, in test and train sets. Italic values indicate the ASR case, while bold font show chance performance in each case.

current boundary is a sentence boundary, can declarative sentence ends (i.e., periods , exclamation points) be distinguished from questions? From observation (D. Jurafsky et al., 1998) and theory, it is known that many questions end with a pitch rise. Non−question ends, on the other hand, are spoken more softly and usually have a marked pitch drop.

Because for this task only locations that are sentence ends are considered, a significant amount of data is dropped, and the LM is no longer used, as it requires all data points for modeling purposes. Table 4.4 summarizes the class distributions for Task 3. Note the tremendous decrease in data points considered.

	Class Priors		Class Tokens		Downsampled Priors	Downsampled Tokens	
	Train	Test	Train	Test		Train	Test
Not Question							
True Words	**89.56**	88.86	24443	4387	0.50	2848	550
ASR	**89.10**	*88.86*	20910	*4014*	*0.50*	*2557*	*503*
Question							
True Words	10.54	11.14	2848	550	0.50	2848	550
ASR	*10.90*	*11.14*	*2557*	*503*	*0.50*	*2557*	*503*

Table 4.1.4: Data distribution for classes in Task 3. Italics indicate ASR distributions, bold values are the chance performance.

4.1.4 Task 4: Predicting Jump−In points

The remaining tasks deal less with punctuation events and are more specific to the dialog events within the MR corpus. These tasks deal in modeling and predicting points of overlap or interruptions, from the perspective of both the interrupter and the speaker interrupted. As mentioned above, ASR versions of these tasks are not available in this report, because of the complexity involved in accurately labeling and scoring these tasks.

Task 4 is a two way classification task that attempts to predict if a foreground speaker will be interrupted by another speaker, given a set of prosodic and lexical features from the foreground speaker's words. In other words, this task will predict which prosodic features could be used by a background speaker in determining a good place to interrupt another speaker.

This task is particularly difficult for two reasons. First, an attempt is made to predict when a background speaker will jump in, (the Jump−In Point), despite not having any access to that speaker's features. Secondly, only the points where a background speaker (or speakers) actually jumps in are known, without knowledge of the places where a background speaker thought about jumping in (found a prosodically appropriate place for an interruption) but did not follow through on the interruption for whatever reason.

Decision trees will attempt to classify the two classes of "jump−in point" and "not jump−in point". These points are labeled in the overlapping scheme described in Chapter 2, where each speaker is considered a foreground speaker against all other speakers. Table 4.1.5 shows prior class distributions for this task. Note the inherent difficulty in this task; over 96 % of the tokens are of the majority class.

	Class Priors		Class Tokens		Downsampled Priors	Downsampled Tokens	
	Train	Test	Train	Test		Train	Test
No Jump−In Point True Words	**96.27**	96.21	242845	52762	0.50	9417	2080
Jump−In Point True Words	3.73	3.79	9417	2080	0.50	9417	2080

Table 4.1.5: Class distributions in train/test for Task 4. No ASR results are presented, as only true words were used in this Task. Bold value indicates chance performance.

One point that should be made about jump−in point calculations is that jump−in points are defined to be within a spurt. In other words, if a foreground speaker is cut off by a background speaker and does not continue speaking within 0.5s , this is considered the end of his or her spurt, and therefore there cannot be a jump−in point here, by definition.

4.1.5 Task 5: Predicting Jump−In words

Task 5 is the final experiment conducted in this report. This task also deals with interruptions, but now the foreground speaker is considered as the trees try to classify the first words of each spurt as either an interruption (starting in someone else's speech) or not an interruption (starting in silence). This first spurt word is called a "jump−in word" if it is spoken while some background speaker is also speaking. From a high level perspective, the following question is being asked: "Is there a prosodic or lexical difference in first spurt words that start in someone else's speech rather than those that start in silence?" From observation the answer to this question should be "yes": speakers who are attempting to talk over someone may start with particularly large energy and high prosodic features.

As in Task 3, only a subset of all the data is considered in this task, namely only the first word in all the spurts. Because of this, the language model is not used, as contiguous word streams are not available. Table 4.1.6 describes data distributions for Task 5.

	Class Priors		Class Tokens		Downsampled Priors	Downsampled Tokens	
	Train	Test	Train	Test		Train	Test
Jump−In Word True Words	**76.09**	74.65	16077	3224	0.50	5052	1095
Not Jump−In Word True Words	23.91	25.35	5052	1095	0.50	5052	1095

Table 4.1.6: Class distributions in train/test for Task 5. No ASR results are presented, as only true words were used in this Task. Bold value indicates chance performance.

4.2 Task 1 results

This section presents a variety of experiments on Task 1, a two class sentence/non−sentence classification of word boundaries. In the example shown below, the word boundaries following the words "know" and "that" are considered sentence boundaries, where as all others are not:

```
do    you   –     i     know   .   what   was     that  ?
 <ns>     <ns>     <ns>      <s>      <ns>    <ns>       <s>
```

4.2.1 All feature regions

The variations within this task, called "cases", include running the experiments on different train/test sets (i.e., real word vs. ASR) and including all versus only past features so as to simulate an online system. Table 4.2.1 shows results for Task 1, including all features, but with variations on the test/train data sets in terms of true versus ASR words.

Case	Forward Features?	Train Set	Test Set	Chance	LM	Tree	LM+Tree
Case 1	Yes	True	True	**90.91** *0.00*	**93.95** *53.12*	**93.41** *39.00*	**94.84** *59.68*
Case 2	Yes	ASR	ASR	**91.25** *0.00*	**91.77** *26.86*	**91.25** *25.62*	**92.62** *31.59*
Case 3	Yes	True	ASR	**91.25** *0.00*	**91.25** 10.56	**92.43** *28.28*	**92.43** *28.28*

Table 4.2.1: Accuracies and efficiencies for three different train/test cases for Task 1, with results in percentages. Accuracies are in boldface and efficiencies are in italics. Chance accuracy based on choosing most frequent class.The column headed by "Forward Features?" shows if features following the event boundary are used.

Except in Case 3, the decision tree using prosodic features does slightly worse than the Language Model, but in Case 2 and 3 the combination model outperforms either individual model. In Case 3, where there is a data mismatch between training on true words and testing on ASR, the LM does not perform above chance whereas the decision

tree yields an improvement. In fact, the decision tree, without any word knowledge at all, does almost as well as the LM in Cases 1 and 2 and outperforms the LM in Case 3. Combining the minority classes consisting of disfluencies and fluent boundaries into one larger minority class causes a decline in performance from the prosodic standpoint because disfluencies and fluent word boundaries have quite different tonal profiles. This crude approximation is taken care of in Task 2, as is shown in Section 4.3. The combination of minority classes may be hurting the decision tree. Nonetheless, performance in these cases are impressive; even with chance accuracies around 90%, Cases 1 and 2 accuracies outperform chance by 4.3% and 1.5% relative using the combination model.

As expected, the introduction of word error rates into the models has a detrimental effect on event classification. Using ASR, the LM accuracy degrades by 2.32%, the prosodic feature tree by 2.31% and the combined model by 2.3%, relative to models trained and tested on true words. In Case 3, where the learning algorithms train on true words and test on ASR words, the individual performance for the LM and decision tree actually improve slightly, but the combined tree does worse than in Case 2. Both models are obviously sensitive to word errors in training, and access to true word boundaries in training appears to be an advantage regardless of testing data set.
The feature usage for the three cases for Task 1 are provided in Table 4.2.2. Feature usage (E. Shriberg, 2000) is a measure that counts how many decisions in which the feature played a role. Features higher in the tree affect more datapoints and therefore have higher usages. Usages over the whole tree sum to 1.0 .

Two striking observations emerge from this table. First, feature usage in Case 1 and Case 3 are identical, which is expected since the trees were trained on the same data set. Secondly, all three examples use following pause durations most heavily, but Case 2 stands out as it uses this feature 88.36% of the time. Following pauses are very good indicators of sentence boundaries, as sentences often have large following pauses. Note that pauses may also indicate incomplete sentences or disfluencies, but these phenomena are included in the minority class, so large following pause durations may cause classification errors in this case. The marked lack of vowel durations in Case 2's feature usage is interesting − it may indicate that these features become unreliable because of

word recognition errors. Figure 4.2.1 shows the trees for the cases discussed above. In order to conserve trees (pun wholly intended), only the top four levels of splits are included. Finally, as the trees below show, when either the current or following word is overlapped by other speakers, decision trees generally point to a sentence boundary, since interruptions often begin when a speaker is almost done. This phenomenon is called "precision timing" (G. Jefferson, 1973).

Case 1		Case 2		Case 3	
Feature	Usage	Feature	Usage	Feature	Usage
Vowel Durations	**48.96**	**Pause Durations**	**92.58**	**Vowel Durations**	**48.96**
C_VOWEL_DUR	22.06	F_PAU_DUR	88.36	C_VOWEL_DUR	22.06
C_TRIVOWEL_DUR_Z	6.77	P_PAU_DUR	3.17	C_TRIVOWEL_DUR_Z	6.77
F_TRIVOWEL_DUR_N	6.77	PP_PAU_DUR	1.05	F_TRIVOWEL_DUR_N	6.77
C_VOWEL_DUR_N	3.46	**Overlap Features**	**7.41**	C_VOWEL_DUR_N	3.46
C_VOWEL_DUR_Z	2.77	F_IN_OVERLAP	4.77	C_VOWEL_DUR_Z	2.77
P_TRIVOWEL_DUR_N	1.88	C_IN_OVERLAP	2.64	P_TRIVOWEL_DUR_N	1.88
P_VOWEL_DUR_Z	1.79			P_VOWEL_DUR_Z	1.79
F_TRIVOWEL_DUR_Z	1.75			F_TRIVOWEL_DUR_Z	1.75
F_VOWEL_DUR	1.71			F_VOWEL_DUR	1.71
Pause Durations	**48.84**			**Pause Durations**	**48.84**
F_PAU_DUR	45.16			F_PAU_DUR	45.16
P_PAU_DUR	3.68			P_PAU_DUR	3.68

Table 4.2.2 Feature Usage in Task 1 a two−class sentence/non−sentence word boundary classification task. for Cases 1,2,3. Usages are given in percentages. Bold is total for the feature type, plain are individual features. C_, F_, P_ are the current, previous and following features, respectively, relative to the current word. Feature definitions are given in the Appendix.

```
**********************
     Cases 1 & 3
**********************
 0.5 0.5 ns s
F_PAU_DUR < 0.175:  0.7673 0.2327 ns
|   C_VOWEL_DUR < 6.5:  0.8678 0.1322 ns
|   |   F_PAU_DUR < 0.075:  0.8769 0.1231 ns
|   |   F_PAU_DUR >= 0.075:  0.5838 0.4162 ns
|   |   |   F_TRIVOWEL_DUR_N < 0.5:  0.4404 0.5596 s
|   |   |   F_TRIVOWEL_DUR_N >= 0.5:  0.6537 0.3463 ns
|   C_VOWEL_DUR >= 6.5:  0.6646 0.3354 ns
|   |   F_TRIVOWEL_DUR_N < 0.3:  0.5101 0.4899 ns
|   |   |   F_PAU_DUR < 0.035:  0.5448 0.4552 ns
|   |   |   F_PAU_DUR >= 0.035:  0.2764 0.7236 s
F_PAU_DUR >= 0.175:  0.1315 0.8685 s
|   F_PAU_DUR < 0.7075:  0.2924 0.7076 s
|   |   C_VOWEL_DUR < 26.5:  0.2612 0.7388 s
|   |   C_VOWEL_DUR >= 26.5:  0.6385 0.3615 ns
|   |   |   P_PAU_DUR < 1.77:  0.7128 0.2872 ns
|   |   |   P_PAU_DUR >= 1.77:  0.2267 0.7733 s
|   F_PAU_DUR >= 0.7075:  0.06411 0.9359 s
|   |   F_PAU_DUR < 1.9135:  0.1266 0.8734 s
|   |   |   C_VOWEL_DUR < 28.5:  0.1081 0.8919 s
|   |   |   C_VOWEL_DUR >= 28.5:  0.4031 0.5969 s
|   |   F_PAU_DUR >= 1.9135:  0.03152 0.9685 s


**********************
       Case  2
**********************
 0.5 0.5 ns s
F_PAU_DUR < 0.105:  0.7874 0.2126 ns
|   F_PAU_DUR < 0.045:  0.7975 0.2025 ns
|   F_PAU_DUR >= 0.045:  0.5753 0.4247 ns
|   |   F_IN_OVERLAP in 2,1,3,4,5 :  0.3953 0.6047 s
|   |   F_IN_OVERLAP in 0 :  0.6377 0.3623 ns
F_PAU_DUR >= 0.105:  0.1647 0.8353 s
|   F_PAU_DUR < 0.4375:  0.4153 0.5847 s
|   |   F_IN_OVERLAP in 2,1,3,4,5 :  0.2723 0.7277 s
|   |   F_IN_OVERLAP in 0 :  0.4738 0.5262 s
|   |   |   C_IN_OVERLAP in 2,1,3,4,5 :  0.3027 0.6973 s
|   |   |   C_IN_OVERLAP in 0 :  0.4988 0.5012 s

|   F_PAU_DUR >= 0.4375:  0.107 0.893 s
```

Figure 4.2.1 Abridged decision trees for Cases 1,2 and 3 for a two−way sentence/non−sentence classification task. Class labels are ns (not a sentence boundary) and s (sentence) and probabilities are listed in that order. Case 1 trains and tests models on true words, Case 2 trains and tests on ASR words, and Case 3 trains on true words and tests on ASR. Decision trees for Cases 1 and 3 are identical are not listed separately. Classification relies heavily on following pauses (large pauses mainly indicating sentence boundaries) and current vowel durations (long durations usually indicating a sentence end).

4.2.2 Online experiments

As mentioned in Chapter 1, a real−time approach to punctuation (and other) classification tasks is a goal of this project. In order to understand the performance of such a system, an examination of the above tasks with only features before the event available to the LM and the decision tree classifier is reported here. A significant degradation in decision tree performance is expected, especially since the previous experiments heavily rely on F_PAU_DIR, the pause duration following the current word. Similarly, the language model can be expected to perform worse than in previous experiments, as it uses future word identity. Table 4.2.3 sums up results for experiments which only have access to previous features (**P**revious **O**nly), compared to the results for the full feature set, showing the full feature version for comparison, and showing the degradation between these two cases.

Case	Forward Features?	Train Set	Test Set	Chance Accuracy	LM Accuracy	Tree Accuracy	LM+Tree Accuracy
Case 1	Yes	True	True	90.91	93.95	93.41	94.84
Case 1−PO	No	True	True	90.91	92.91	91.06	92.91
Degradation					**−1.11%**	**−2.52%**	**−2.04%**
Case 2	Yes	ASR	ASR	91.25	91.77	91.25	92.62
Case 2−PO	No	ASR	ASR	91.25	91.52	91.25	91.59
Degradation					**−0.27%**	**−0.00%**	**−1.11%**
Case 3	Yes	True	ASR	91.25	92.43	92.43	92.43
Case 3−PO	No	True	ASR	91.25	91.25	91.25	91.39
Degradation					**−1.28%**	**−1.28%**	**−1.13%**

Table 4.2.3: Accuracies for classifiers using previous only (PO) features, given in percentages.

The largest losses shown above are sustained by the decision tree, most notably in the case of training and testing on true words (Case 1). While this loss is larger than all other losses, it should be noted that many of the values, such as in Case 2, are so close to chance that any further degradation would be minimal. The decision trees certainly suffer in all cases above, though, and fall to chance performance when testing and training on ASR. Obviously the combination of unreliable word boundaries, the confusion between

incomplete sentences, disfluent boundaries, and fluent boundaries, and the lack of future features (mainly following pause durations) are too difficult to overcome. Although not listed, degradations in efficiencies are similar to the drops in accuracies listed in Table 4.2.3; both language model and prosodic classifier efficiencies drop precipitously from the full feature set to the previous only case.

Interestingly, although prosodic feature based trees do not perform particularly well in some of the experiments above, the combination models in the ASR−tested cases always outperform the LM on its own, even when the prosodic model alone performs at chance. These improvements show that having a variety of information sources is better than limiting training to words or pitch features alone.

Table 4.2.4 shows feature usages for the PO experiments discussed above:

Case 1−PO		Case 2−PO		Case 3−PO	
Feature	Usage	Feature	Usage	Feature	Usage
Vowel Durations	**28.87**	**Vowel Durations**	**18.18**	**RMS Features**	**26.13**
C_VOWEL_DUR	20.37	C_VOWEL_DUR	18.18	C_RMS_V_MIN_Z	15.37
C_TRIVOWEL_DUR_Z	6.11	**Pitch Features**	**40.3**	P_RMS_V_MAX_Z	5.45
C_VOWEL_DUR_Z	2.39	C_F0K_LOGRATIO_SEGMIN_WORDMIN	16.5	C_RMS_V_MAX_Z	5.31
RMS Features	**20.50**	C_F0K_LOGDIFF_LASTPWLWIND100_BASELN	13.14	**Vowel Durations**	**26.06**
C_RMS_V_MIN_R	14.75	C_F0K_LOGRATIO_WIND50MIN_BASELN	4.93	C_VOWEL_DUR	20.48
P_RMS_V_MAX_Z	3.01	C_F0K_LOGRATIO_LASTPWLWIND100_BASELN	2.30	C_TRIVOWEL_DUR_Z	5.58
C_RMS_MIN_R	2.74	C_LAST_SLOPE_WIND_100	3.43	**Pitch Features**	**21.21**
Pitch Features	**23.28**	**RMS Features**	**22.6**	C_F0K_LOGRATIO_SEGMIN_WORDMIN	13.91
C_F0K_LOGRATIO_LASTPWLWIND100_BASELN	12.30	C_RMS_V_MIN_Z	12.31	C_F0K_LOGDIFF_WORDMIN_BASELN	3.85
C_F0K_LOGRATIO_SEGMIN_WORDMIN	10.98	C_RMS_V_MAX_R	10.29	C_F0K_LOGRATIO_WORDMIN_BASELN	3.45
Pause Features	**11.12**	**Pause Features**	**6.84**	**Other Features**	**10.85**
P_PAU_DUR	11.12	P_PAU_DUR	6.84	NAME	6.81
Overlap Features	**7.43**			C_WORD_WDPOS	4.04
P_IN_OVERLAP	7.43			**Overlap Features**	**8.12**
Other Features	**5.22**			P_IN_OVERLAP	8.12
NAME	5.22				

Table 4.2.4: Feature usages (in percentages) for Task 1, previous features only

The most heavily used feature, not surprisingly, is the current vowel duration, as this was the most widely used non−future feature in Cases 1 and 3. The most notable additions to the feature sets are the various normalizations of the energy feature C_RMS_V_MIN, which measures the minimum energy of the voiced frames in the current word. The inclusion of this feature in the prosodic model indicates that when the following pauses are not available, the tree begins looking at how loudly a person is speaking, as soft speech is a good indicator of sentence ends. Similarly, many pitch features are included in the table above. Most commonly used is C_F0K_LOGRATIO_LASTPWLWIND100_BASELN which measures the last valid (i.e., not halved, doubled or unvoiced) stylized F0 value in a window that starts at the end of the word and stretches back 100 frames. This feature indicates how speakers end their words, and if the last frame is particularly low, this can be a good indication of a sentence boundary.

Figure 4.2.2 compares performance of the prosodic tree, LM, and combination models *vis a vis* the exclusion of the future features. Note the change of chance accuracy from Case 1 to Cases 2 and 3.

From the figures it is clear that although the prosodic model does not necessarily perform better than the LM in Cases 2 and 3 the combination model performs better than either model on its own. Performance in Case 1, when the tree and LM are trained and tested on true words and are allowed to use both future and previous features is impressive at almost 95%, and there are large improvements in efficiency for Cases 1 and 2.

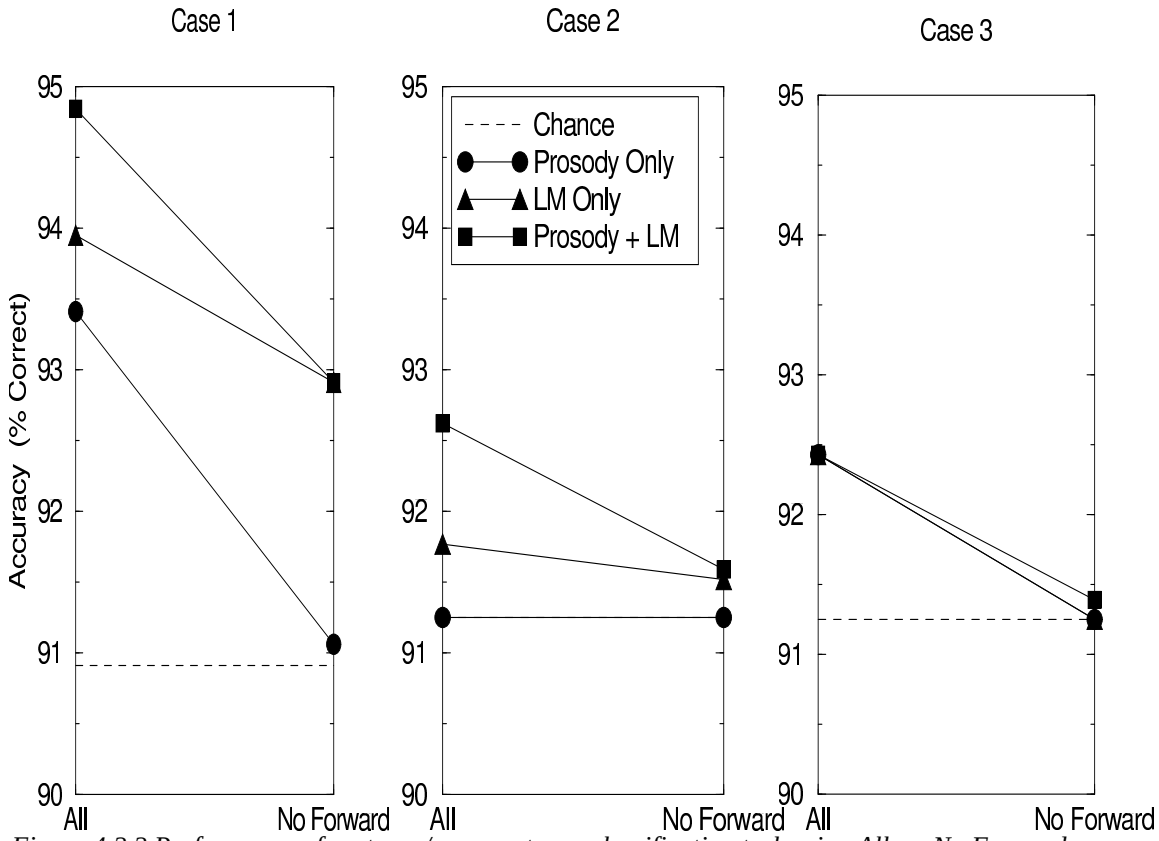


Figure 4.2.2 Performance of sentence/non−sentence classification task using All vs. No Forward features. Note that in Case 2, prosody performs at chance.

4.3 Task 2 results

Task 2 is similar to Task 1 in that a language model and prosodic feature based decision tree are incorporated to perform a punctuation classification task. In this task, however, disfluent boundaries are separated from fluent boundaries and introduce a second minority disfluency class, as shown below.

do	you	–	i	know	.	what	was	that	?
<n>		<d>		<n>	<s>		<n>	<n>	<s>

4.3.1 All feature regions

Case	Forward Features?	Train Set	Test Set	Chance Accuracy	LM Accuracy	Tree Accuracy	LM+Tree Accuracy
Case 1	Yes	True	True	**80.74**	**89.79**	**86.31**	**91.70**
				0.00	*56.51*	*35.64*	*63.61*
Case 2	Yes	ASR	ASR	**80.70**	**82.69**	**84.08**	**85.16**
				0.00	*21.02*	*22.86*	*28.21*
Case 3	Yes	True	ASR	**80.70**	**82.09**	**84.27**	**84.39**
				0.00	*7.55*	*23.72*	*24.48*

Table 4.3.1: Accuracies and efficiencies of three different train/test cases for Task 2. All results are given in percentages. Accuracies are in boldface, efficiencies in italics.

The roles of different data sets (ASR v. true words) and the effect of removing all future features from the decision tree model is examined below. As with Task 1, the effect of training/testing on ASR, removing future features, and also removing the NAME (speaker name) feature is examined. The last alteration is relevant because NAME has strong correlation to non−downsampled priors, and since the trees are using downsampled data in our experiments, the inclusion of this feature belies class prior equalization. Table 4.3.1 shows results for Task 2, including all features, but with variations on the test/train data sets. The variety of cases here is useful in determining the effect of performance drop when word errors are present in either the test or train sets (or both).

As mentioned in Chapter 2, downsampling of the data was performed for numerous reasons. Because of computation time and LM mismatch, running non−downsampled experiments on all the cases in Table 4.3.1 was not explored. Again, as Chapter 2 states, downsampled data is required for direct integration with the LM posteriors, so combination models for the full data sets are not available either. For the prosodic classifiers, however, running the classification task on all the data, results in an efficiency of 36.73%, a 8.89% relative increase from Case 1. It is necessary to compare efficiencies in this case, since priors are unequal. These values indicate that there is some performance drop from downsampling and that to some extent, the effectiveness of the prosodic trees as detailed in Tables 4.2.1 and 4.3.1 is underreported.

From the results above, it is clear that the combination of LM and decision tree trained on prosodic features performs better than either classifier alone. Also, prosodic features are much more robust to recognition errors, as the decision tree accuracy degrades by 2.65% relative to Case 1. The LM accuracy, on the other hand, degrades by 7.9% relative. The combination model suffers a loss of 7.13% relative.

The language model is clearly more susceptible to word errors, and the dramatic drop in LM classification efficiencies from Case 1 to Cases 2 and 3 reflect this. This is mostly likely due to the abundance of words such as "uh" and "uhhuh" which are automatically marked as disfluencies in our transcription process. When word errors are introduced many of the free gains the LM gets from these words are lost, since the erroneous words will no longer be marked as disfluencies. In contrast, the prosodic features are only peripherally affected by word errors rates insofar as they compromise word boundaries.

The decision tree actually performs better in Case 3 than it does in Case 2. While this difference is relatively small, it is important because despite an initial hit taken by the tree in training on recognized words, which ultimately degrade prosodic feature integrity, the tree is relatively robust to this data mismatch, as opposed to the LM, which clearly suffers.

Table 4.3.2 shows relative improvements above chance for Cases 1 and 2 in Tasks 1 and 2 .

Task *(classes)*	Test/Train	LM Improvement	Decision Tree Improvement	Combination Improvement
1 *(s ns)*	True/True	3.34%	2.75%	4.32%
1 *(s ns)*	ASR/ASR	0.57%	0%	1.50%
2 *(s di n)*	True/True	11.21%	6.97%	13.57%
2 *(s di n)*	ASR/ASR	2.47%	4.19%	5.53%

Table 4.3.2 Relative improvement above chance for Task 1(sentence/non−sentence) and Task2 (sentence/disfluency/fluent boundary) task when training/testing on ASR and true data.

In comparing Tasks 1 and 2, it is seen how using a separate disfluency class is extremely useful for both the prosodic feature based decision tree and the Language Model. As mentioned above, the LM in Task 2 gets many disfluencies such as "uh" and "um" for free. In Task 1, however, the LM gets nothing for free and the poorer relative

improvement reflects this disadvantage. Similarly, the decision trees strongly prefer the addition of a distinct disfluency class, since this class disambiguates disfluencies from fluent word boundaries, which were all lumped into one class in Task 1. Trained to spot these disfluencies, the decision tree in Task 2 clearly outperforms the tree in Task 2. The combinations models, which are strongly dependent on the individual model performances also do much better in the three class task than in Task 1.

The features used by the decision trees in Table 4.3.1 are mainly those from the vowel duration and pause duration subset. Table 4.3.3. shows feature usage percentages. These values indicate the percentage of all decisions made using any feature.

Case 1		Case 2		Case 3	
Feature	Usage	Feature	Usage	Feature	Usage
Vowel Durations	**61.80**	**Vowel Durations**	**52.22**	**Vowel Durations**	**46.26**
C_VOWEL_DUR	28.24	C_VOWEL_DUR	25.53	C_VOWEL_DUR	28.64
C_VOWEL_DUR_Z	9.72	P_VOWEL_DUR	9.32	C_VOWEL_DUR_N	17.62
F_TRIVOWEL_DUR_N	6.17	C_VOWEL_DUR_Z	8.81	**Pause Durations**	**32.46**
C_VOWEL_DUR_N	5.83	F_VOWEL_DUR	2.58	F_PAU_DUR	28.12
C_TRIVOWEL_DUR_Z	5.08	F_VOWEL_DUR_N	2.42	P_PAU_DUR	4.28
P_TRIVOWEL_DUR_Z	2.77	P_VOWEL_DUR_N	2.24	**Other Features**	**14.12**
P_VOWEL_DUR_Z	2.12	C_TRIVOWEL_DUR_N	1.00	NAME	***14.12***
F_VOWEL_DUR	1.87	P_TRIVOWEL_DUR_Z	0.32	**RMS Features**	**7.22**
Pause Durations	**32.66**	**Pause Durations**	**47.58**	C_RMS_V_MIN_R	7.22
F_PAU_DUR	25.47	F_PAU_DUR	40.32		
P_PAU_DUR	4.16	P_PAU_DUR	5.38		
PP_PAU_DUR	3.03	PP_PAU_DUR	1.88		

Table 4.3.3 Feature Usage in Task 2, Cases 1,2,3. Usages are given in percentages. Note the inclusion of the NAME feature in Case 3.

Similar to what is seen in Task 1, the table indicates that the two main feature sets used across cases are vowel durations (most notably C_VOWEL_DUR, the duration of the longest vowel in the current word), and pause durations (in which F_PAU_DUR, the amount of silence following the current word, appears most frequently). These results concur with theoretical expectations; a large pause after a word tends to indicate that the word is at some sort of semantic boundary. Similarly, speakers generally draw out sounds towards the end of sentences, while speaking quickly at sentence onset.

As mentioned above, the presence of the NAME feature is not preferred as inherent prosodic properties of these events, rather than prior related features, would be

more appropriate inputs to the decision trees. The feature is omitted and the results for this new case, Case 3B are presented in Table 4.3.4.

Case	NAME feature?	Train Set	Test Set	Chance Accuracy	LM Accuracy	Tree Accuracy	LM+Tree Accuracy
Case 3	Yes	True	ASR	80.70	82.09	84.39	84.24
Case 3B	No	True	ASR	80.70	82.09	84.12	83.98
Degradation						**−0.30%**	**−0.30%**

Case 3		Case 3B	
Feature	Usage	Feature	Usage
Vowel Durations	**46.26**	**Vowel Durations**	**52.57**
C_VOWEL_DUR	28.64	C_VOWEL_DUR	24.22
C_VOWEL_DUR_N	17.62	C_VOWEL_DUR_N	8.55
Pause Durations	**32.40**	C_TRIVOWEL_DUR_N	8.11
F_PAU_DUR	28.12	P_TRIVOWEL_DUR_N	6.16
P_PAU_DUR	4.28	C_VOWEL_DUR_Z	5.53
Other Features	**14.12**	**Pause Durations**	**47.43**
NAME	**14.12**	F_PAU_DUR	26.46
RMS Features	**7.22**	P_PAU_DUR	20.96
C_RMS_V_MIN_R	7.22		

Table 4.3.4: Accuracies and Feature usage of variation of Case 3 , where NAME is excluded

Removing the NAME feature causes only a slight degradation of 0.3% relative to experiment with all the features present. While the top three features are the same, Case 3B looks at new features such as vowel triphones in order to make up for the lack of the speaker identities.

Finally, trees for Cases 1, 2 and 3B are examined. Figure 4.3.1 displays all three trees . Because of space limitations, only the top four splits are included. The trees confirm what is expected from theory and practical observations: long vowel and following pause durations are generally used to classify sentence ends , where as short values for these features usually indicate fluent boundaries. The case of disfluencies isn't as clear. In terms of pauses, the trees often pick disfluencies when the following pause duration is longer than a fluent boundary, but not quite as long as a sentence end, as shown in this example from Case 3B :

```
F_PAU_DUR < 0.145:  0.3378 0.5158 0.1464 n
F_PAU_DUR >= 0.145:  0.3269 0.06896 0.6042 s
       C_VOWEL_DUR < 25.5:  0.3839 0.1485 0.4676 s
  F_PAU_DUR < 0.373:  0.4606 0.1694 0.37 di
F_PAU_DUR >= 0.373:  0.3087 0.1281 0.5632 s
       C_VOWEL_DUR >= 25.5:  0.7909 0.09444 0.1146 di
```

```
                          **********************
                                  Case 1
                          **********************


 0.3333 0.3333 0.3333 di n s
F_PAU_DUR < 0.195:  0.3461 0.5042 0.1496 n
|   C_VOWEL_DUR < 19.5:  0.2759 0.5654 0.1587 n
|   C_VOWEL_DUR >= 19.5:  0.7745 0.1311 0.0944 di
F_PAU_DUR >= 0.195:  0.313 0.06217 0.6248 s
|   F_PAU_DUR < 1.666:  0.4021 0.1043 0.4936 s
|   |   C_VOWEL_DUR < 24.5:  0.3231 0.1091 0.5678 s
|   |   C_VOWEL_DUR >= 24.5:  0.7433 0.08358 0.1732 di
|   F_PAU_DUR >= 1.666:  0.1947 0.00614 0.7992 s


                          **********************
                                  Case 2
                          **********************

0.3333 0.3333 0.3333 di n s
F_PAU_DUR < 0.497:  0.3486 0.4772 0.1742 n
|   F_PAU_DUR < 0.065:  0.3105 0.5528 0.1367 n
|   |   C_VOWEL_DUR < 15.5:  0.2701 0.596 0.1339 n
|   |   C_VOWEL_DUR >= 15.5:  0.5242 0.3243 0.1515 di
|   F_PAU_DUR >= 0.065:  0.4852 0.2064 0.3084 di
F_PAU_DUR >= 0.497:  0.3028 0.04606 0.6511 s
|   C_VOWEL_DUR < 25.5:  0.2708 0.04589 0.6833 s
|   |   F_PAU_DUR >= 0.829:  0.2631 0.02545 0.7115 s
|   C_VOWEL_DUR >= 25.5:  0.4512 0.04686 0.5019 s


                          **********************
                                  Case 3B
                          **********************

 0.3333 0.3333 0.3333 di n s
F_PAU_DUR < 0.145:  0.3378 0.5158 0.1464 n
|   C_VOWEL_DUR < 11.5:  0.2472 0.6216 0.1313 n
|   |   F_PAU_DUR < 0.045:  0.2258 0.6495 0.1247 n
|   |   |   C_VOWEL_DUR < 5.5:  0.2066 0.7129 0.08049 n
|   |   |   C_VOWEL_DUR >= 5.5:  0.2477 0.5768 0.1754 n
|   |   F_PAU_DUR >= 0.045:  0.5541 0.2206 0.2253 di
|   C_VOWEL_DUR >= 11.5:  0.5739 0.2402 0.1859 di
|   |   C_VOWEL_DUR < 32.5:  0.5 0.2774 0.2226 di
|   |   |   C_VOWEL_DUR_Z < -0.3:  0.935 0.04469 0.02028 di
|   |   |   C_VOWEL_DUR_Z >= -0.3:  0.4286 0.3156 0.2558 di
|   |   C_VOWEL_DUR >= 32.5:  0.8931 0.07938 0.0275 di
F_PAU_DUR >= 0.145:  0.3269 0.06896 0.6042 s
|   F_PAU_DUR < 0.7345:  0.4565 0.1388 0.4046 di
|   |   C_VOWEL_DUR < 25.5:  0.3839 0.1485 0.4676 s
|   |   |   F_PAU_DUR < 0.373:  0.4606 0.1694 0.37 di
|   |   |   F_PAU_DUR >= 0.373:  0.3087 0.1281 0.5632 s
|   |   C_VOWEL_DUR >= 25.5:  0.7909 0.09444 0.1146 di
|   |   |   C_VOWEL_DUR_N < 1.7:  0.9477 0.02254 0.02972 di
|   |   |   C_VOWEL_DUR_N >= 1.7:  0.6685 0.1506 0.1809 di
```

Figure 4.3.1 Abridged decision trees for Cases 1,2, and 3B, for three−way classification of sentences, disfluencies, and fluent boundaries, trained on equal class priors. Class labels are di(disfluency), n (fluent boundary) , s (sentence) and probabilities are listed in that order. Widely used features include F_PAU_DUR (the pause duration after the current word) and C_VOWEL_DUR, the maximum vowel duration of the current word, and normalized vowel duration statistics.

While it is difficult to generalize from just a few trees, these disfluencies are probably being distinguished from sentences at turn taking boundaries, which in general may have longer pause lengths than intraturn sentence boundaries.

4.3.2 Online experiments

Results using the **P**revious **O**nly feature set is examined in Table 4.3.5.

Case	Forward Features?	Train Set	Test Set	Chance Accuracy	LM Accuracy	Tree Accuracy	LM+Tree Accuracy
Case 1	Yes	True	True	80.74	89.79	86.37	91.70
Case 1−PO	No	True	True	80.74	86.37	83.93	86.53
Degradation					**−3.81%**	**−2.83%**	**−5.64%**
Case 2	Yes	ASR	ASR	80.70	82.69	84.08	85.16
Case 2−PO	No	ASR	ASR	80.70	81.78	81.76	82.76
Degradation					**−1.10%**	**−2.76%**	**−2.82%**
Case 3B	Yes	True	ASR	80.70	82.09	84.12	83.98
Case 3B−PO	No	True	ASR	80.70	81.85	81.43	82.06
Degradation					**−0.29%**	**−3.20%**	**−2.29%**

Table 4.3.5: Results for previous feature only (PO) cases of three way classification experiment, Task 2. Degradation relative to original experiments are given in bold

Both the LM and the decision trees suffer in all cases, which is understandable since the knowledge of the following word or the following pause duration are critical in good classification. The greatest degradation comes in Case 1, where the trees are trained and tested on the true words, and depriving the trees and LM of forward features creates a loss of 5.64% relative in the combined model. Even in this case, however, the decision tree degrades more gracefully than the LM (2.8% for tree vs. 3.8% for LM) and the combination model still does better than words alone.

In Cases 2 and 3B, where the LM already incurs a strong hit because of word recognition errors, as discussed above, LM degradation isn't as large as the decision tree, but their accuracies are very close. Interestingly, the combined model in these cases still outperforms either model on its own. In these cases, where word errors rob the LM of obvious easy decisions, the majority of the loss in the LM comes from word error rates rather than the deprivation of future word knowledge.

Case 1−PO		Case 2−PO		Case 3B−PO	
Feature	Usage	Feature	Usage	Feature	Usage
Vowel Durations	**52.87**	**Pitch Features**	**24.83**	**Vowel Durations**	**53.99**
C_VOWEL_DUR	27.79	C_F0K_RATIOSHIFT_SEGMIN_WORDMIN_BASELN	15.10	C_VOWEL_DUR	29.57
C_VOWEL_DUR_N	16.76	C_F0K_LOGDIFF_LASTPWLWIND100_BASELN	6.14	C_VOWEL_DUR_Z	16.48
C_TRIVOWEL_DUR_Z	8.32	C_F0K_LOGRATIO_WORDMIN_BASELN	3.59	C_TRIVOWEL_DUR_Z	7.94
Pitch Features	**18.56**	**RMS Features**	**24.72**	**Pitch Features**	**17.71**
C_F0K_RATIOSHIFT_SEGMIN_WORDMIN_BASELN	8.56	C_RMS_V_MIN_R	14.73	C_F0K_LOGRATIO_LASTPWLWIND100_BASELN	11.06
C_F0K_LOGRATIO_LASTPWLWIND100_BASELN	6.82	C_RMS_V_MAX_Z	9.99	C_F0K_LOGRATIO_WORDMIN_BASELN	6.12
C_F0K_LOGDIFF_WORDMIN_BASELN	3.18	**Vowel Durations**	**22.57**	CP_FOK_LOGDIFF_MAXPWLWORD_MAXPWL_P	0.53
RMS Features	**12.56**	C_VOWEL_DUR	18.98	**Pause Features**	**12.71**
C_RMS_V_MIN_R	12.56	C_VOWEL_DUR_N	3.59	P_PAU_DUR	12.71
Pause Features	**9.98**	**Other Features**	**15.25**	**RMS Features**	**10.44**
P_PAU_DUR	8.14	NAME	15.25	C_RMS_V_MIN_R	10.44
PP_PAU_DUR	1.84	**Pause Features**	**7.88**	**Overlap Features**	**3.99**
Overlap Features	**3.48**	P_PAU_DUR	7.88	P_IN_OVERLAP	3.99
C_IN_OVERLAP	3.48	**Other Features**	**1.8**	**Other Features**	**0.69**
		C_WORD_WDPOS	1.8	C_PERC_DOUB	0.69

Table 4.3.6: Feature usage for experiments with access to previous features only. Vowel durations are still extremely useful, but all experiments use more pitch features than they did in the all features case. Feature usages are reported in percentages.

Feature usages for the previous only experiments are in Table 4.3.6. Not surprisingly and similarly to Task 1, when not allowed to look into the future, the decision trees look elsewhere for discriminating features. Pitch features tend to be used much more often in these experiments. One feature seems to occur more than others, and this is C_F0K_RATIOSHIFT_SEGMIN_WORDMIN_BASELN, which is measure of the difference between the lowest pitch value in the current word and the lowest pitch value in the segment. When this value is low, the boundary is generally classified as a sentence. Also, in the case of test and train on ASR word, the NAME feature once again appears. When the decision tree is left with few options (i.e., no forward features allowed and in the presence of word errors) it will begin using prior−correlated features, which is what happens in Case2−PO. Figure 4.3.2 graphically depicts the effects of removing forward features from these different cases.

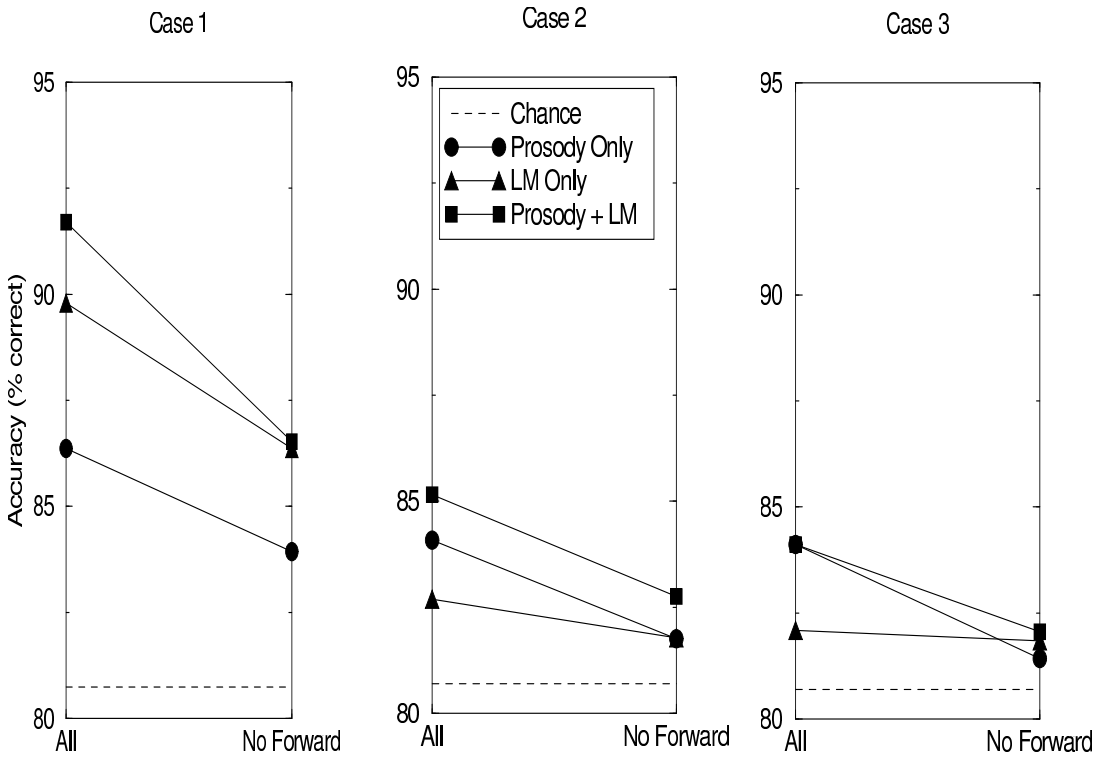


Figure 4.3.2: Event detection accuracy for Task 1. "True" = true words (forced alignments); "ASR" = 1−best recognizer output; "LM" = Language Model

4.3.3 Speaker specific results

Finally, in this task prediction results for one speaker at a time are considered. Motivating these experiments is a desire to more fully understand the value of a large training set versus a smaller training set which is comprised of only one speaker. Models tested on trained on only one participant are reported, and the tradeoff between less but cleaner training data and the an analysis of model degradation with respect to individual speaker word error rates are shown here.

The speech of three speakers who were present in at least 10 meetings were chosen as the data sets for these experiments and Table 4.3.7 discusses the amount of data and word error rates for the speakers in this section.

Speaker	Gender	Meeting Type	Overall Word Error Rate	Words in Train	Words in Test
Speaker A	Male	Bmr	39.58%	25708	5056
Speaker B	Female	Bmr	44.99%	21236	3789
Speaker C	Male	Bmr	46.09%	30154	8553
All	M/F	Bed, Bmr, Bro	52.46%	252261	52069

Table 4.3.7: Data and word error rates for three speakers and the entire data set.

For these speakers, experiments were only only conducted with the entire feature set, and without the mismatched train/test set (i.e., Case 3 is excluded). Accuracies and relative performance degradations for this classification experiment are reported in Table 4.3.8.

Speaker	Forward Features?	Train Set	Test Set	Chance Accuracy	LM Accuracy	Tree Accuracy	LM+Tree Accuracy
A	Yes	True	True	83.50	87.94	89.66	91.81
	Yes	ASR	ASR	83.10	84.84	88.67	89.01
Degradation					**−3.53%**	**−1.10%**	**−3.04%**
B	Yes	True	True	81.10	88.12	85.09	90.39
	Yes	ASR	ASR	81.01	82.82	84.87	85.46
Degradation					**−6.01%**	**−0.26%**	**−5.45%**
C	Yes	True	True	82.58	89.37	84.75	91.20
	Yes	ASR	ASR	82.61	84.34	84.85	85.82
Degradation					**−5.63%**	**+0.18%**	**−5.90%**

Table 4.3.8: Model performance across three speakers. Degradations are given in bold.

Clearly, word recognition rates play a significant role in the degradation from the manual test/train sets to the ASR data. Speakers B and C, who have the largest word error rates amongst this speaker subset, experience the largest loss in the LM and the combined model. Curiously, it is Speaker A who incurs the biggest hit in the prosodic model, despite relatively good recognition results. This could be due to insufficient data − Speaker C, who has the most total number of or words actually does better in the ASR case. These observations offer insights into performances for these particular speakers, but it is difficult to conclude anything with certainty for so few participants.

Another interesting observation from Table 4.3.8 is how well Speaker A's prosodic decision tree does with respect to his language model classifier. Note that this is

the only speaker in which the prosodic model outperforms the language model. This may be an indication that some speakers exhibit more consistent, or more clear, prosody than other speakers and that variable prosodic performance across meeting participants can be expected.

Figure 4.3.3 compares LM, prosodic tree, and combination model performance degradations for these three speakers, along with the overall results reported in Table 4.3.1. In the figure the disparity between LM and combination model losses versus the relatively mild degradation in the prosodic tree is quite striking. Certainly the combination model's performance is more dependent on the LM performance as these two losses are always within one percent of one another.

Table 4.3.8 and Figure 4.3.3 indicate a relationship between word error rates, total number of words and overall performance. From these figures, one can surmise it is better to have speaker specific training data, rather than more data. In other words, allowing the prosodic feature based decision tree to learn the speaking pattern of one speaker is more helpful for classification than providing many, but prosodically different, speakers. Table 4.3.9 shows results of testing on individual speakers' data with decision trees that have been trained on the entire data set, along with the change relative to training on individual speakers. The results show that including multiple speakers into the training set does not necessarily cause decision tree performance to degrade. In fact, while the changes may be statistically insignificant, in all but one case shown in the table the training on the all the speakers increases the performance of the decision tree.

Speaker	Train	Test	True/ASR?	Chance	Prosody	Change
A	All	A	True	83.50	89.89	0.26%
A	All	A	ASR	83.10	88.79	0.14%
B	All	B	True	81.09	86.51	1.67%
B	All	B	ASR	81.00	84.85	−0.02%
C	All	C	True	82.58	87.67	3.4%
C	All	C	ASR	82.60	85.15	0.35%

Table 4.3.9: Results for a decision tree trained on full training data set but tested on individual speakers, along with performance changes relative to models trained on individual speakers.

Similarly, it is difficult to conjecture about the relationship between word error

rates and the decision tree performance. In certain cases it seems that speakers that have smaller word error rates are more robust to classification performance degradation, whereas sometimes it appears that the number of words is a more important factor in determining how a model performs on any given speaker. Unfortunately, there were no speakers in the database which provided both a comparable number of words as Speakers A, B, C while also sustaining large word error rates. The inclusion of such a speaker could provide significant insight into the effects of all of these factors.

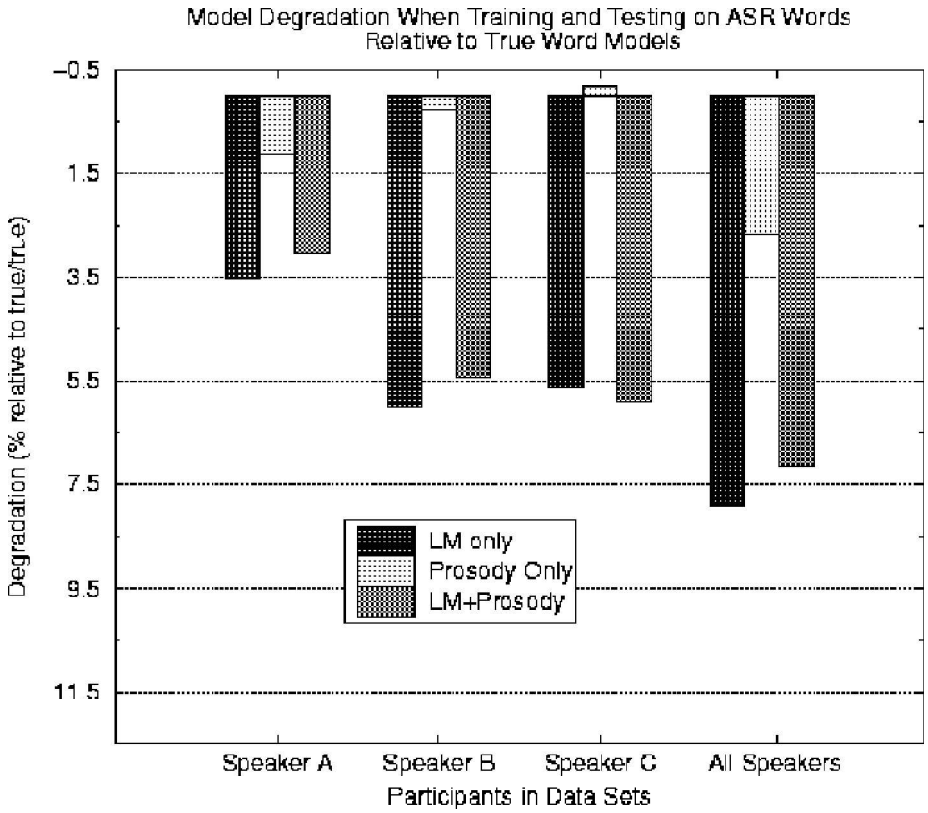


Figure 4.3.3: Relative degradations for the three models, for individual speakers and all speakers. Language and combination models both degrade much more significantly than the prosodic model. Word Error Rates (WER) and total word counts are given in the figure as well. Note that total words reported here are the total from the training set only.

4.4 Task 3 results

The final punctuation experiment aims to classify sentence ends as either questions or declarative sentences. Unlike the previous two tasks, Task 3 is not just a punctuation but also a dialog act distinction, where discourse−level annotation is

predicted. As discussed in (D. Jurafsky, et al., 2000) , some questions are indicated prosodically by pitch and energy rises, whereas sentence ends are generally accompanied by drops in these features. Thus, F0 and RMS values are expected to appear as discriminating features in the classifiers.

4.4.1 All feature regions

As mentioned previously, all words that are not followed by a sentence boundary are discarded, and a language model is not used to model these events, as the current LM requires a contiguous word stream. Also, for this task all incomplete sentences and disfluencies were excluded from the data set, so only well formed sentence ends are classified:

```
who    is     that   ?    oh      i      see   .
 EXC    EXC        <Q>       EXC    EXC       <P>

  EXC = Excluded, Q = Question, P = Period
```

These two phenomena were carefully observed in order to find inherent prosodic cues between these two classes. Results are in Table 4.4.1.

Case	Forward Features?	Train Set	Test Set	Efficiency
Case 1	Yes	True	True	13.00
Case 2	Yes	ASR	ASR	10.88
Case 3	Yes	True	ASR	11.32

Table 4.4.1: Results for a two−way question/sentence classification task for 3 different test/train combinations. Efficiencies are given in percentages.

As expected, the prosodic tree in Case 1 clearly outperforms the other cases, yielding an efficiency of 13.00%. Interestingly, Case 3 does better than Case 2, despite the test/train mismatch. Table 4.4.2 describes feature usages in all three cases. Case 2 uses the NAME feature for almost half of all decisions (45.80%) and is clearly trying to learn the class priors as some speakers ask questions at a different rate than others.

Similarly, Case 3 also uses name quite extensively at 23.54 %. From these results it appears that NAME is used in cases that tested on ASR. This may be because word errors encountered in testing make prosodic features less reliable, forcing the decision trees to examine prior−correlated features such as NAME. NAME is omitted since inherent prosodic differences between the two classes are sought.

Case 1		Case 2		Case 3	
Feature	Usage	Feature	Usage	Feature	Usage
Pause Features	**64.33**	**Other Features**	**45.80**	**Pause Features**	**36.74**
P_PAU_DUR	31.44	NAME	45.80	P_PAU_DUR	26.65
F_PAU_DUR	19.96	**Pause Features**	**34.22**	PP_PAU_DUR	10.08
PP_PAU_DUR	12.93	P_PAU_DUR	20.99	**Pitch Features**	**32.23**
		PP_PAU_DUR	8.29	C_F0K_LOGRATIO_ LASTPWLWIND100_ BASELN	32.23
Pitch Features	**35.67**				
C_F0K_LOGDIFF_ LASTPWLWIND100_ BASELN	20.01	F_PAU_DUR	5.05		
				Other Features	**23.54**
C_F0K_LOGRATIO_ WIND50MAX_BASELN	8.57			NAME	23.54
		Pitch Features	**19.87**		
C_F0K_LOGRATIO_ LASTPWLWIND100_ BASELN	7.08	C_F0K_LOGDIFF_ LASTPWLWIND100_ BASELN	18.53		
				Vowel Features	**7.51**
		C_F0K_RATIOSHIFT_ SEGMAX_WORDMAX_ BASELN	1.34	C_VOWEL_DUR_Z	7.51

Table 4.4.2: Feature Usage in Task 3, Cases 1,2,3. Usages are given in percentages. Note the inclusion of the NAME feature in Cases 2 and 3. Also note the heave dependence on pitch features and previous pause durations.

Table 4.4.3 shows results when after the removal of the NAME feature from Cases 2 and 3. Note that without the inclusion of NAME in the feature set in Case 2B, performance increases, as the tree gives up on this feature and finds that prosodic features such as durations and pitch are more useful. This could be due to overtraining in the cross−validation stage, since the cross−validation was not partitioned by speaker, as noted in Chapter 2. Nonetheless, the fact that performance is comparable to those cases with the NAME feature included is quite encouraging.

Case	NAME feature?	Train Set	Test Set	Efficiency
Case 2	Yes	ASR	ASR	10.88
Case 2B	No	ASR	ASR	11.42
Case 3	Yes	True	ASR	11.32
Case 3B	No	True	ASR	10.73

Table 4.4.3: Results for Task 3 with NAME feature removed from Cases 2 and 3.

Feature usages for these two cases show that, at least for Case 2, when not given the opportunity to learn NAME, the decision tree classifiers utilize features such as pitch and durations. Case 3, interestingly, attempts to learn other features proportional to class priors, namely NATIVE (is a speaker a native American English speaker?) and MTYPE (meeting type, i.e., Bed, Bmr, Bro). Table 4.4.4 shows features for Cases 1, 2B, and 3B.

Case 1		Case 2B		Case 3B	
Feature	Usage	Feature	Usage	Feature	Usage
Pause Features	**64.33**	**Pause Features**	**59.08**	**Pitch Features**	**44.77**
P_PAU_DUR	31.44	PP_PAU_DUR	30.50	C_F0K_LOGRATIO_ LASTPWLWIND100_ BASELN	27.85
F_PAU_DUR	19.96	P_PAU_DUR	28.58	CP_FOK_DIFF_ MAXPWLWORD_ MAXPWL_P−WORD	11.33
PP_PAU_DUR	12.93			C_F0K_RATIOSHIFT_ SEGMIN_WORDMIN _BASELN	5.59
		Pitch Features	**36.69**		
		C_F0K_LOGDIFF_WORD MAX_BASELN	36.69		
Pitch Features	**35.66**			**Pause Features**	**28.18**
C_F0K_LOGDIFF_ LASTPWLWIND100_ BASELN	20.01			P_PAU_DUR	28.18
		RMS Features	**4.22**		
C_F0K_LOGRATIO_ WIND50MAX_BASELN	8.57	C_RMS_MAX_R	4.22		
				Other Features	**27.06**
C_F0K_LOGRATIO_ LASTPWLWIND100_ BASELN	7.08			NATIVE	15.12
				MTYPE	11.94

Table 4.4.4: Feature Usage in Task 3, Cases 1,2B,3B. Usages are given in percentages.

The table above indicates that previous pause durations and current word pitch features are the most useful in discriminating question boundaries from sentence

boundaries. The pitch features, comprised of mostly last, minimum and maximum word F0 measures, are similar to those used in Tasks 1 and 2. However, whereas in Tasks 1 and 2 following pause features were extremely useful, these do not appear extensively in the feature lists above, indicating that there is little or no difference in the pause values following declarative sentences and questions.

Similarly, vowel durations no longer appear on this list. Duration features are useful in discriminating sentence/question end words from words within a sentence, but apparently not between the sentence and question end words themselves. Figure 4.4.1 shows the trees for Cases 1, 2B, and 3B. In this case the full decision trees are given, as they are much more tractable than the trees in Tasks 1 and 2.

The trees confirm the theoretical assumptions about helpful prosodic differences between sentence and question end words. In numerous cases in Figure 4.4.1 low pitch values indicate periods, whereas pitch rises are used to delineate question ends. In Case 2B, for example, the following decision split indicates a decision based on pitch drop:

```
C_F0K_LOGDIFF_WORDMAX_BASELN < 3.5615:  0.5432 0.4568 PER
C_F0K_LOGDIFF_WORDMAX_BASELN >= 3.5615:  0.4219 0.5781 Q
```

In this case the tree uses a feature which measures the difference between the maximum stylized F0 value for the current word, and the baseline. If the value is above a certain threshold (i.e., high pitch) the decision is Q, whereas low pitches are classified as PER. The use of previous pause durations (i.e., P_PAU_DIR and PP_PAU_DIR) probably indicates that the classifier is learning the nature of the many backchannels in the corpus. These are short utterances (one or two words) that are usually preceded by a large amount of silence. Backchannels are very well defined in this sense, and the classifier is probably learning the nature of these utterances in this tree.

4.4.2 Previous only features

The main features used in Cases 1, 2B, and 3B involved current pitch and energy features along with previous pauses. Because of this feature usage, it is predicted that questions are not isolated events, but rather tied to the prosodic and lexical context in the sentence preceding it. That said, a relatively small degradation due to removal of all

```
**********************
Case 1
**********************
P_PAU_DUR < 0.885:  0.4329 0.5671 PER Q
|   C_F0K_LOGDIFF_LASTPWLWIND100_BASELN < 3.4809:  0.5266 0.4734 PER
|   |   PP_PAU_DUR < 0.335:  0.5055 0.4945 PER
|   |   |   F_PAU_DUR < 2.465:  0.5383 0.4617 PER
|   |   |   |   C_F0K_LOGRATIO_WIND50MAX_BASELN < 0.76845:  0.5461 0.4539 PER
|   |   |   |   |   F_PAU_DUR < 1.297:  0.5597 0.4403 PER
|   |   |   |   |   |   P_PAU_DUR < 0.055:  0.5794 0.4206 PER
|   |   |   |   |   |   P_PAU_DUR >= 0.055:  0.4125 0.5875 Q
|   |   |   |   |   F_PAU_DUR >= 1.297:  0.45 0.55 Q
|   |   |   |   C_F0K_LOGRATIO_WIND50MAX_BASELN >= 0.76845:  0.3481 0.6519 Q
|   |   |   F_PAU_DUR >= 2.465:  0.4166 0.5834 Q
|   |   PP_PAU_DUR >= 0.335:  0.7327 0.2673 PER
|   C_F0K_LOGDIFF_LASTPWLWIND100_BASELN >= 3.4809:  0.2619 0.7381 Q
|   |   C_F0K_LOGRATIO_LASTPWLWIND100_BASELN < 0.32256:  0.6886 0.3114 PER
|   |   C_F0K_LOGRATIO_LASTPWLWIND100_BASELN >= 0.32256:  0.2393 0.7607 Q
P_PAU_DUR >= 0.885:  0.8184 0.1816 PER

**********************
Case 2B
**********************
 0.5 0.5 PER Q
PP_PAU_DUR < 0.9005:  0.4612 0.5388 Q
|   P_PAU_DUR < 1.316:  0.4255 0.5745 Q
|   |   C_F0K_LOGDIFF_WORDMAX_BASELN < 3.7097:  0.5218 0.4782 PER
|   |   |   C_F0K_LOGDIFF_WORDMAX_BASELN < 3.5615:  0.5432 0.4568 PER
|   |   |   C_F0K_LOGDIFF_WORDMAX_BASELN >= 3.5615:  0.4219 0.5781 Q
|   |   C_F0K_LOGDIFF_WORDMAX_BASELN >= 3.7097:  0.3218 0.6782 Q
|   P_PAU_DUR >= 1.316:  0.7574 0.2426 PER
PP_PAU_DUR >= 0.9005:  0.7415 0.2585 PER
|   C_RMS_MAX_R < 1.0363:  0.8852 0.1148 PER
|   C_RMS_MAX_R >= 1.0363:  0.6215 0.3785 PER
|   |   P_PAU_DUR < 0.03:  0.4641 0.5359 Q
|   |   |   C_F0K_LOGDIFF_WORDMAX_BASELN < 3.7642:  0.6923 0.3077 PER
|   |   |   C_F0K_LOGDIFF_WORDMAX_BASELN >= 3.7642:  0.3551 0.6449 Q
|   |   P_PAU_DUR >= 0.03:  0.7607 0.2393 PER

**********************
Case 3B
**********************
 0.5 0.5 PER Q
P_PAU_DUR < 0.885:  0.4329 0.5671 Q
|   C_F0K_LOGRATIO_LASTPWLWIND100_BASELN < 0.34727:  0.5327 0.4673 PER
|   |   NATIVE = non:  0.6205 0.3795 PER
|   |   |   MTYPE in Bmr,Bro, :  0.688 0.312 PER
|   |   |   MTYPE in Bed, :  0.5044 0.4956 PER
|   |   |   |   C_F0K_LOGRATIO_LASTPWLWIND100_BASELN < -0.099466:  0.3691 0.6309 Q
|   |   |   |   C_F0K_LOGRATIO_LASTPWLWIND100_BASELN >= -0.099466:  0.5688 0.4312 PER
|   |   NATIVE = nat:  0.5034 0.4966 PER
|   |   |   CP_FOK_DIFF_MAXPWLWORD_MAXPWL_P-WORD < 9.545:  0.5213 0.4787 PER
|   |   |   |   MTYPE in Bed,Bmr, :  0.5437 0.4563 PER
|   |   |   |   MTYPE in Bro, :  0.4725 0.5275 Q
|   |   |   |   |   C_F0K_RATIOSHIFT_SEGMIN_WORDMIN_BASELN < -1.0651:  0.5719 0.4281 PER
|   |   |   |   |   C_F0K_RATIOSHIFT_SEGMIN_WORDMIN_BASELN >= -1.0651:  0.414 0.586 Q
|   |   |   CP_FOK_DIFF_MAXPWLWORD_MAXPWL_P-WORD >= 9.545:  0.4574 0.5426 Q
|   |   |   |   C_F0K_LOGRATIO_LASTPWLWIND100_BASELN < 0.3236:  0.4451 0.5549 Q
|   |   |   |   |   C_F0K_RATIOSHIFT_SEGMIN_WORDMIN_BASELN < -5.3643:  0.6383 0.3617 PER
|   |   |   |   |   C_F0K_RATIOSHIFT_SEGMIN_WORDMIN_BASELN >= -5.3643:  0.4196 0.5804 Q
|   |   |   C_F0K_LOGRATIO_LASTPWLWIND100_BASELN >= 0.3236:  0.6911 0.3089 PER
|   C_F0K_LOGRATIO_LASTPWLWIND100_BASELN >= 0.34727:  0.248 0.752 Q
P_PAU_DUR >= 0.885:  0.8184 0.1816 PER
```

Figure 4.4.1: Tree for Task 3, Cases 1, 2B and 3B. The classes are Period (PER) and Question(Q) and posteriors are listed in that order. Note than low pitch features are good indicators for sentence ends, while pitch rises usually correspond to questions, as predicted. Similarly in Case 2B, when energy and pitch are high, the classifier will guess Question. Case 3B tries to recover from the data mismatch by using the meeting type and whether a speaker is a native American English speaker, both of which are features that correlate to class priors

future features is expected, which is a clear departure from hypotheses made in previous tasks. Table 4.4.5 describes the change in performance in these cases for these Cases.

What is interesting about the above results is that in distinguishing sentence ends from question ends, there is no need for future features at all, indicating that these events are not isolated in time, but rather related to previous prosody. It seems that forcing the prosodic classifier to only look at previous features increases performance in Cases 1 and 2B. Case 3B, in which the data mismatch occurs, is learning the prosodic characteristics of the manually transcribed data, which differs from the test set, since the latter has markedly poorer word boundaries. Finally, Table 4.4.6 examines the feature usage for the PO cases.

Case	Forward Features?	NAME included?	Train Set	Test Set	Efficiency
Case 1	Yes	Yes	True	True	13.00
Case 1−PO	No	Yes	True	True	13.41
Change					**+3.15%**
Case 2B	Yes	No	ASR	ASR	11.42
Case 2B−PO	No	No	ASR	ASR	11.54
Change					**+1.05%**
Case 3B	Yes	No	True	ASR	10.73
Case 3B−PO	No	No	True	ASR	9.72
Change					**−9.41%**

Table 4.4.5: Efficiencies for Cases 1, 2B, and 3B in the case of All and Previous Only (PO) features

Case 1−PO		Case 2B−PO		Case 3B−PO	
Feature	Usage	Feature	Usage	Feature	Usage
Pitch Features	**51.1**	**Pause Durations**	**40.99**	**Pitch Features**	**40.69**
C_F0K_LOGDIFF_ LASTPWLWIND100_ BASELN	26.34	P_PAU_DUR	20.68	C_F0K_LOGDIFF_ LASTPWLWIND100_ BASELN	20.76
C_F0K_LOGRATIO_ WIND80MIN_BASELN	15.44	PP_PAU_DUR	20.31	C_F0K_LOGRATIO_ SEGMAX_ WORDMAX	8.71
C_F0K_LOGRATIO_ LASTPWLWIND100_ BASELN	9.32	**Pitch Features**	**30.40**	C_DIST_ SEGPWLMAXLOC_ WORDSTART	5.68
Pause Durations	**48.91**	C_F0K_LOGDIFF_ LASTPWLWIND100_ BASELN	25.62	P_F0K_LOGRATIO_ SEGMIN_WORDMIN	5.54
		C_F0K_RATIOSHIFT_ SEGMAX_WORDMAX_ BASELN	3.75		
		CP_FOK_LOGDIFF_ MAXPWLWORD_ MAXPWL_P	1.03		

Case 1−PO		Case 2B−PO		Case 3B−PO	
P_PAU_DUR	31.89	**RMS Features**	**19.24**	**Other Features**	**34.82**
				C_WORD_WDPOS	34.82
PP_PAU_DUR	17.02	P_RMS_V_MAX_R	19.24	**Pause Features**	**13.07**
		Other Features	**9.42**	PP_PAU_DUR	13.07
		NATIVE	7.27	**Other Features**	**11.42**
		P_PERC_HALF	2.15	NATIVE	11.42

Table 4.4.7: Feature Usage for Sentence/Questions classification task, using only Previous Features.

The features shown above mostly consist of current pitch features and previous pause durations. Though the pitch features used in these cases are not exactly the same as their full featured counterparts, note the continued importance of the last F0 feature in the word, along with a variety of minimum and maximum F0 measures. Cases 2B−PO and 3B−PO use the class prior−correlated feature NATIVE, and the latter experiment also attempts to make use of the current word position inside the spurt. In that case the following decision is made:

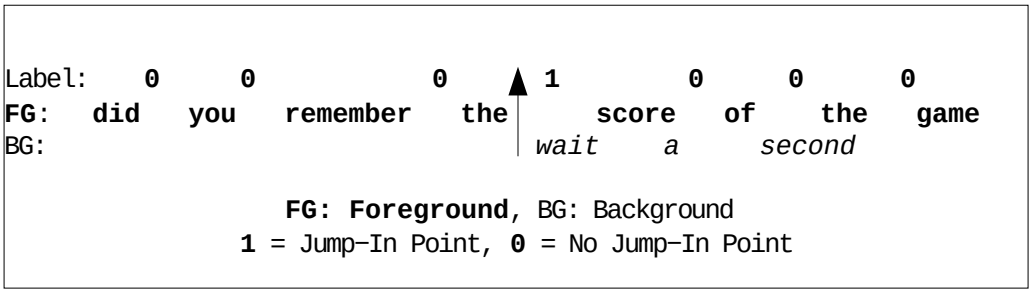
```
P_F0K_LOGRATIO_SEGMIN_WORDMIN < −0.33268:  0.5314 0.4686 PER
P_F0K_LOGRATIO_SEGMIN_WORDMIN >= −0.33268:  0.4512 0.5488 Q
      C_WORD_WDPOS < 6.5:  0.5964 0.4036 PER
      C_WORD_WDPOS >= 6.5:  0.4172 0.5828 Q
```

The decision here says that if the word has a high pitch and is the at least the seventh word in the spurt, it is classified as a period. This split indicates that given this prosodic context, longer spurts in the training set are generally questions.

4.5 Task 4 results

The remaining tasks deal with speaker interaction tasks, which are much more exploratory. In Task 4 prosody feature based decision trees along with the language model are used in predicting points of interruption. Specifically, a speaker's foreground speech is examined at the point where some other speaker(s) interrupts, as shown below. The word boundary immediately following the word where the foreground speaker was interrupted is labeled as a "Jump−In Point" and discriminative features that help in

prediction of this event are analyzed.



As opposed to previous tasks, where automatic recognition results are reported, experiments based on ASR words and word boundaries are not reported, since insertions and deletions may cause false interruption points thereby making training and scoring difficult and beyond the scope of this work.

In addition, it was found that the language model used in the previous tasks is not helpful at all in this task. This is an interesting result because it shows that despite having access to previous words relative to the interruption event, such knowledge does not help to predict the event itself, indicating that speakers are not waiting for particular words as indicative cues for interruption points.

Table 4.5.1 reports efficiencies for two cases of Task 4. All experiments for Task 4 exclude punctuation marks as these are "cheating" features. Efficiencies are shown here because the class priors are extremely skewed, with over 96% of the word boundaries belonging to the non−interruption class. Also, the inherent uncertainty in this task , along with lack of performance from the LM gives no accuracy above chance in the prior−adjusted case. Still, a non−zero efficiency on the downsampled data in promising, and those numbers are given here.

Case	Previous Only?	NAME included?	LM Efficiency	Tree Efficiency
1	No	No	0.00%	7.60%
2	Yes	No	0.00%	5.46%

Table 4.5.1 Efficiencies for the prediction of "Jump−In Points" using two feature sets

Table 4.5.2 shows feature usages for these two cases, and Figure 4.5.1 shows the corresponding trees.

```
************
   Case 1
************
F_PAU_DUR < 0.445:  0.5418 0.4582 0
|   P_PAU_DUR < 0.045:  0.6018 0.3982 0
|   |   SEX = m:  0.6246 0.3754 0
|   |   |   C_VOWEL_DUR < 8.5:  0.6386 0.3614 0
|   |   |   |   C_F0K_LOGRATIO_SEGMIN_BASELN < −0.15818:  0.6641 0.3359 0
|   |   |   |   C_F0K_LOGRATIO_SEGMIN_BASELN >= −0.15818:  0.5807 0.4193 0
|   |   |   |   |   C_F0K_LOGRATIO_SEGMIN_BASELN < 0.12874:  0.603 0.397 0
|   |   |   |   |   C_F0K_LOGRATIO_SEGMIN_BASELN >= 0.12874:  0.4073 0.5927 1
|   |   |   C_VOWEL_DUR >= 8.5:  0.5926 0.4074 0
|   |   SEX = f:  0.5029 0.4971 0
|   |   |   C_F0K_LOGDIFF_WORDMAX_BASELN < 4.5762:  0.5385 0.4615 0
|   |   |   |   C_F0K_LOGDIFF_WORDMAX_BASELN < 2.5689:  0.4044 0.5956 1
|   |   |   |   C_F0K_LOGDIFF_WORDMAX_BASELN >= 2.5689:  0.5483 0.4517 0
|   |   |   |   |   F_PAU_DUR < 0.045:  0.5682 0.4318 0
|   |   |   |   |   |   C_VOWEL_DUR < 10.5:  0.592 0.408 0
|   |   |   |   |   |   C_VOWEL_DUR >= 10.5:  0.4506 0.5494 1
|   |   |   |   |   F_PAU_DUR >= 0.045:  0.4123 0.5877 1
|   |   |   C_F0K_LOGDIFF_WORDMAX_BASELN >= 4.5762:  0.4153 0.5847 1
|   P_PAU_DUR >= 0.045:  0.3776 0.6224 1
F_PAU_DUR >= 0.445:  0.3188 0.6812 1
|   F_PAU_DUR < 1.5985:  0.422 0.578 1
|   |   P_PAU_DUR < 0.005:  0.4892 0.5108 1
|   |   |   C_VOWEL_DUR < 18.5:  0.4598 0.5402 1
|   |   |   C_VOWEL_DUR >= 18.5:  0.5943 0.4057 0
|   |   P_PAU_DUR >= 0.005:  0.306 0.694 1
|   F_PAU_DUR >= 1.5985:  0.2396 0.7604 1
************
   Case 2
************
P_PAU_DUR < 0.035:  0.5718 0.4282 0
|   C_F0K_LOGRATIO_LASTPWLWIND100_BASELN < 0.45292:  0.587 0.413 0
|   |   C_F0K_LOGRATIO_SEGMIN_WORDMIN < −0.38721:  0.6398 0.3602 0
|   |   C_F0K_LOGRATIO_SEGMIN_WORDMIN >= −0.38721:  0.5401 0.4599 0
|   |   |   C_F0K_LOGRATIO_SEGMIN_WORDMIN < −0.25901:  0.5973 0.4027 0
|   |   |   C_F0K_LOGRATIO_SEGMIN_WORDMIN >= −0.25901:  0.5103 0.4897 0
|   |   |   |   C_F0K_LOGRATIO_SEGMIN_WORDMIN < −0.044127:  0.5368 0.4632 0
|   |   |   |   C_F0K_LOGRATIO_SEGMIN_WORDMIN >= −0.044127:  0.4545 0.5455 1
|   C_F0K_LOGRATIO_LASTPWLWIND100_BASELN >= 0.45292:  0.5029 0.4971 0
|   |   C_F0K_LOGRATIO_SEGMIN_BASELN < 0.1858:  0.5181 0.4819 0
|   |   |   P_RMS_V_MAX_Z < −0.5064:  0.403 0.597 1
|   |   |   P_RMS_V_MAX_Z >= −0.5064:  0.5291 0.4709 0
|   |   C_F0K_LOGRATIO_SEGMIN_BASELN >= 0.1858:  0.308 0.692 1
P_PAU_DUR >= 0.035:  0.3381 0.6619 1
|   P_PAU_DUR < 0.525:  0.2757 0.7243 1
|   P_PAU_DUR >= 0.525:  0.4286 0.5714 1
|   |   P_PAU_DUR < 1.571:  0.5397 0.4603 0
|   |   |   C_F0K_LOGRATIO_SEGMIN_WORDMIN < −0.15997:  0.5973 0.4027 0
|   |   |   C_F0K_LOGRATIO_SEGMIN_WORDMIN >= −0.15997:  0.4305 0.5695 1
|   |   P_PAU_DUR >= 1.571:  0.349 0.651 1
```

Figure 4.5.1: Decision trees for Cases 1 and 2 for classification of Jump−In points, trained on equal class priors. Class labels are 0 (no interruption) and 1 (jump−in point) and probabilities are listed in that order.

Pause Durations	**56.76**	**Pitch Features**	**55.57**
F_PAU_DUR	29.21	C_F0K_LOGRATIO_ SEGMIN_WORDMIN	32.09
P_PAU_DUR	27.55	C_F0K_LOGRATIO_ LASTPWLWIND100_BASELN	19.88
Pitch Features	**15.66**	C_F0K_LOGRATIO_ SEGMIN_BASELN	3.60
C_F0K_LOGRATIO_ SEGMIN_BASELN	11.24	**Pause Durations**	**41.10**
C_F0K_LOGDIFF_ WORDMAX_ BASELN	4.42	P_PAU_DUR	41.10
Vowel Durations	**13.89**	**RMS Features**	**8.34**
C_VOWEL_DUR	13.89	P_RMS_V_MAX_Z	8..34
Other Features	**13.79**		
SEX	13.79		

Table 4.5.2 Feature usages (given in percentages) for Cases 1 and 2 of Task 4, a two class Jump−In point prediction experiment

In Case 1, where the full feature set is available, one sees that following and previous pause durations are clearly the most widely used features. As the following splits (top two levels from Figure 4.5.1) indicate, long durations for both of these pause features usually result in the prediction of a jump−in point:

```
F_PAU_DUR < 0.445:  0.5418 0.4582 0
       P_PAU_DUR < 0.045:  0.6018 0.3982 0
       P_PAU_DUR >= 0.045:  0.3776 0.6224 1
F_PAU_DUR >= 0.445:  0.3188 0.6812 1
       F_PAU_DUR < 1.5985:  0.422 0.578 1
       F_PAU_DUR >= 1.5985:  0.2396 0.7604 1
```

All but one of the splits shown above confirm our hypothesis that long pauses on either (or both) side of a word are generally good indicators for Jump−In points. In terms of previous pause durations, it appears that speakers wait for a suitable Jump−In point which is demarcated by a longer than usual pause duration. Since speakers do not have direct access to following features (although perhaps they can predict them to some extent), the inclusion of the F_PAU_DUR here is a reflection that when speakers are interrupted, they often pause and wait to hear what the interrupting speaker has to say, but this is not a feature that could be used in a real system.

After removing access to future features, the classifiers experience a precipitous fall in efficiency from Case 1 to Case 2. While depriving the decision tree of future features induces this relative degradation of 28.16%, Case 2 is more interesting, since

human participants and online machines only have access to this feature set.

In Case 2 , the decision tree makes much more use of current pitch features along with some energy features, as it must look elsewhere to compensate for the lack of following pauses. Only the "lower half" of the decision tree is considered here, conditioned on P_PAU_DUR >= 0.035, since this is where most of the Jump−In Points are classified:

```
P_PAU_DUR < 0.035:  0.5718 0.4282 0
P_PAU_DUR >= 0.035:  0.3381 0.6619 1
       P_PAU_DUR < 0.525:  0.2757 0.7243 1
       P_PAU_DUR >= 0.525:  0.4286 0.5714 1
              P_PAU_DUR < 1.571:  0.5397 0.4603 0
              C_F0K_LOGRATIO_SEGMIN_WORDMIN < −0.15997:  0.5973 0.4027 0
              C_F0K_LOGRATIO_SEGMIN_WORDMIN >= −0.15997: 0.4305 0.5695 1
                     P_PAU_DUR < 1.571:  0.5397 0.4603 0
                     P_PAU_DUR >= 1.571:  0.349 0.651 1
```

From the splits above it is clear that long previous pause durations indicate good interruption points. Also when `C_F0K_LOGRATIO_SEGMIN_WORDMIN` is greater than some number, boundaries are generally classified as Jump−In Points. This feature measures the log difference between the segment minimum and the current word minimum so it is always less then or equal to zero, with equality holding when the word minimum is the actual segment minimum. Apparently the closer the word minimum is to the segment pitch valley, the more likely someone is to interrupt at that point.

While the efficiencies of these experiments are not very high, any performance above chance in this experiment can be considered an accomplishment considering 1) the inherent uncertainty involved in this event detection and 2) that the LM with true words performs at chance. As mentioned in Section 4.1 , it is not known when a background speaker (or speakers) intend to interrupt the foreground participant, but decide not too. Our goals are to identify prosodic or lexical cues which other speakers use as markers for allowable interruption points, but their decision to interrupt or not blurs our ability to model these events with no uncertainty. Similarly, the models predict events that indicate an interruption in the background speaker without any knowledge of the background speaker’s prosodic or lexical features. Thus inherent uncertainties make this task a difficult one, but the use of prosodic features and pauses certainly help in

identifying possible "jump−in points", especially considering a language model trained on true words does no better than chance.

4.6 Task 5 results

Task 5 asks about how people jump in − do they change their prosody depending on whether or not somebody is already talking? The first word in each spurt is considered and classified as a "Jump−In Words", or words which are spoken during another person's speech, as shown below. As only the first word in each spurt is being considered, the data set is not a contiguous word stream, and the language model is not used in these experiments. Also, as seen in Task 4, experiments trained on ASR are not available in these dialogue tasks, since determining overlap regions for ASR is beyond the scope of this project.

```
Label:                1   EXC EXC     0  EXC EXC
FG:                 when was this?   oh  i'm sorry
BG: well  the other day i was working

                      FG: Foreground, BG: Background
                   1:Jump In Word, 0: Not a Jump−In word
                    EXC: Excluded Word (not classified)
```

For Task 5 an efficiency of 12.55% is attained for the prosodic feature based decision tree. Table 4.6.1 describes the features used in the model and Figure 4.6.1 shows the corresponding tree.

Task 5	
Feature	Usage
Pause Durations	**91.05**
P_PAU_DUR	42.00
F_PAU_DUR	32.12
PP_PAU_DUR	16.93
Pitch Features	**8.95**
C_F0K_LOGRATIO_ SEGMIN_BASELN	5.62
C_F0K_LOGDIFF_ SEGMAX_BASELN	2.43

Table 4.6.1 Feature usage for Task 5, a two class Jump−In word classification task

```
TREE WITH PROBABILITES:
**********************
 0.5 0.5 0
P_PAU_DUR < 1.535:  0.6872 0.3128 0
|   F_PAU_DUR < 1.5875:  0.7223 0.2777 0
|   |   PP_PAU_DUR < 1.175:  0.7461 0.2539 0
|   |   PP_PAU_DUR >= 1.175:  0.4877 0.5123 1
|   |   |   C_F0K_LOGRATIO_SEGMIN_BASELN < −0.22934:  0.5822 0.4178 0
|   |   |   C_F0K_LOGRATIO_SEGMIN_BASELN >= −0.22934:  0.406 0.594 1
|   F_PAU_DUR >= 1.5875:  0.3985 0.6015 1
|   |   F_PAU_DUR < 2.3105:  0.5561 0.4439 0
|   |   F_PAU_DUR >= 2.3105:  0.3552 0.6448 1
P_PAU_DUR >= 1.535:  0.3486 0.6514 1
|   F_PAU_DUR < 0.925:  0.4007 0.5993 1
|   |   P_PAU_DUR < 3.365:  0.475 0.525 1
|   |   |   PP_PAU_DUR < 2.513:  0.5042 0.4958 0
|   |   |   |   C_F0K_LOGRATIO_SEGMIN_BASELN < 0.11873:  0.5227 0.4773 0
|   |   |   |   |   C_F0K_LOGDIFF_SEGMAX_BASELN < 4.888:  0.5425 0.4575 0
|   |   |   |   |   |   P_PAU_DUR < 1.7525:  0.6288 0.3712 0
|   |   |   |   |   |   P_PAU_DUR >= 1.7525:  0.5155 0.4845 0
|   |   |   |   |   |   |   C_F0K_LOGRATIO_SEGMIN_BASELN < −0.39073: 0.6321 0.3679 0
|   |   |   |   |   |   |   C_F0K_LOGRATIO_SEGMIN_BASELN >= −0.39073: 0.4854 0.5146 1
|   |   |   |   |   |   |   |   PP_PAU_DUR < 0.035:  0.4514 0.5486 1
|   |   |   |   |   |   |   |   PP_PAU_DUR >= 0.035:  0.5546 0.4454 0
|   |   |   |   |   C_F0K_LOGDIFF_SEGMAX_BASELN >= 4.888:  0.3408 0.6592 1
|   |   |   |   C_F0K_LOGRATIO_SEGMIN_BASELN >= 0.11873:  0.3914 0.6086 1
|   |   |   PP_PAU_DUR >= 2.513:  0.3305 0.6695 1
|   |   P_PAU_DUR >= 3.365:  0.363 0.637 1
|   F_PAU_DUR >= 0.925:  0.2718 0.7282 1
```

Figure 4.6.1 Decision trees for classification of Jump−In Words, trained on equal class priors. Class labels are 0 (first word in silence) and 1 (first word in other speech) and probabilities are listed in that order.

Pause durations, both previous and following are used primarily in classifying Jump−In words. The tree splits generally indicate that shorter previous pauses imply starting in silence. Note that by definition, the shortest previous pause possible in this experiment is 0.5s , since all data words are at the beginning of a spurt. Shorter previous pause durations probably mean that a speakers simply paused during his or her utterance, and continued after a short time. Longer previous pause durations reflect more isolated words, and seem to be more indicative of jumping in during another speaker's utterance.

Pitch features are also used in classifying Jump−In Words, as seen in the following decision:

```
C_F0K_LOGRATIO_SEGMIN_BASELN < −0.39073: 0.6321 0.3679 0
C_F0K_LOGRATIO_SEGMIN_BASELN >= −0.39073: 0.4854 0.5146 1
```

In this case, low segment pitch minima are used as cues to indicate starting in silence (0) whereas a high pitch indicates starting during another speaker's speech. This result concurs with theory; when attempting to grab the floor, it has been shown that

speakers will start with high energy and high pitch (French and Local, 1983). While energy features are not included in the tree above, the use of pitch features is promising.

4.7 Cross task comparisons and overall discussions

The results given above point to many common threads across tasks and cases. First, degradation caused by ASR words and word boundaries strongly affects accuracy performance, especially in Task 2, where the LM was making many gains for free. Prosodic feature based classifiers generally tend to be more robust to word recognition errors than the LM since the LM depends on word identities and prosody only on word boundaries.

With regard to feature usages, vowel and pause durations are extremely important in both punctuation and dialog classification tasks. Across tasks, the use of pitch and energy features increases tremendously in cases where the classifiers are allowed to only see the previous features. Decision trees utilize information in pitch features when future features, especially following pause features, are not available. This point is extremely important because in real time systems, future features are not available, so the ability to automatically derive and use prosodic features is crucial.

Also, an attempt was made to correlate word error rates with performance degradation in all models. Though a clear relationship was not found between these variables, in the process it was noted that testing on speakers with low word error rates will undoubtedly increase model performance, but it is still better to train on as much data as possible rather than on one specific speaker alone. Additionally, and perhaps more interestingly, it was found that certain speakers have prosodic models which perform better than their language models. The variability between speakers in this sense indicates that some meeting participants probably have more consistent and predictable prosodic cues than others, and that speaker specific modeling of these tasks should take this fact into account.

In Task 3, a number of interesting observations can be seen in the sentence/question distinction task. First, the lack of vowel durations used in the trees distinguishes this task from Tasks 1 and 2; our results show that question and sentence ends have little or no difference in terms of vowel durations. Second, distinguishing these

events does not require future knowledge, and depriving the trees of future features actually increases performance. The implication here is that sentence and question ends are not isolated events, but rather very much dependent on the previous sentence context.

In Task 4, where classifiers attempt to predict Jump−In Points, it is again shown that pitch features are extremely useful in cases where systems don't have access to future features. Specifically, it was found that the combination of long previous pause durations along with pitch drops yield prosodically acceptable places for other speakers to jump in and interrupt the current speaker. When these speakers do interrupt, Task 5 shows that they often start with an elevated pitch. Results also indicate that the longer the pause before a speaker starts speaking the more likely he or she is to interrupt , rather than start in silence.

Finally, results across tasks pointed to the fact training on manually transcribed, forced alignment based prosodic feature sets, while testing on ASR based features, yielded small performance degradations. This fact is extremely important because it points to the possibility of training on a data set over which the computationally expensive recognition process had not been performed, even for models which will be used for recognition output word streams and ASR based prosodic data. Thus, human transcripts, which can be assumed to exist already since recognition systems require manual transcription for their training, can be used in training prosodic models, without requiring the additional steps of word recognition and feature extraction on the entire training data set.

5. Conclusion

This project aimed to use automatically derived prosodic features, in conjunction with lexical features, to classify various punctuation, dialog and interaction events in the Meeting Recorder corpus. The use of prosodic cues in the classification of these events were found to be extremely useful. Pause and vowel durations were often selected by prosodic decision trees, especially when the classifier had access to both future and past information. In the "online" case, where the decision tree only had access to features occurring before the current event location, pitch features played an important role, especially in punctuation Tasks 1 (sentence/non−sentence) and 2 (sentence/non−sentence/disfluency). Pitch was also an extremely useful cue in distinguishing declarative sentence ends from question ends.

In addition, the effect of using recognition output rather than true words for punctuation tasks was analyzed, and it was found that decision trees using ASR based feature sets were quite robust to alignment boundary and word errors. These trees always performed above chance, and further more usually *outperformed language models*, which degraded less gracefully when using an error ridden word stream. It was also found that training on forced−alignment based features and testing on ASR words yielded performances comparable to experiments which trained ASR based feature sets. This is both numerically and practically important, since it means that running ASR on, and computing features for training data can be avoided. In all cases, a combination model using both words and prosodic features almost always outperformed either model on its own.

In more exploratory experiments, prosody was used to help prediction of interruption points and whether the first word in a spurt was in speech or silence. In the former task, a language model was found to not be at all helpful in predicting points in time at which other speakers jump in to interrupt the foreground speaker. Prosodic decision trees, however, achieved efficiencies that were large enough to sugges that pitch and pause features are useful in discriminating such locations. Trees were also able to help distinguish whether a speaker’s first word in a spurt is in silence or another person’s speech. Speakers generally start at high pitch levels when wanting to take the floor.

A long term goal of this work is a comprehensive model of conversational speech that would allow for the automatic recovery of pragmatic and semantic structures, and eventually the creation of conversational agents that behave naturally as meeting participants. This study provides a good starting point in assessing the importance of prosodic and lexical cues for this purpose, but more work is necessary to determine the effect of different meeting types and word errors on prosodic feature sets.

Prosodic feature based classification trees can also provide extremely useful insights into the inherent cues used by humans to convey and detect these events. The extensive database developed in this work can offer a clear picture of the complex interactions of pause, pitch, and energy features within both the conversational and dialog domains; these observations could be very useful for linguistic theory.

Finally, robotic meeting participants, as described in (Y. Matsusaka, 2001) could serve as meeting proctors, note takers, or facilitators. In order for such machines to function naturally in a meeting setting, they must be able to predict and respond to both pragmatic and structural events, which are not available in word transcripts. It is hoped that the research conducted in this study provides a useful starting point for the use of prosody in such domains.

There are many potential areas for future work. For example, the inclusion of fully automatic segmentations is desired, since hand−adjusted ones were used in this study. Similarly, automatic speech recognition systems need to be developed specifically for the meeting domain, accounting for the effect of multiple speakers, room acoustics, informal speech patterns, and language usage. In this study the recognizer was not tuned to the specific acoustic or language characteristics of the Meeting Recorder corpus, partially because of the lack of data but also to expedite experiments. A better recognizer would increase performance in both the prosodic and language models. As seen in Chapter 4, prosodic models trained on true words and tested on ASR output performed better than those trained and tested on ASR words. As better recognizers approach this upper bound of human transcribed words, performance of prosodic trees can therefore also be expected to increase.

In terms of the classification experiments, it would be beneficial to more critically analyze the specific role of word errors in the prosodic trees, language models, and

combination classifiers, and to more fully understand the role of these errors within each of these frameworks. A more comprehensive study of the effect of word error rates to specific speaker performance would also be of great value. Understanding the role of word errors in both of these contexts could be used to determine if certain models or features are better in various noise conditions or speakers. Perhaps word error rates themselves could be included as predictive features in the future.

This work has shown that prosody is a powerful information source for the classification of a variety of types of events in the meeting domain. Such event detection should be fully exploited in a machine transcription annotation framework, where ASR word streams require punctuation and dialogue markings for both readability and deeper machine understanding. The ability to accomplish this fully automatically would have major implications in the field of automatic speech processing and ultimately influence the fields of automatic understanding, information extraction, summarization, human−machine interaction and rich transcription.

6. References

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth and Brooks, Pacific Grove, CA, 1984.

J. Buckow, V. Warnke, R. Huber, A. Batliner, E. Noth, and H. Niemann. "Fast and Robust Features for Prosodic Classification." In V. Matousek, P. Mautner, J. Ocelikova, and P. Sojka, editors, *Proc. Workshop on TEXT, SPEECH and DIALOG (TSD 99)*, volume 1692 of Lecture Notes for Artificial Intelligence, pages 193 198, Berlin, September 1999. Springer Verlag.

W. Buntine, and Caruana, R. (1992). *Introduction to IND Version 2.1 and Recursive Partitioning.* Moffett Field, CA.

E. Couper−Kuhlen and M. Selting, editors. *Prosody in Conversation.* Cambridge University Press, Cambridge, 1996.

V. Digalakis, P. Monaco and H. Murveit. "Genones: Generalized Mixture Tying in Continuous Hidden Markov Model−Based Speech Recognizers." in *IEEE Transactions Speech and Audio Processing*, July 1996, pp. 281−289.

Entropic Research Laboratory, Washington, D.C. *ESPS Version 5.0 Programs Manual*, 1993.

P. French and J.K. Local 1983. "Turn−competitive incomings." *Journal of Pragmatics*, 7: 17−38.

P. Heeman and J. Allen. "Intonational boundaries, speech repairs, and discourse markers: Modeling spoken dialog," *in Proc. ACL/EACL*, Madrid, 1997.

W. Hess. *Pitch Determination of Speech Signals: Algorithms and Devices.* Springer−Verlag, Berlin: 1983.

G. Jefferson (1973). "A case of precision timing in ordinary conversation: Overlapping tag−positioned address terms in closing sequences." *Semiotica*, 3, 47−96.

D. Jurafsky, Shriberg, E., Fox, B. & Curl, T. (1998). "Lexical, Prosodic, and Syntactic Cues for Dialog Acts." *Proceedings of ACL/COLING 98 Workshop on Discourse Relations and Discourse Markers*, pp. 114−120, Montreal.

M. Mast, R. Kompe, S. Harbeck, A. Kießling, H. Niemann, E. N¨ oth, E. G. Schukat−Talamazzini, and V. Warnke. "Dialog act classification with the help of prosody," in H. T. Bunnell and W. I dsardi, editors, *Proc. ICSLP*, vol. 3, pp. 1732 1735, Philadelphia, 1996.

Y. Matsusaka, S. Fujie, and T. Kobayashi. "Modeling of conversational strategy for the robot participating in the group conversation." In P. Dalsgaard, B. Lindberg, H. Benner, and Z. Tan, editors, *Proc. EUROSPEECH*, vol. 3, pp. 2173 2176, Aalborg, Denmark, 2001.

N. Morgan, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke. "The ICSI Meeting Project," in J. Allan, editor, *Proc. HLT 2001*, pp. 246 252, San Diego, 2001. Morgan Kaufman.

T. Pfau, D. P. W. Ellis, and A. Stolcke, "Multispeaker speech activity detection for the ICSI Meeting Recorder," *in Proceedings IEEE Automatic Speech Recognition and Understanding Workshop*, Madonna di Campiglio, Italy, Dec. 2001.

B. Secrest, G. Doddington. "An integrated pitch tracking algorithm for speech systems," i*n Proceedings*

of 1983 ICASSP, Boston, MA, vol. 3, pp. 1352−1355, 1983.

E. Shriberg, R. Bates, and A. Stolcke. "A prosody−only decision−tree model for disfluency detection," in G. Kokkinakis, N. Fako−takis, and E. Dermatas, editors, *Proc. EUROSPEECH*, vol. 5, pp. 2383 2386, Rhodes, Greece, 1997.

E. Shriberg, A. Stolcke, and D. Baron. "Observations on overlap: Findings and implications for automatic processing of multi−party conversation." In P. Dalsgaard, B. Lindberg, H. Benner, and Z. Tan, editors, *Proc. EUROSPEECH*, vol. 2, pp. 1359 1362, Aalborg, Denmark, 2001.

E. Shriberg, A., Stolcke, and D. Baron. "Can Prosody Aid the Automatic Processing of Multi−Party Meetings? Evidence from Predicting Punctuation, Disfluencies, and Overlapping Speech." *Proc. ISCA Tutorial and Research Workshop on Prosody in Speech Recognition and Understanding*, Red Bank, NJ, 2001.

E. Shriberg, A. Stolcke, D. Hakkani−Tur, and G. Tur. "Prosody−based automatic segmentation of speech into sentences and topics." *Speech Communication*, 32(1−2):127 154, 2000. Special Issue on Accessing Information in Spoken Audio.

K. Sonmez, E. Shriberg, L. Heck, and M. Weintraub. "Modeling dynamic prosodic variation for speaker verification." In R. H. Mannell and J. Robert−Ribes, editors, *Proc. ICSLP*, vol. 7, pp. 3189 3192, Sydney, 1998. Australian Speech Science and Technology Association.

A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. Rao Gadde, M. Plauche, C. Richey, E. Shriberg, K. Sonmez, F. Weng, and J. Zheng. "The SRI March 2000 Hub−5 conversational speech transcription system," *in Proceedings NIST Speech Transcription Workshop*, College Park, MD, 2000.

A. Stolcke, E. Shriberg, R. Bates, M. Ostendorf, D. Hakkani, M. Plauche, G. Tur, and Y. Lu. "Automatic detection of sentence boundaries and disfluencies based on recognized words," in R. H. Mannell and J. Robert−Ribes, editors, P*roc. ICSLP*, vol. 5, pp. 2247 2250, Sydney, 1998. Australian Speech Science and Techology Association.

D. Talkin. "A robust algorithm for pitch tracking (RAPT)," *in Speech Coding and Synthesis*, W. B. Kleign and K. K. Paliwal, eds., Elsevier Science, Amsterdam, pp. 495−518, 1995.

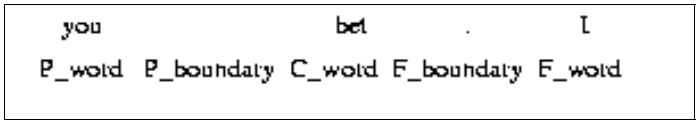
A. Waibel, M. Bett, M. Finke, and R. Stiefelhagen. "Meeting Browser: Tracking and summarizing meetings," *in Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, pp. 281 286, Lansdowne, VA, 1998. Morgan Kaufman.

Appendix

Feature Descriptions

KEY:
PWL= Piecewise−linear fitted
SW=Switchboar**d**
WINLENGTH= Number of frames in window, Values = 10,20,50,80,100

Region Key:



P_ = Previous
C_ = Current
F_ = Following

Feature	Description
Pause Features	
PP_PAU_DUR	Pause before previous word
{P_,F_}PAU_DUR	Pause duration before or after word
Vowel Durations	
{P_,C_,F_}TRIVOWEL_DUR_N	Normalized maximum trivowel duration in word
{P_,C_,F_}TRIVOWEL_DUR_Z	Z−score maximum trivowel duration in word
{P_,C_,F_}VOWEL_DUR	Maximum vowel duration in word
{P_,C_,F_}VOWEL_DUR_N	Normalized maximum vowel duration in word (by SW stats)
{P_,C_,F_}VOWEL_DUR_Z	Z−score maximum vowel duration in word (by SW stats)
F0 Features	
{P_,C_,F_}F0K_DIFF_ FIRSTPWLWORD_BASELN	Difference between first PWL value of word and baseline
{P_,C_,F_}F0K_LOGDIFF_ FIRSTPWLWORD_BASELN	Log of difference between first PWL value of word and baseline
{P_,C_,F_}F0K_LOGRATIO_ FIRSTPWLWORD_BASELN	Log of ratio between first PWL value of word and baseline
{P_,C_,F_}F0K_DIFF_ LASTPWLWORD_BASELN	Difference between last PWL value of word and baseline

Feature	Description
{P_,C_,F_}F0K_LOGDIFF_ LASTPWLWORD_BASELN	Log of difference between last PWL value in word and baseline
{P_,C_,F_}F0K_LOGRATIO_ LASTPWLWORD_BASELN	Log of ratio between last PWL value in word and baseline
{P_,C_,F_}F0K_LOGDIFF_ SEGMAX_WORDMAX	Log of difference between segment PWL max and word PWL max
{P_,C_,F_}F0K_LOGRATIO_ SEGMAX_WORDMAX	Log of ratio between segment maximum PWL value and word max PWL
{P_,C_,F_}F0K_LOGDIFF_ SEGMIN_WORDMIN	Log of difference between segment minimum PWL value and baseline for word
{P_,C_,F_}F0K_LOGRATIO_ SEGMIN_WORDMIN	Log of ratio between segment minimum PWL value and word min PWL
{P_,C_,F_}F0K_LOGDIFF_ LASTPWLWIND**WINLENGTH**_BASELN	Log of difference between last PWL value in window of length **WINLENGTH** for word and baseline
{P_,C_,F_}F0K_LOGRATIO_ LASTPWLWIND**WINLENGTH**_BASELN	Log of ratio between last PWL value in window of length **WINLENGTH** for word and baseline
{P_,C_,F_}F0K_LOGDIFF_ WIND**WINLENGTH**MAX_BASELN	Log of difference between max PWL value in window of length **WINLENGTH** for word and baseline for follwing word
{P_,C_,F_}F0K_LOGRATIO_ WIND**WINLENGTH**MAX_BASELN	Log of ratio between max PWL value in window of length **WINLENGTH** and baseline
{P_,C_,F_}F0K_LOGDIFF_ WIND**WINLENGTH**MIN_BASELN	Log of difference between min PWL value in window of length **WINLENGTH** for word and baseline
{P_,C_,F_}F0K_LOGRATIO_ WIND**WINLENGTH**MIN_BASELN	Log of ratio between min PWL value in window of length **WINLENGTH** for word and baseline
{P_,C_,F_}F0K_LOGDIFF_ WORDMAX_BASELN	Log of difference between word maximum and baseline
{P_,C_,F_}F0K_LOGRATIO_ WORDMAX_BASELN	Log of ratio between word max PWL value and baseline
{P_,C_,F_}F0K_LOGDIFF_ WORDMIN_BASELN	Log of difference between word minimum and baseline
{P_,C_,F_}F0K_LOGRATIO_ WORDMIN_BASELN	Log of ratio between word min PWL value and baseline
{P_,C_,F_}F0K_RATIOSHIFT_ SEGMAX_WORDMAX_BASELN	Difference between PWL segment maximum and baseline divided by difference between PWL word max
{P_,C_,F_}F0K_RATIOSHIFT_ SEGMIN_WORDMIN_BASELN	Difference between PWL segment minimum and baseline divided by difference between PWL word min
PC_F0K_DIFF_ P−LASTPWLWORD_C−FIRSTPWLWORD	Difference between last PWL value in previous word and first PWL value in current word

Feature	Description
PC_F0K_LOGDIFF_ P−LASTPWLWORD_C−FIRSTPWLWORD	Log of difference between last PWL value in previous word and first PWL value in current word
PC_F0K_LOGRATIO_ P−LASTPWLWORD_C−FIRSTPWLWORD	Log of ratio between last PWL value in previous word and first PWL value in current word
PC_F0K_DIFF_ P−LASTSLOPE_C−FIRSTSLOPE	Difference between last slope in previous word and first slope in current word
PC_F0K_RATIOSHIFT_P−LASTPWLWORD_C−FIRSTPWLWORD_F0KBASELN	Difference of last PWL value of previous word and baseline divided by difference of first PWL of current word and baseline
CP_FOK_DIFF_ MAXPWLWORD_MAXPWL_P−WORD	Difference between maximum PWL value of current word and maximum PWL value of previous word
CP_FOK_LOGDIFF_ MAXPWLWORD_MAXPWL_P−WORD	Log of difference between maximum PWL value of current word and maximum PWL value of previous word
CF_F0K_DIFF_ LASTPWLWORD_F−FIRSTPWLWORD	Difference between last PWL value in current word and first PWL value in word
CF_F0K_DIFF_ LASTSLOPE_F−FIRSTSLOPE	Difference between last slope in current word and first slope in word
CF_F0K_LOGDIFF_ LASTPWLWORD_F−FIRSTPWLWORD	Log of difference between last PWL value in current word and first PWL value in word
CF_F0K_LOGRATIO_ LASTPWLWORD_F−FIRSTPWLWORD	Log of ration between last PWL value in current word and first PWL value in word
CF_FOK_DIFF_ P−MAXPWLWORD_ F−MAXPWLWORD	Difference between maximum PWL value of current word and maximum PWL value of word
CF_FOK_LOGDIFF_ P−MAXPWLWORD_ F−MAXPWLWORD	Log of difference between maximum PWL value of current word and maximum PWL value of word
CF_F0K_RATIOSHIFT_ LASTPWLWORD_F− FIRSTPWLWORD_F0KBASELN	Difference of last PWL value of current word and baseline divided by difference of first PWL of word and baseline
RMS Features	
{P_,C_,F_}RMS_MAX_R	Ratio of maximum RMS value of all frames in word to the mean RMS for all frames
{P_,C_,F_}RMS_MAX_Z	Z−Score of maximum RMS value of all frames in word to the mean RMS for all frames
{P_,C_,F_}RMS_MIN_R	Ratio of minimum RMS value of all frames in word to the mean RMS for all frames
{P_,C_,F_}RMS_MIN_Z	Z−Score of minimum RMS value of all frames in word to the mean RMS for all frames
{P_,C_,F_}RMS_V_MAX_R	Ratio of maximum RMS value of voiced frames in word to the mean RMS for voiced frames

Feature	Description
{P_,C_,F_}RMS_V_MAX_Z	Z−Score of maximum RMS value of voiced frames in word to the mean RMS for voiced frames
{P_,C_,F_}RMS_V_MIN_R	Ratio of minimum RMS value of voiced frames in word to the mean RMS for voiced frames
{P_,C_,F_}RMS_V_MIN_Z	Z−Score of minimum RMS value of voiced frames in word to the mean RMS for voiced frames
Octaval Features	
{P_,C_,F_}PERC_DOUB	Percentage of frames assumed to be doubled in word
{P_,C_,F_}PERC_HALF	Percentage of frames assumed to be halved in word
Sentence Boundary Features	
{P_,F_}Q	boundary a question mark?
{P_,F_}S	boundary a sentence boundary?
{P_,F_}S_TYPE	Type of boundary (i.e., sentence, incomplete, fluent)
{P_,C_,F_}DFIP	boundary a disfluent or incomplete sentence boundary?
Special Word Features	
{P_,C_,F_}IN_CC	word a coordinated conjunction?
{P_,C_,F_}IN_BA	word in backchannel?
{P_,C_,F_}IN_DM	word a discourse marker?
{P_,C_,F_}IN_FP	word a filled pause?
{P_,C_,F_}IN_RP	word a repeat?
{P_,C_,F_}IN_SW_ALL	word a special word, i.e., in 5 categories above?
{P_,C_,F_}IN_SW_BACCDM	word a backchannel, coordinated conjunction or discourse marker?
{P_,C_,F_}IN_SW_CCDM	word a coordinated conjunction or discourse marker?
{P_,C_,F_}IN_SW_FPRP	word a filled pause or repeat?
{P_,C_,F_}WORD_FREQ	Word frequency of word in SW
Contextual Features	
MIC	Microphone type
MTYPE	Meeting type
NAME	Speaker name
NATIVE	Native American−English speaker?
SEX	Gender