Can Prosody Aid the Automatic Processing of Multi-Party Meetings? Evidence from Predicting Punctuation, Disfluencies, and Overlapping Speech

Elizabeth Shriberg[1,2] *Andreas Stolcke*[1,2] *Don Baron*[3,2]

[1]International Computer Science Institute, Berkeley, CA
[2]SRI International, Menlo Park, CA
[3]University of California at Berkeley, Berkeley, CA
{ees,stolcke}@speech.sri.com, dbaron@icsi.berkeley.edu

Abstract

We investigate whether probabilistic modeling of prosody can aid various automatic labeling tasks essential for processing of multi-party meetings. Task 1, *automatic punctuation*, seeks to classify sentence boundaries and disfluencies. Task 2, *jump-in points*, predicts locations within foreground speech at which background speakers start talking; Task 3, *jump-in words*, examines characteristics of the speech they use to do so. Data are from the ICSI Meeting Recorder corpus. To infer inherent cues, analyses are based on close-talking microphone signals and recognizer forced alignments. As a generous baseline for word-level cues, we compare prosodic models to those of a language model given the true words. Results for Task 1 show prosody reduces classification error by 10% relative over the cheating language model; furthermore when this task is run in "online" mode the prosodic model degrades less than does the language model. For Task 2, the language model provides no information, while the prosodic model reduces entropy by 13% over chance. For Task 3, a prosodic model reduces entropy by 25% over chance. Analyses also show interesting prosodic patterns, which differ over tasks. Task 1 uses cues similar to those for Switchboard (but not Broadcast News) data. Task 2 predicts jump-in points that *look prosodically* like sentence boundaries but that are not actually such boundaries. And Task 3 shows that speakers "raise" their voice when starting during another's talk, compared to starting during silence. These results provide evidence that prosodic modeling can be of use for the automatic processing of meetings. Further results and implications for future automatic meeting processing systems are discussed.

1. Introduction

Natural multi-party meetings have garnered recent interest in the speech recognition and understanding communities [1, 2]. For meeting recordings and automatic transcripts to be truly useful, we need to be able to automatically annotate the data with a variety of labels that are either taken for granted or absent in written documents. These include, among others, the locations of sentence boundaries, the locations of disfluent self-interruptions (including incomplete sentences), and the extents of speaker turns.

Sentence segmentation and disfluency modeling apply to both monologues and dialogues, and have been the focus of much recent work in speech processing [3, 4, 5, 6]. Turn-taking, on the other hand, is an inherent property of multi-party dialogues that has received much attention in the linguistics literature (especially in conversation analysis, cf. [7, 8, 9, 10]), but relatively little in the field of speech processing. Spontaneous (as opposed to formal or scripted) multi-party speech is replete with *overlaps*, or regions during which more than one person is speaking. In a preliminary study of overlaps in meetings and telephone conversations, we found that between 9-17% of all words and over 50% of all "spurts" (stretches of speech with no pauses longer than 0.5 seconds) were overlapped [11].

Our long-term goal is a comprehensive model of conversational speech that would allow automatic recovery of pragmatic and semantic structures, and eventually the creation of conversational agents that behave naturally as meeting participants. First research prototypes of machines that function as conversational participants are being developed [12], and such agents will have to master the subtleties of appropriate turn-taking, backchanneling, and interruptions. In the present paper, we investigate if and how prosodic cues can serve toward this goal, by focusing on a few specific prediction and classification tasks based on meeting data.

First, extending work that processed only one speaker at a time [5, 6], we examine "automatic punctuation" (sentence boundary and disfluency detection) in the meeting domain. Second, we ask: do speakers "jump in" during someone else's talk at random locations, or in certain prosodic contexts? Third, does prosody differ when jumping in during another party's speech ("jump-in words") versus when starting during silence? We also examine how prosodic models compare to purely word-based language models, and investigate whether the two can be combined for better accuracy.

2. Method

2.1. Data and annotations

We analyzed data from multi-party meetings collected as part of the ICSI Meeting Project [2]. We drew data from two types of ongoing meetings, "Meeting Recorder" (MR) and "Robustness" (ROB), with between 4 and 8 participants each. MR meetings are discussions about the Meeting Project itself, and are highly democratic. ROB meetings cover research in a particular field and are dominated by a single person (the group leader). The amount of data from each source is summarized in Table 1. Meetings were transcribed by human labelers and annotated for sentence boundaries, incomplete sentences, and disfluencies.

For modeling purposes we split our corpus into a training set and a disjoint test portion. The test portion consisted of one MR and one ROB meeting, chosen so as to make the total amount of data (number of words) in the test set about 30% of the total for each meeting type. Training and test sets did overlap in speakers, since the bulk of the data is from speak-

Table 1: Types and amounts of data used in study: Meeting Recorder (MR) and Robustness (ROB) meetings. Speech duration includes only speech regions (excludes long silent regions) but counts overlapped speech multiple times. The notion of speech "spurt" is defined in Section 2.3.

	MR	ROB
Meetings	5	3
Speech duration	7.7h	4.8h
Transcribed words	60,403	32,384
Speech spurts	5,688	4,100

ers that participate in almost all meetings. We consider this not atypical of real-world application scenarios where meetings often involve recurring participants.

2.2. Transcript alignment

The Meeting Recorder project collects simultaneous signals from both close-talking and far-field microphones. For all experiments reported here the close-talking microphone signals were used exclusively, since the goal was to use the highest-quality features possible to capture the inherent properties of the speech. The quality on the far-field microphone signals for these data is severely degraded—particularly in the regions of overlap. Furthermore, although active research on far-field processing may lead to significant improvements, it is not unreasonable to assume that a future application could use close-talking microphones if high quality speech is an issue in the interim.

To obtain word times and other time-related features, the word transcripts were automatically aligned to the waveforms using SRI's large vocabulary conversational speech recognizer, originally developed for the NIST Hub-5 domain [13]. In this study we based all feature extraction on knowledge of the correct words, again because our interest is in the properties of the speech itself, not that of any particular recognition system.

In future work, we plan to carry out similar experiments based on time alignments of automatically recognized words. Recognition is poor in some cases, especially for lapel microphones [2], and would severely degrade lexical features and the language model. Prosodic features that are based on word and phone boundaries would also be degraded. However, some prosodic features that are used in our models do not depend on word boundary information, but rather on a computation over a window of speech frames; others, such as pause boundaries, should be reasonably accurate for most current recognition systems.

2.3. Spurt units

We base our investigation of overlap on a unit of speech above the word level that can be defined without reference to theoretical constructs from linguistics. We divide each speakers' speech into contiguous stretches called "spurts", where a spurt is defined as a stretch of contiguous speech containing no pauses longer than 0.5 seconds. Thus, spurts can be defined without reference to the words, have the same definition regardless of speaking style, and are available in speech recognition output via the word time alignments.

2.4. Task definitions

Given these time alignments and annotations, we define three classification tasks:

Task 1: "Automatic Punctuation" Classify each word boundary into one of three categories: complete sentence end, incomplete sentence end or disfluent interruption point, or sentence-internal fluent boundary.

Task 2: "Jump-in points" Classify each word boundary as to whether or not any other speaker "jumped in" at that location (specifically, whether or not any other speaker's spurt started during the current word or within any spurt-internal pause that immediately precedes that word.)

Task 3: "Jump-in words" Within the set of all spurt-initial words, classify each spurt-initial word as to whether it is a "jump-in" (starts during ongoing speech by another talker) or not (starts during silence).

Examples of boundary types in each of the classes are given in Figure 1, showing the temporal arrangement of spurts from two speakers. (The figure also shows automatically stylized pitch contours to illustrate the F0-based processing described below.)

Note that for purposes of this study, in which we use spurt units based solely on observable pause lengths, we use the neutral terms "jump-in points" and "jump-in words" to specify overlap onsets of spurts. This is to avoid any confusion with terminology taken from the turn-taking literature that refers to turn units, since there is not a one-to-one mapping between spurts and turns.

Automatic punctuation is important for a variety of downstream processing tasks in both speech recognition and understanding [6], including language modeling, parsing, delimiting of dialog acts, and marking of semantic boundaries for information extraction and retrieval. This task has obvious applications for both offline and online processing. Thus, we explore both an offline and a forward-only model for this task.

The speaker overlap prediction tasks differ formally from punctuation prediction. The latter involves prediction of "hidden" locations in the word stream output by a speech recognizer. Speaker overlaps, on the other hand, are directly observable in the current data, since by definition overlaps occur at points of simultaneous speech on more than one of the (individually recorded) channels. What we are interested in is finding out whether there is any correlation between the onset of overlaps and prosodic features (of both the *overlappee* and the *overlapper*). Thus for both of these tasks we consider only classifiers that have access to features extracted from speech up to the point of interest.

Both punctuation and overlap have been discussed in the literature as correlating with prosodic cues. For example, past computational work has discussed prosodic features for sentence boundaries as well as disfluency boundaries [3, 14, 15, 6]. Past work in conversation analysis, discourse analysis, and linguistics has shown prosody to be a useful cue in turn-taking behavior [7, 8, 9, 10]. Such studies suggest a potential contribution from prosody for our tasks, but to our knowledge the tasks have not yet been explored within a framework of automatic feature extraction for multi-party meetings.

For all tasks, the processing is round robin, considering each speaker in turn as the foreground speaker, and treating all others as background speakers. For each task, we extract prosodic cues completely automatically, based on recognizer forced alignments.

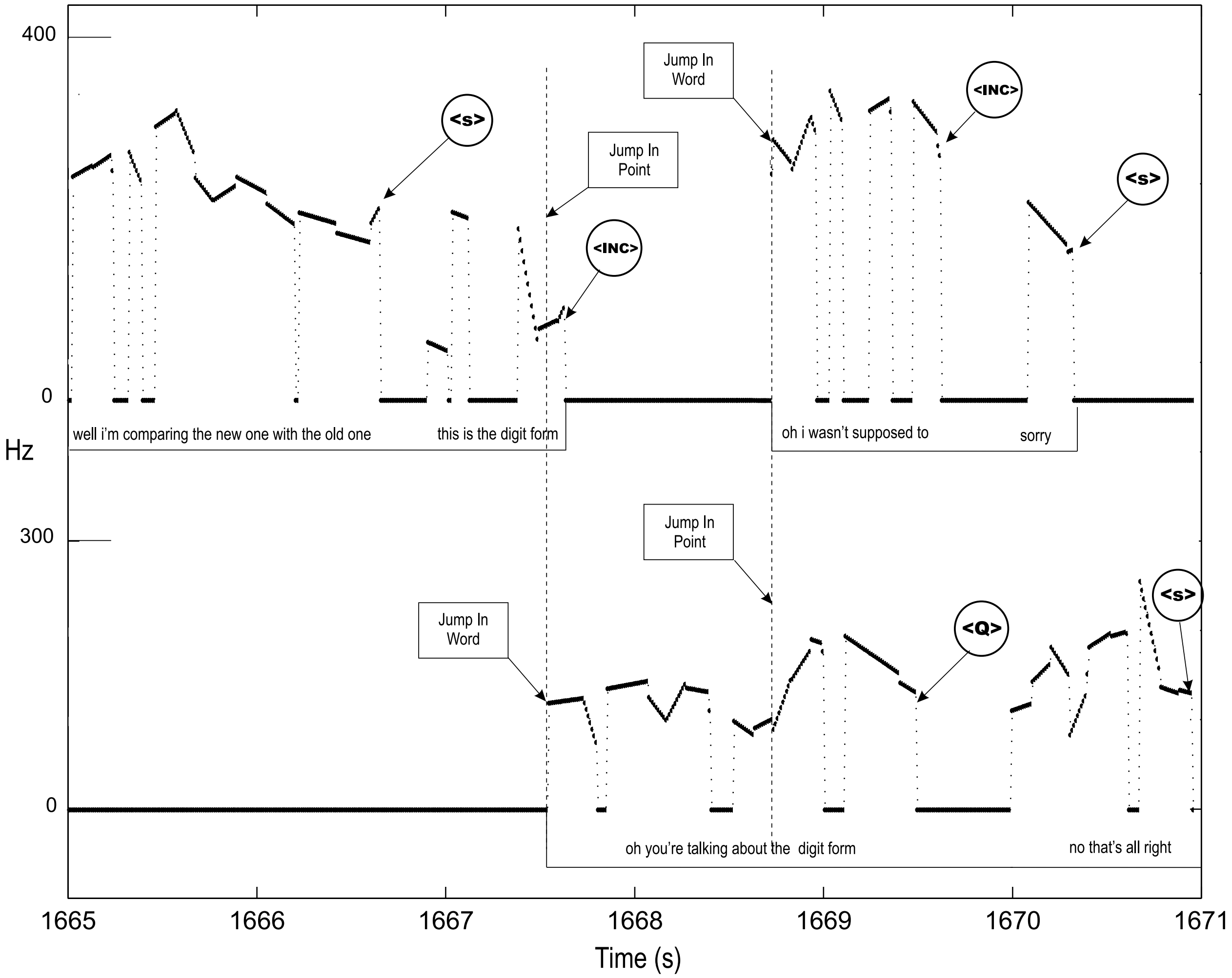

Figure 1: Excerpt from a meeting illustrating overlapping spurts, punctuation boundaries, jump-in points, jump-in words, and stylized F0 contours for two speakers (female = top, male = bottom). Spurts are delimited by vertical lines enclosing the word transcripts. (Note that due to layout limitations, words do not line up exactly with pitch contours within a spurt.) Circled tags mark punctuation events, including <s> (sentence boundary), <INC> (incomplete sentence), and <Q> questions (grouped with the complete sentence class here); no disfluency example is shown. 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 that jump-in words are always spurt-initial (see definition in text).

2.5. Language models

To provide a conservative baseline against which to evaluate the contribution of prosodic features relative to lexical information, we compare performance of the prosodic model with that of an N-gram language model (LM) based on the true words. We allow the LM to know the true words to provide a ceiling on the N-gram LM performance for this dataset. Given the high error rates for this task (between 30% and 60%) [2] we expect the LM-based classifier to deteriorate severely if it were based on recognized words. One factor that currently limits the performance of the LM is the very small amount of training data. We did not experiment with LMs trained from other sources, since much of the phenomena studied here are unique to multi-party meetings.

The LM for punctuation is a hidden-event N-gram model of the type used in our earlier work [5]. 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. We use the forward-backward algorithm for HMMs to recover the best boundary types as well as their posterior probabilities. We also combine the predictions of a decision tree with the HMM to construct a combined classifier, by converting the decision tree probabilities into additional HMM observation likelihoods [5].

2.6. Prosodic features

We automatically extracted and computed a large set of features associated with each word. Various versions of each feature were used, indexed by the location at which the feature was extracted, relative to the point of interest:

PP_boundary P_word P_boundary C_word F_boundary F_word

where P = previous, C = current, F = following. The basic prosodic feature types include pause and duration features, stylized F0 features, speaking rate features and energy features.

Pause and duration features. Pause features included raw pause durations. For phone durations we chose to start with a simple approach and model only vowel durations; this may also be more robust to ASR errors than modeling all phone durations. We used vowel durations with ratio and z-score normalizations on both context-independent and context-dependent duration statistics from the Switchboard corpus [16] (since there is more data in Switchboard; we would expect that given enough data, using statistics from the meeting training data would only help us).

Intonation features. F0 features were based on frame-level postprocessed output of the ESPS/Waves pitch tracker `get_f0` [17] with default parameter settings. Raw pitch values were median filtered and then "stylized" using a piece-wise linear algorithm (PWL) described in [18]. This approach yields straight-line approximations of the original contour, helps removes outliers, and allows for more robust extraction of local slopes and minimum and maximum F0 values. In addition we automatically computed an F0 "baseline" using a log-normal tied-mixture model of F0 [18], based on all voiced regions from a particular speaker. This proved to be a valuable normalization parameter.

To estimate local pitch range, we extracted minimum, mean, maximum, and last F0 values at each word position. We also created "windowed" versions, in which the window started at the end of the word, and moved backward in time by N frames, where $N = 10, 20, 50, 80, 100$. These local values were normalized by the log difference and log ratio to the speaker's baseline F0 value. As a more global measure of pitch range, we computed features based on the minimum, mean, and maximum stylized F0 in the segment, again normalized by the speaker's F0 baseline. For an estimate of local pitch movements, we used the last stylized slope of the word, as well as versions for each of the window lengths. We also compared the F0 values in the current word to both those in the previous and those in the following word; similar features compared stylized slope values of adjacent words. Finally, we included features capturing the distance of the current word, positive or negative, from the maximum and minimum stylized F0 in the segment.

Speaking rate features. Using the automatic alignments, we estimated local speaking rate by computing the number of vowels per unit time of speech, excluding pauses. The averages were reset at spurt boundaries. (Speaking rate features turned out not to be used by any of the classifiers learned for the tasks discussed here.)

Energy features. From the frame-level RMS energy output by `get_f0` we extracted values in two modes: all frames, and voiced-frames only. We used the minimum, mean, and maximum values for each statistic within a word. Features were either raw, or normalized by the mean or mean plus standard deviation for the speaker over the meeting.

Nonprosodic features. In addition to prosodic features we included a variety of other features (including cheating features) for the purpose of exploring how they interact with prosody in characterizing our locations of interest. These included: the current word position, punctuation features (sentence boundaries, disfluencies, and incomplete sentences), overlap features (start, during, end of overlapping speech), and lexical features denoting "special" words found in recent work to correlate with some of our locations of interest [11]. The special words included backchannels (e.g., "uh-huh"), filled pauses (e.g., "um"), coordinating conjunctions (e.g., "and"), and discourse markers (e.g., "well"). Finally, we included a set of global contextual features, to determine whether prosodic features correlated with these factors. The set included the speaker's identity, sex, and native language, and the microphone type and meeting type. As discussed in Section 3, our prosodic models were able to perform well for all tasks without use of the nonprosodic features, although some correlations are worth exploring in future work on integrated models.

2.6.1. Decision tree modeling

As in earlier work, we used CART-style decision trees [19] as classifiers that learn to predict classes and their posterior probabilities from input features. We observed that classifier performance is usually improved if a feature subset selection algorithm is used. This algorithm searches for a close to optimal subset of input features, instead of relying on the greedy feature selection strategy used by the decision tree learning algorithm.

We built and tested decision trees both on the raw distribution of cases in the data, and on downsampled data sets that equate the frequency of all classes. The downsampled training has several purposes. First, the raw class distribution is often highly skewed, and as a result decision trees focus their resources on the majority class without revealing much about the prosodic properties of the minority classes. Second, the prior distribution of classes can vary greatly over different meeting types; the downsampling can thus be considered as a form of normalization across genres. Finally, a tree with equal priors can be more directly integrated with the LM-based classifier since its posterior probability estimates are proportional to class likelihoods [5, 6].

3. Experiments and Results

We report results for four experiments. Each experiment consists of a classification task in which one or more classifier is trained to predict the posterior probability of all possible classes given the features, at each word boundary location. The classifier can be a prosodic decision tree, LM, or combination. Results are reported in terms of two metrics: *accuracy* (the percentage of cases in which the class with the highest posterior probability is correct) and *efficiency* (the reduction in class entropy achieved by the classifier relative to the prior distribution.) Formally, efficiency is defined as $\frac{H(p_0)-H(p)}{H(p_0)}$, where p_0 denotes the prior distribution, p the estimated posterior distribution, and H is the entropy. Due to the normalization relative to $H(p_0)$, efficiency allows an approximate comparison of classifiers on different tasks and with different priors, and therefore different inherent difficulties. Efficiency 1 entails a perfect classifier, whereas zero efficiency characterizes a classifier that does no better than chance. Where appropriate, we report decision tree performance both on equal priors (downsampled) and on the raw class distribution. Language models can only be evaluated on the raw distribution since contiguous data points are required by the model.

3.1. Task 1a: Predicting punctuation

This task is a three-way classification of word boundaries as either complete sentence ends, incomplete sentence/disfluent

	Equal priors		Raw priors	
Model	Acc.	Eff.	Acc.	Eff.
Chance	33.33	0	78.47	0
DT only	62.23	22.06	82.44	23.63
LM only	n/a	n/a	88.69	51.76
DT + LM	n/a	n/a	89.72	56.71

Table 2: Accuracy and efficiency results on Task 1a (automatic punctuation), using decision (DT) alone, true-word language model (LM) alone, and combining both models. Chance denotes to a baseline classifier that always chooses the class with the highest prior probability.

boundaries, and fluent sentence-internal word boundaries. To avoid small class sizes, incomplete sentence ends and other disfluencies are grouped together. This is reasonable since both represent disfluent (self-)interruption points; but note that from a forward-looking perspective incomplete sentence ends are more like complete sentence ends in that both mark the beginning of a new sentence. We chose here not to put question sentence ends into a separate category due to their low frequency. Instead, they were grouped with other sentences; this may hurt us on questions that have rising intonation.

In our decision tree experiments we disallowed lexical features, in order to encourage the use of prosodic features and create classifiers that might complement the LM. We also excluded speaker identity as a feature since it strongly correlates with the priors for the various event classes.

The decision tree induced on the equal-priors version of this task is shown in Figure 2. It relies most heavily on pause duration at the boundary following the current words (F_PAU_DUR) and on the vowel duration of the current word (C_VOWEL_DUR), with minor roles for various features encoding F0 range and energy.

The results for decision trees with both equalized and raw priors, as well as for the word-based LMs, are summarized in Table 2. Note that the language model has access to the correct word identities, and can therefore predict some events perfectly, such as disfluencies accompanied by filled pauses (“uh” and “um”).

Interestingly, the results found here on meeting speech are similar to those previously obtained on telephone conversations (the Switchboard corpus). As in [5, 6], prosodic features contribute significantly to disfluency and sentence boundary detection, both without and with a language model as the baseline classifier. Furthermore, the most important features used are pause duration and vowel duration, as previously found on Switchboard, and in contrast to Broadcast News speech, where pitch played a more prominent role [6]. That is, people in meetings use a prosodic style more similar to casual phone conversations than to broadcast speech. This result confirms a similarity between meeting speech and conversational telephone speech that we recently observed when studying speech recognition performance [11].

3.2. Task 1b: Predicting punctuation online

We were further interested in how successfully the punctuation task could be accomplished using only *preceding* speech as context. Such an *online* (or *causal*) prediction of events would be of practical interest for conversational agents that follow or participate in meetings in real time [12]. Also, the task is of interest from a psycholinguistic perspective, since it can be performed near-causally by humans.

	Equal priors		Raw priors	
Model	Acc.	Eff.	Acc.	Eff.
Chance	33.33	0	78.47	0
DT only	53.96	11.96	79.35	6.90
LM only	n/a	n/a	84.56	31.53
DT + LM	n/a	n/a	84.58	34.64

Table 3: Accuracy and efficiency results on Task 1b (automatic punctuation task using online prediction).

To investigate this question we created online versions of our classifiers. In the case of the decision trees we only allowed features that could be computed from the preceding speech. For the LM-based classifier, we estimated posteriors using only the “forward” probabilities in the underlying HMM.

Results are summarized in Table 3. We observe a substantial increase in classification error ($1 - Accuracy$) for both the decision tree and the LM classifiers. The relative error rate increase is 22% for the decision tree classifier with equal priors. The degradation can be attributed to the removal of the single most important feature, the pause duration at the location of interest. The LM-based classifier degrades even more, by 37% relative. This is also not surprising, given that most distinctive segmentation cues occur at the starts of sentences, i.e., *after* the location of interest, and are therefore not accessible to a LM that operates strictly online.

3.3. Task 2: Predicting “Jump-in Points”

We turn now to a task relevant to turn taking: predicting the locations at which a background speaker “jumps in” (starts an overlapping spurt). We would ideally want to find the locations at which speakers *could have* jumped in, but in fact we only have evidence of the locations where speakers actually *did* jump in. Therefore this task, unlike the others, has a large amount of inherent uncertainty, since we cannot distinguish locations where the background speaker wanted to interrupt but didn’t choose that location, from cases where the background speaker did not want to interrupt.

The prior for this task is low: only 2.3% of all word locations, since we are considering only the locations of interrupt starts. We therefore ran all experiments with equalized priors, to encourage the tree to look at features rather than priors. We also restricted the models (both prosodic and LM) to use only information from locations prior to the potential jump-in, since this is what the “jumper-in” has available via direct observation (although a human listener could presumably predict to some extent what the current speaker will say next).

It should be pointed out that we considered only those cases in which a background spurt started *within* a foreground speaker’s spurt, using our (arbitrary) duration for minimum spurt pauses of 0.5 seconds. This means we do not consider any jump-ins occurring *between* spurts, even if the jump-in occurred less than 0.5s after the end of the previous foreground spurt, as illustrated in Figure 3. Since we do not consider forward-looking features in this task, it is quite possible that at least for cases of spurt onsets at locations like (a) in Figure 3, in which a background speaker starts soon after the end of foreground speech, the prosodic features may be similar for FG(1) and FG(2). So, our models would be expected to do better if we

```
 0.3333 0.3333 0.3333 NONE DF/INC SENT
F_PAU_DUR < 0.335:  0.4557 0.3442 0.2001 NONE
|   C_VOWEL_DUR < 7.5:  0.597 0.2591 0.1439 NONE
|   |   F_PAU_DUR < 0.02:  0.6612 0.2218 0.117 NONE
|   |   F_PAU_DUR >= 0.02:  0.1823 0.5 0.3177 DF/INC
|   C_VOWEL_DUR >= 7.5:  0.3071 0.4338 0.2592 DF/INC
|   |   C_VOWEL_DUR < 17.5:  0.3822 0.3125 0.3053 NONE
|   |   |   F_PAU_DUR < 0.0575:  0.4476 0.2908 0.2616 NONE
|   |   |   |   C_VOWEL_DUR < 11.5:  0.5159 0.246 0.2381 NONE
|   |   |   |   C_VOWEL_DUR >= 11.5:  0.3471 0.3568 0.2961 DF/INC
|   |   |   |   |   F_F0K_LOGRATIO_WORDMAX_BASELN < 0.35252:  0.3789 0.2501 0.371 NONE
|   |   |   |   |   |   C_RMS_MAX_R < 0.54694:  0.2468 0.6667 0.08658 DF/INC
|   |   |   |   |   |   C_RMS_MAX_R >= 0.54694:  0.3863 0.2267 0.387 SENT
|   |   |   |   |   |   |   C_F0K_LOGRATIO_LASTPWLWIND100_BASELN < 0.13333:  0.2671 0.1901 0.5428 SENT
|   |   |   |   |   |   |   C_F0K_LOGRATIO_LASTPWLWIND100_BASELN >= 0.13333:  0.4291 0.2399 0.331 NONE
|   |   |   |   |   |   |   |   F_RMS_MAX_R < 0.47863:  0.1375 0.1426 0.7199 SENT
|   |   |   |   |   |   |   |   F_RMS_MAX_R >= 0.47863:  0.4491 0.2465 0.3044 NONE
|   |   |   |   |   F_F0K_LOGRATIO_WORDMAX_BASELN >= 0.35252:  0.3129 0.4715 0.2156 DF/INC
|   |   |   |   |   |   F_RMS_MAX_R < 3.1594:  0.2568 0.505 0.2381 DF/INC
|   |   |   |   |   |   F_RMS_MAX_R >= 3.1594:  0.4915 0.3646 0.1439 NONE
|   |   |   F_PAU_DUR >= 0.0575:  0.1396 0.3928 0.4676 SENT
|   |   |   |   C_F0K_LOGRATIO_WORDMIN_BASELN < 0.1216:  0.1008 0.2856 0.6136 SENT
|   |   |   |   C_F0K_LOGRATIO_WORDMIN_BASELN >= 0.1216:  0.1667 0.4677 0.3656 DF/INC
|   |   C_VOWEL_DUR >= 17.5:  0.1475 0.6913 0.1612 DF/INC
F_PAU_DUR >= 0.335:  0.04798 0.3079 0.6441 SENT
|   C_VOWEL_DUR < 26.5:  0.04884 0.2499 0.7012 SENT
|   C_VOWEL_DUR >= 26.5:  0.04235 0.6853 0.2724 DF/INC
|   |   P_PAU_DUR < 3.1035:  0.04818 0.766 0.1858 DF/INC
|   |   P_PAU_DUR >= 3.1035:  0.008097 0.2105 0.7814 SENT
```

Figure 2: Prosodic decision tree for Task 1a: three-way classification of punctuation classes. The tree is trained on equal class priors (downsampled to smallest class size). Class labels are *SENT* (complete sentence boundary), *DF/INC* (disfluencies or incomplete sentence boundary), and *NONE* (not a sentence, disfluency, or incomplete sentence boundary); order of probabilities listed is: *NONE DF/INC SENT*. Feature names are [P,C,F]_ (measure is taken at the [previous,current,following] word or boundary) plus measure. Measures are: PAU_DUR = pause duration, VOWEL_DUR = raw vowel duration, F0K_LOGRATIO_WORD[MAX,MIN]_BASELN = log ratio of [max,min] stylized F0 in word to F0 of estimated speaker baseline, F0K_LOGRATIO_LASTPWLWIND100_BASELN = log ratio of last stylized F0 value in window of 100 msec ending at end time of word, to F0 of estimated speaker baseline, RMS_MAX_R = ratio of max RMS energy in word to mean RMS over all data for that speaker.

included such datapoints than they do here.

We ran two versions of the experiment, one including punctuation and lexical features to test for any correlation, and one excluding such features. Both experiments yielded the same tree, i.e., no lexical or punctuation features were chosen by the tree.

The decision tree for the jump-in location prediction task achieved an accuracy of 64.74% (efficiency = 12.59%), well above the chance level of 50%. Figure 4 depicts the tree. A language model trained for the same task and using only past word context did not perform better than chance.

These results are remarkable for several reasons. First, even this modest accuracy is noteworthy given the inherent uncertainty of the task. Second, none of the (cheating) word features or punctuation features available are used; the tree uses only prosodic features. The fact that an N-gram LM is not effective for this task suggests the irrelevance of word identities. Third, adding forward-looking features to the tree yields almost no accuracy improvement, so predicting where another speaker will jump in is—from a prosodic standpoint—based on *past* prosody. Fourth, the features used in this tree suggest that background speakers jump in at those locations in the foreground speaker's talk that *look similar to sentence boundaries, but which are not actually sentence boundaries*. We know that they are not better predicted by actual sentence boundaries, because if that were the case, the punctuation feature would have been queried. As shown in Figure 4 the preferred jump-in points

```
          0    .5   1.0 sec
<-------->         <--------->FG1
<--------:   :--------------->FG2
           (a)   (b)
```

	`----`	=	speech
	`< >`	=	begin, end of spurt
Legend:	`:`	=	begin, end of pause within spurt
	`FG`	=	ForeGround talker
	`a,b`	=	spurt start by background talker

Figure 3: Illustration of dependence of jump-in point definitions on spurt pause threshold. Spurt boundaries are defined by the pause threshold of .5 seconds in the foreground (FG) talker's speech. FG1 and FG2 illustrate two different foreground speaker patterns, which are identical up to the point marked "0" seconds, but that differ in spurt definition due to the (future-determined) total pause duration. FG1 (but not FG2) pauses long enough to create a spurt boundary at point "0". If some background talker starts a spurt at either time "a" or point "b", the spurt onset will be considered a jump-in relative to FG2, but not to FG1—even though in the case of a spurt at time "a" the speech/nonspeech history of FG1 and FG2 are identical up to that point.

```
0.5 0.5  0(no jump-in)  1(jump-in)
P_PAU_DUR < 0.105:  0.5226 0.4774 0
|   P_RMS_MIN_R < 0.25621:  0.4686 0.5314 1
|   |   PP_PAU_DUR < 1.4635:  0.4882 0.5118 1
|   |   |   CP_FOK_DIFF_MAXPWLWORD_MAXPWL_P-WORD < -16.83:  0.4222 0.5778 1
|   |   |   CP_FOK_DIFF_MAXPWLWORD_MAXPWL_P-WORD >= -16.83:  0.5113 0.4887 0
|   |   |   |   CP_FOK_DIFF_MAXPWLWORD_MAXPWL_P-WORD < 8.2:  0.5424 0.4576 0
|   |   |   |   |   P_TRIVOWEL_DUR_Z < 1.7:  0.556 0.444 0
|   |   |   |   |   |   CP_FOK_DIFF_MAXPWLWORD_MAXPWL_P-WORD < -8.755:  0.4627 0.5373 1
|   |   |   |   |   |   CP_FOK_DIFF_MAXPWLWORD_MAXPWL_P-WORD >= -8.755:  0.5844 0.4156 0
|   |   |   |   |   P_TRIVOWEL_DUR_Z >= 1.7:  0.3096 0.6904 1
|   |   |   |   CP_FOK_DIFF_MAXPWLWORD_MAXPWL_P-WORD >= 8.2:  0.4373 0.5627 1
|   |   PP_PAU_DUR >= 1.4635:  0.3034 0.6966 1
|   P_RMS_MIN_R >= 0.25621:  0.6526 0.3474 0
P_PAU_DUR >= 0.105:  0.2437 0.7563 1
```

Figure 4: Prosodic decision tree for Task 2: two-way classification of "jump-in" classes, trained on equal class priors (downsampled to smallest class size). Class labels are *0* (no jump-in), and *1* (jump-in); probabilities are listed in that order. Feature names are [PP,P,C,CP,F]_ (measure is taken at the [2-previous,previous,current,current-to-previous,following] word or boundary) plus measure. PAU_DUR = pause duration, RMS_MIN_R = ratio of min RMS energy in word to mean RMS over all data for that speaker, FOK_DIFF_MAXPWLWORD_MAXPWL_P-WORD = difference between max stylized F0 in current word and that in previous word, TRIVOWEL_DUR = vowel duration z-score-normalized by triphone duration.

are locations with pause boundaries, low RMS, a recent previous fall in F0 from the previous word to the current word, and extended preceding triphone-normalized vowel durations. These results suggest, consistent with [10], that speakers use prosody, not just words, to coordinate turn-taking in conversation.

3.4. Task 3: Predicting "Jump-in Words"

The final question we investigated was whether there is any difference in the prosody of spurt onsets that start during silence, versus during speech by another talker. In other words, do speakers modify their prosody when they have to compete with an already active speaker? As shown in Figure 1, the jump-in words for one speaker correspond to the jump-in points in another speaker's spurt.

This task differs from the previous two tasks in that it considers only a subset of word datapoints, rather than all word locations, since it looks only at words corresponding to the start of a speaker's spurt. For this reason, a language model for this task would be most naturally formulated not in terms of hidden event N-grams, but rather over the words making up the spurts themselves. We have not yet investigated LM-based classifiers for this task, and report results using prosodic decision trees only. Again, to remove the skew in the raw distribution we trained and tested the classifiers with equal priors.

The decision tree trained with all features achieved an accuracy of 77.28%, far above the 50% chance level, as shown in Table 4. For this task, unlike previous tasks, we noticed that RMS was used heavily by the trees. This could indicate that speakers raise their voices when interrupting others, or simply that the interrupted speakers' signal is picked up by the interrupting speaker's microphone. In the latter case we would have to consider RMS a "cheating" feature since it indirectly measures what we want the classifier to predict. Further investigation is required to assess to what degree the raised RMS reflects a contribution from background speech. However, we found that the classifier still does quite well if we remove all RMS features from the candidate feature list, as shown in Table 4. It does so in part by making increased use of F0 features (F0 is raised when starting during current speech, relative to starting during silence). Thus overall, this task provides evidence that speakers adjust for the background level when starting up, by "raising" their voice.

Table 4: Results for Task 3: detection of spurt starts in overlapped speech. Both accuracy (Acc.) and efficiency (Eff.) are listed.

Model	Acc.	Eff.
Chance	50.00	0
With RMS	77.28	24.90
No RMS	72.58	15.31

4. Summary

We have studied the use of prosodic features for three classification tasks on meeting speech. In Task 1, automatic punctuation, we found that prosodic features, modeled by decision trees, can provide a significant boost to classification accuracy (10% relative error reduction) over that of a language model based on the true words. The main features used in the prosodic classifier were pause and phone durations, consistent with what we had found in conversational telephone speech [6]. We also investigated how the LM and prosodic classifiers deteriorate when only information from the past of a given location is used, simulating online processing. Here the accuracy is considerably worse (22-37% error increase), but the prosodic classifier degrades considerably less than does the LM.

The remaining two tasks involved the characterization of speech at locations of speaker overlap, a pervasive phenomenon in casual conversation. First, we trained classifiers to predict the location of background speaker jumps-ins, based on the foreground speaker's prosody and words (Task 2). Although earlier work [11] showed that jump-in locations are correlated with certain word classes (such as discourse markers and disfluencies), a simple word-based N-gram model was not effective for this task. Prosodic cues, however, were able to predict jump-in locations at better than chance, despite the inherent uncertainty for this task. Analysis of the features used showed that jumping-in is predicted at locations that look prosodically like sentence boundaries—even if they are not actual sentence boundaries.

The other overlap-related task (Task 3) involved classifying spurt onset words into overlap and nonoverlap cases. Only prosodic features were investigated for this task. The induced classifiers showed that jumping-in is correlated with higher RMS energy as well as higher F0, suggesting that speakers adjust for the background level when starting to speak, by "raising" their voice.

Finally, across the different tasks, we observed different features to be be important. Sentence boundary and disfluency detection used mostly temporal features (pause and phone duration), while prediction of jump-ins used both durations and F0. Energy and F0 was important for identifying jump-in words. Also we found that generally, unnormalized durations seemed to work better than normalized ones. The most useful F0 features were those that measured F0 distance from the local word or window, to an estimated speaker baseline—rather than local pitch movements. F0 extremes (minima and maxima) rather than means, seem to be the most useful metrics for computing the pitch range features.

In conclusion, although further research as well as more data are needed in studying these and related areas, we find evidence that prosody provides useful cues for a variety of tasks relevant to the automatic processing of meeting data. While the focus in the present work was on prosody, the results suggest the need for an integrated model combining lexical and prosodic aspects of speech communication in natural, multi-party contexts.

5. Acknowledgments

We thank our ICSI collaborators: Nelson Morgan, Chuck Wooters, Jane Edwards, Dan Ellis, Dave Gelbart, Adam Janin, Thilo Pfau, Sonali Bhagat, Kai Filion, and Rajdip Dhillon; and SRI collaborator Kemal Sönmez. This work was funded by an ICSI DARPA Communicator project (via U. Washington), supplemented by an award from IBM. Additional support came from DARPA-TRVS, NASA, and NSF-STIMULATE (IRI-9619921) projects at SRI.

6. References

[1] 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 Kaufmann.

[2] 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.

[3] M. Mast, R. Kompe, S. Harbeck, A. Kießling, H. Niemann, E. Nöth, E. G. Schukat-Talamazzini, and V. Warnke. Dialog act classification with the help of prosody. In H. T. Bunnell and W. Idsardi, editors, *Proc. ICSLP*, vol. 3, pp. 1732–1735, Philadelphia, 1996.

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

[5] A. Stolcke, E. Shriberg, R. Bates, M. Ostendorf, D. Hakkani, M. Plauché, G. Tür, and Y. Lu. Automatic detection of sentence boundaries and disfluencies based on recognized words. In R. H. Mannell and J. Robert-Ribes, editors, *Proc. ICSLP*, vol. 5, pp. 2247–2250, Sydney, 1998. Australian Speech Science and Technology Association.

[6] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür. 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.

[7] G. Jefferson. A case of precision timing in ordinary conversation: overlapped tag-positioned address terms in closing sequences. *Semiotica*, 9:47–96, 1973.

[8] S. C. Levinson. *Pragmatics*. Cambridge University Press, Cambridge, 1983.

[9] E. Ochs, E. A. Schegloff, and S. A. Thompson. *Interaction and Grammar*. Cambridge University Press, Cambridge, 1996.

[10] E. Couper-Kuhlen and M. Selting, editors. *Prosody in Conversation*. Cambridge University Press, Cambridge, 1996.

[11] 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.

[12] 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.

[13] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. Rao Gadde, M. Plauché, C. Richey, E. Shriberg, K. Sönmez, 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.

[14] E. Shriberg, R. Bates, and A. Stolcke. A prosody-only decision-tree model for disfluency detection. In G. Kokkinakis, N. Fakotakis, and E. Dermatas, editors, *Proc. EUROSPEECH*, vol. 5, pp. 2383–2386, Rhodes, Greece, 1997.

[15] J. Hirschberg and C. Nakatani. Acoustic indicators of topic segmentation. In R. H. Mannell and J. Robert-Ribes, editors, *Proc. ICSLP*, pp. 976–979, Sydney, 1998. Australian Speech Science and Technology Association.

[16] J. J. Godfrey, E. C. Holliman, and J. McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *Proc. ICASSP*, vol. 1, pp. 517–520, San Francisco, 1992.

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

[18] K. Sönmez, 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.

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