A PROSODY-BASED APPROACH TO END-OF-UTTERANCE DETECTION THAT DOES NOT REQUIRE SPEECH RECOGNITION

Luciana Ferrer *Elizabeth Shriberg* *Andreas Stolcke*

Speech Technology and Research Laboratory
SRI International, Menlo Park, CA, USA
http://www.speech.sri.com/

ABSTRACT

In previous work we showed that state-of-the-art end-of-utterance detection (as used, for example, in dialog systems) can be improved significantly by making use of prosodic and/or language models that predict utterance endpoints, based on word and alignment output from a speech recognizer. However, using a recognizer in endpointing might not be practical in certain applications. In this paper we demonstrate that the improvements due to the prosodic knowledge can be realized largely without alignment information, i.e., without requiring a speech recognizer. A prosodic end-of-utterance detector using only speech/nonspeech detection output is still considerably more accurate and has lower latency than a baseline system based on pause-length thresholding.

1. INTRODUCTION

Every human-machine dialog system must be able to detect when a user has finished speaking and is waiting for an answer from the system. This task is referred to as "end-of-utterance (EOU) detection". Current systems rely only on a "pause threshold" for making this decision [1]. A related task is that of detecting the pauses, or nonspeech regions. In this work, as in our previous work in [2], we assume that good detection of nonspeech regions is given (by some state-of-the-art method), and focus our attention solely on the EOU detection algorithm itself.

As we demonstrated in [2], current systems are suboptimal and can be significantly improved by the use of prosodic and language model cues. In that work, we use a state-of-the-art recognition system to obtain the speech and nonspeech regions and the word hypotheses with their corresponding phone alignments. Using the alignments and the acoustic signal we compute a set of prosodic features, and model them by using decision trees. The trees yield a posterior probability estimate for the presence of an EOU for each location that combines with the probability given by a language model to produce a score. That score is compared with a chosen threshold to make the final decision on whether or not that location is an EOU. We compare the performance of our system with that of a "baseline" system that uses only pause duration for its decision.

In this paper we introduce further improvements to that system, and focus on prosodic EOU detection in a scenario where speech recognition output is not available at the endpointing stage, as might be the case in real-time or resource-constrained applications. As we will show, ASR-independent prosodic EOU detection is still much better than the baseline system, although it has some degradation with respect to the ASR-dependent system which uses the alignment and word output from a speech recognizer.

2. SYSTEM DESCRIPTION

After each pause in the speech input, there is always a possible EOU. Our approach is to first detect pauses by using either the alignments output by the recognizer or a pause detector and then to use prosody and grammar information (when available) to obtain a score that measures the probability of that pause being an EOU. The final decision is obtained by comparing that score with a chosen threshold.

As a baseline against which to evaluate our system we use the method employed by current speech dialog systems, which is based only on the duration of the pause in the current boundary. If that pause duration is bigger than a certain threshold (typically in the range of 0.5 to 1 second) the system decides that the EOU has been reached.

Our proposed system (Figure 2), in contrast, makes an EOU decision at various points after a pause has been detected. It does so by computing a set of prosodic features whose extraction is based mainly on the acoustic signal. If a recognizer is used at the first step, more features can be computed using the word and phone alignments. These features become an input to a decision tree classifier that estimates the posterior probability that the speaker is done with the utterance at that location. As the pause gets longer, the system continuously queries new classifiers (the need for different classifiers depending on the current pause length will be explained in section 2.4). In practice, we limit these queries to a finite set of waiting times, called *decision points* (DPs), to reduce computation. When the resulting posterior probability at any DP exceeds a threshold or a maximum pause duration is reached, the system outputs the decision that an EOU was found. Figure 1 shows the location of the DPs and the pause threshold in relation to the pause.

If the recognizer output is available, the stream of hypothesized words can also be used to obtain a language model posterior probability to combine with the prosodic model probability. The resulting combined score is then compared with the threshold to obtain the decision.

2.1. Prosodic model decision trees

As in prior work on disfluency and sentence boundary detection [3], we trained CART-style decision trees to predict EOUs from

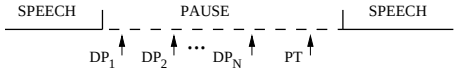

Fig. 1. Decision points (DP_i) and pause threshold (PT).

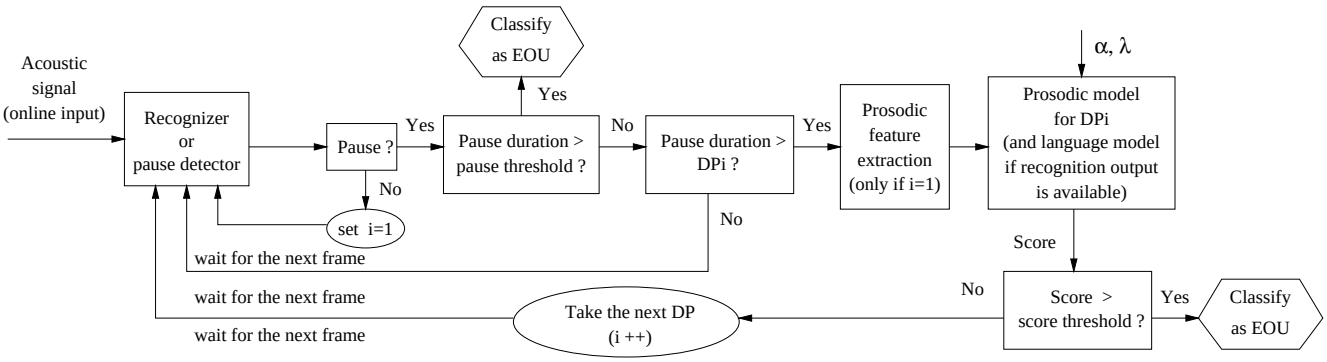

Fig. 2. Diagram of the proposed system. Each time the recognizer or the pause detector finds a pause longer than the first decision point (DP_1), the prosodic features are computed and a score is obtained using a smoothed version of the decision tree posterior for that DP_i and, if recognition output is available, the LM posterior. If that score is lower than the score threshold, the system waits for a new DP to be reached. The same features are then used to compute a new score, using the tree for the new DP. The process repeats until the score for a DP exceeds the score threshold, the pause duration is greater than the pause threshold, or the pause ends (i.e., the speaker resumes speaking).

automatically extracted prosodic characteristics around the point of interest. In this case, because we are interested in *online* detection, we use only those features that can be extracted from the signal *before* the current DP time is reached.

Two main types of prosodic features can be computed based on duration and pitch (fundamental frequency, or F0). As seen in Figure 2, these features are computed only for points at which the recognizer or the pause detector finds a pause (i.e., only at interword pauses at least 30 ms long, the minimum duration of our recognizer's pause model).

Most of the duration features are extracted from the time alignments that the recognizer outputs, for example, the duration of the last rhyme (i.e., the time from the last vowel in the word to the end of the word) in the word before the pause. This features can be normalized in different ways by using the phone duration statistics, speaker-specific duration statistics, and so on. Other duration features can be computed without the need of time alignments, for example, the time from the start of the last voiced region to the current pause.

To obtain F0 features, pitch tracks are extracted from the signal and then post-processed using an improved version of the approach in [4]. Pitch contours are "stylized," octave errors are estimated, and, most important, a set of speaker-specific pitch range parameters is computed. These parameters include a value that allows us to estimate a speaker's "floor" or lowest typical F0 value. The pitch features are computed using the stylized pitch and those parameters. For some of these features, the time alignments obtained by the recognizer are also used for the computation; for the rest, a fixed time window is used instead. An example of a pitch feature is the distance from the average pitch in the last word before the boundary to the speaker's floor pitch.

Finally, a tree is trained for each DP, using all boundaries with a pause duration longer than that DP. Thus the first tree includes every sample with pause duration beyond the first DP; the second tree uses a subset of the previous samples, and so on. As a result, the classifiers for later DPs are trained using less data than those for earlier DPs, and are expected to be less robust. Therefore, we use the following recursive linear interpolation rule to smooth the later classifiers:

$$\begin{aligned} i = 1: &\quad P_{PM(DP_i)} = P_{DT(DP_i)} \\ i > 1: &\quad P_{PM(DP_i)} = \lambda P_{DT(DP_i)} + (1-\lambda) P_{PM(DP_{i-1})} \end{aligned} \tag{1}$$

where $P_{DT(DP_i)}$ is the probability given by the tree for decision point DP_i, λ is the weight of the current tree relative to all the previous trees, and $P_{PM(DP_i)}$ is the resulting prosodic model score for decision point DP_i.

2.2. Language model

When an automatic speech recognition (ASR) system is used for pause detection, the hypothesized words are available to the system and a language model (LM) can be used to estimate a posterior probability of EOU given those words. For this work, an EOU N-gram LM is trained using transcripts, where the ends of sentences are marked with a special tag. In this way, the <end_of_sentence> tag will be learned as more likely after certain sequences of words than after others. For each pause found, the probability of EOU is obtained as the conditional probability of <end_of_sentence> given the previous $N-1$ words.

2.3. Knowledge source combination

To make use of both prosodic and lexical information, we compute a simple log-linear interpolation of the LM and the prosodic model posterior probability:

$$Sc(DP_i) = \frac{P^{\alpha}_{PM(DP_i)} P_{LM}}{P^{\alpha}_{PM(DP_i)} P_{LM} + (1 - P_{PM(DP_i)})^{\alpha} (1 - P_{LM})} \tag{2}$$

where P_{LM} is the a posteriori probability for EOU given by the LM, $P_{PM(DP_i)}$ is the smoothed probability given by the prosodic model, and α is a combination weight that is empirically optimized using held-out data. The score $Sc(DP_i)$ is then compared with the score threshold for each applicable DP_i as shown in Figure 2.

2.4. Rationale for the Method

There are two motivations for using different trees for each decision point. First, the prior probability and, hence, the posterior probability for an EOU are highly dependent on the pause duration. Second, some features can be better cues at some duration lengths than at others. The duration of the last rhyme in the word before the pause is an example of this kind of feature. Before a pause, the last rhyme in a word is lengthened. Hesitation boundaries followed by pauses show even *more* lengthening than do EOU boundaries, making the duration of the last rhyme in the word a useful feature for discriminating the two cases. Even more,

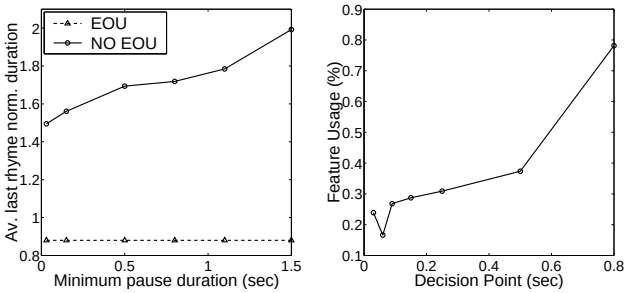

Fig. 3. Left: Average duration of the last rhyme in the word (normalized by the phone statistics and by the number of phones in the rhyme) as a function of the minimum pause duration after the word. Right: tree usage of that feature as a function of the decision point.

this lengthening is related to the length of the pause, as shown in Figure 3 (left) where all the samples with a minimum pause duration value are used to compute the average normalized duration of the last rhyme in the word.

Training a new tree for each decision point allows each tree to use the best set of features to do the classification for the corresponding minimum pause duration. As Figure 3 (right) shows, the trees choose to use the normalized last rhyme duration feature more for later DPs than for earlier DPs, which indicates that at longer pause durations that feature is a stronger cue to EOU detection than at shorter pause durations.

3. EXPERIMENTS

3.1. Methodology

As in our previous work, we tested our approach on the ATIS (Air Travel Information System) corpus [5]. Very few utterances in the database contain more than one sentence (fewer than 1%). We eliminated those utterances from our experiments and also those that contained unfinished sentences, keeping only the utterances having exactly one complete sentence. We split the corpus in three subsets: training set (12,486 utterances), development set (3356 utterances), and test set (1976 utterances).

The experiments used the SRI Decipher^{TM} recognition engine with acoustic models trained for the December 1994 ATIS evaluation [6]. The recognizer LM was trained using the transcription of the training and development sets, representing a total of 164,000 words (we can include the development set here because we use the recognizer only on the test set, performing force alignments on the other two sets). The EOU LM was trained using only the transcriptions for the training set (126,000 words). Both models made use of hand-defined, task-specific word classes for airline names, cities, and so on, to improve generalization.

We generated forced alignments for the training and development sets, and derived an extensive set of prosodic features from the resulting phone-level time alignments and the acoustic signal. Two different systems were evaluated: one using ASR output and one without. For the ASR-independent system we used only those prosodic features that can be computed without requiring time alignments. In practice, this subset of features could be computed using a pause (nonspeech) detector.

Decision trees were trained for each system by using the corresponding set of features. We used the training set for tree induction and the development set for choosing the best set of features for each DP, via an automatic feature subset selection wrapper [3]. The development set was also used to optimize the weights α and λ, and to find the optimal pause threshold.

For expediency, in testing we ran recognition on the full utterance waveforms, and then used the 1-best recognition output up to each decision point as the input to our EOU detector. Prosodic features were based on recognition outputs rather than forced alignments, but were otherwise computed as in training. To test our systems, we set decision points at 30, 60, 90, 150, 250, 500, and 800 ms into a pause.

The overall accuracy of the recognizer on the test set was a word error rate of 5.9%. This represents an idealization, as in realistic applications the decision would have to be based on the partial recognition output at the DP, that is, without the benefit of search over complete sentence hypotheses. We would therefore expect the actual recognition accuracy to be somewhat worse, affecting the performance of the ASR-dependent system, but not that of the ASR-independent system.

Our proposed EOU detection system was compared to a baseline system. The baseline system classifies a boundary as an EOU whenever the pause duration for that boundary is greater than a given pause threshold.

3.2. Results

For the purpose of the experiment we assume that speakers wait indefinitely for the system to respond, i.e., the pause duration at an EOU is infinite. This means that any EOU detector with finite pause threshold will have 100% recall rate[1]. Consequently, we evaluate how well systems trade off false detection of EOUs against the *time* it takes to detect them. Specifically, the two performance measures of interest are the *false alarm rate* (FAR), which is the percentage of non-EOU boundaries classified as EOU, and the *speaker waiting time* (SWT), which is the time between the last frame of speech and the frame at which the EOU is detected (this is the time the speaker would have to wait to obtain an answer from the system if the processing time for the answer itself were zero). The two measures are inversely related: to reduce the SWT we can accept a bigger FAR and vice versa.

Figure 4 shows FAR versus SWT for the following four systems: baseline, prosody-only with the complete set of features (including those dependent on ASR), LM only, and combined system. The prosody-only system uses the smoothed probabilities in Equation 1 as the score, while the combined system uses the combined score from Equation 2. The LM-only system is implemented by replacing $P_{PM(DP_i)}$ in Equation 2 with the *prior* probability of EOU for that DP. This approach represents an improvement over our previous system [2], which used the LM posteriors alone.

For the baseline, the curve is obtained by varying the pause threshold, while for the other systems the curve is obtained by varying the score threshold and keeping the pause threshold at 1.8 seconds, which is the best value for the range of FAR between 2% and 13% for the combined system. Larger pause thresholds could be chosen if the FARs of interest were below 2%, but this would

[1] In the work presented in [2] we erroneously considered the pause length at the end of each utterance as the time the speaker would wait for the system to answer. This new assumption of an "infinite" pause is not completely realistic either (few users would wait 30 seconds for a system to answer), but as all of the presented systems always respond before 2 seconds in the usable range of false alarm rates we consider it a reasonable assumption.

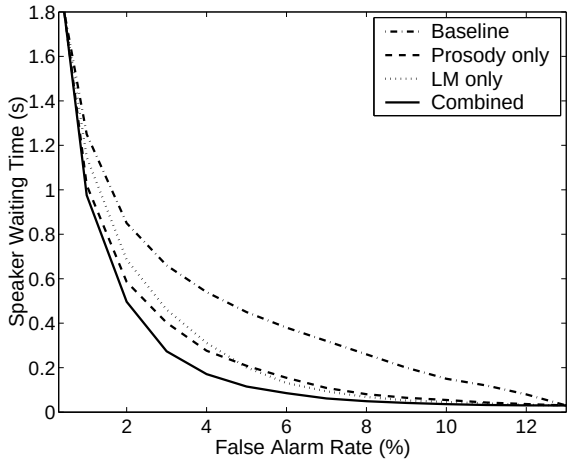

Fig. 4. SWT vs. FAR for the four systems: baseline, prosody-only with ASR, LM-only, and combined system.

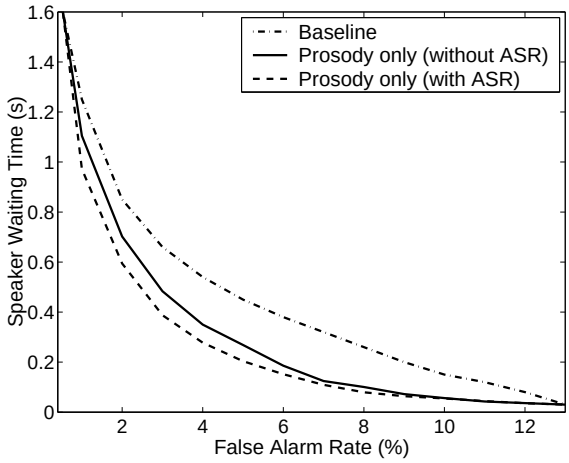

Fig. 5. Comparison of the SWT vs. FAR graphs for two prosody-only systems, one that uses the ASR output and one that does not.

hurt the performance for higher FARs. The values of α and λ were optimized for the combined system for that pause threshold using the development set, obtaining $\alpha = 2.3$ and $\lambda = 0.6$. The optimized performance measure was the average SWT over the range of FAR shown in Figure 4.

For the proposed systems every hesitation boundary with a pause longer than 1.8 seconds is inevitably recognized as an EOU, regardless of the scores obtained at the DPs. As a consequence of this, these systems have a minimum FAR (left edge in the graph) that is obtained when the score threshold is set to 1 (its maximum value) and which is given by the number of hesitations with pause duration longer than the pause threshold. At this point the four systems coincide, although this is not the last point for the baseline system which reaches 0% FAR when the pause threshold is bigger than the longest hesitation. The curves also meet at the right edge of the graph, reflecting the case where the score threshold for the proposed systems and the pause threshold for the baseline system tend toward zero, classifying all pauses as EOUs. (The minimum pause threshold for the baseline system is 30 ms, the minimum detectable pause length).

Clearly, at all shown operating points, all proposed systems consistently outperform the baseline system.

3.3. Results for an ASR-independent system

The following results were obtained using only the subset of the prosodic features that can be extracted without the use of the time alignments given by the ASR system. In this case, we assume that an ASR system is not available, which also precludes using an EOU LM, or a combined lexical/prosodic classifier.

Figure 5 shows FAR versus SWT for the baseline system, the ASR-independent system, and the ASR-dependent prosody-only system again, for comparison. The optimized pause threshold for both prosody-only systems was 1.6 seconds. From the figure we see that prosody-only performance using the restricted set of features degrades somewhat, but is still considerably better than the baseline.

4. CONCLUSIONS

We have shown two new approaches for the online detection of ends of utterances. The first one uses ASR output to obtain a prosodic model and a language model that are then combined to improve system performance. The second one, a restricted version of the first, does not require ASR. Only the prosodic model is used, relying only on features that can be extracted without word or phone alignments.

We show that the speaker waiting time is substantially shortened by the system that uses the ASR output, with reductions at a given false alarm rate as high as 81% compared to the baseline system that uses only pause duration information. We also show that the ASR-independent system, although it degrades slightly with respect to the ASR-dependent prosody-only system still gives reductions as high as 64% with respect to the baseline system.

5. ACKNOWLEDGMENTS

This work was funded by NASA under NCC 2-1256 and by NSF STIMULATE Grant IRI-9619921. The views herein are those of the authors and do not reflect the policies of the funding agencies.

6. REFERENCES

[1] R. Hariharan, J. Häkkinen, and K. Laurila, "Robust end-of-utterance detection for real-time speech recognition applications", *in Proc. ICASSP*, Salt Lake City, May 2001.

[2] L. Ferrer, E. Shriberg, and A. Stolcke, "Is the speaker done yet? Faster and more accurate end-of-utterance detection using prosody in human-computer dialog", in J. H. L. Hansen and B. Pellom, editors, *Proc. ICSLP*, vol. 3, pp. 2061–2064, Denver, Sep. 2002.

[3] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür, "Prosody-based automatic segmentation of speech into sentences and topics", *Speech Communication*, vol. 32, pp. 127–154, Sep. 2000, Special Issue on Accessing Information in Spoken Audio.

[4] 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, Dec. 1998. Australian Speech Science and Technology Association.

[5] MADCOW, "Multi-site data collection for a spoken language corpus", *in Proc. DARPA SNP Workshop*, pp. 7–14, Harriman, NY, Feb. 1992. Defense Advanced Research Projects Agency, Information Science and Technology Office.

[6] M. Cohen, Z. Rivlin, and H. Bratt, "Speech recognition in the ATIS domain using multiple knowledge sources", *in Proceedings ARPA Spoken Language Systems Technology Workshop*, pp. 257–260, Austin, TX, Jan. 1995.