IS THE SPEAKER DONE YET?
FASTER AND MORE ACCURATE END-OF-UTTERANCE DETECTION USING PROSODY

Luciana Ferrer *Elizabeth Shriberg* *Andreas Stolcke*

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

ABSTRACT

We examine the problem of end-of-utterance (EOU) detection for real-time speech recognition, particularly in the context of a human-computer dialog system. Current EOU detection algorithms use only a simple pause threshold for making this decision, leading to two problems. First, especially as speech-driven interfaces become more natural, users often pause inside utterances, resulting in a premature cut off by the system. Second, when users really *are* done, the minimum system wait is always the threshold value, needlessly adding time to the interaction. We have developed a new approach to EOU detection that uses prosodic features to address both of these problems. Prosodic features are modeled by decision trees and combined with an event N-gram language model to obtain a score that measures the likelihood that any nonspeech region is an EOU. We find that this approach dramatically improves both the accuracy and speed of online EOU detection.

1. INTRODUCTION

A 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 typically referred to as "end-of-utterance (EOU) detection". Current systems, such as Nuance [1] and SpeechWorks [2], as well as the VoiceXML standard [3], rely solely on a nonspeech (or "pause") duration threshold for making this decision [4]. A related task is that of detecting the pauses, or nonspeech regions. In clean enviroments this can be achieved by a speech recognizer alone; the resulting alignments will indicate the presence and extent of nonspeech regions. In noisy enviroments, an "endpointing" method is usually applied before recognition, to separate speech segments from background noise. The nonspeech regions are detected using spectral full-band energy values, zero-crossing detectors, entropy, pitch, and so on [5, 6, 7, 8]. The recognizer is then run only on the regions containing speech. In this work, 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.

Current EOU approaches are suboptimal, for two reasons. First, because speakers *think* while they talk, they often pause *inside* utterances; such pauses are typically at high-entropy locations, or before important content information (since the person pauses in order to decide on that content). Human listeners can discriminate hesitation pauses from final pauses using prosodic and gestural cues, but current EOU algorithms cannot, since only pause length is used. This results in systems cutting the speaker off prematurely, often annoying the user, adding time to the interaction due to retries, and even causing system understanding errors due to the processing of an utterance fragment. This problem becomes increasingly important as system developers strive to deal with more natural speaking styles. The second problem occurs when users really have finished speaking. The single pause duration threshold requires that the system wait out this duration before making a decision, adding time to the interaction at each EOU location.

The goal of the work described here is to improve both the accuracy and speed of EOU detection by making use of information beyond pauses, and to evaluate the contribution of the prosodic information alone and in combination with language model (LM) information.

2. SYSTEM DESCRIPTION

After each pause in the speech input, there is always a possible EOU. Our approach is to first detect pauses using the phone-level alignments output by the recognizer, and then to use prosody and grammar information 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, namely, thresholding of pause duration. From what we have found in an informal survey of dialog systems in English, the pause threshold is typically in the range of 0.5 to 1 second. The baseline EOU detector thus always waits for the duration of this threshold before deciding that the end of an utterance has been reached.

Our proposed system, in contrast, makes an EOU decision at every frame after a pause of any length has been detected. It does so by computing a set of prosodic features which depend only on the acoustic signal and recognition output preceding the pause in question. These features are input to a decision tree classifier that estimates the posterior probability that the speaker is done with their utterance at that location (i.e., has reached an EOU). Conceptually, the system continuously applies new classifiers as the pause gets longer; this is necessary because the prior probability, and hence also the posterior, for an EOU is highly dependent on the pause duration. In practice we limit these queries to a finite set of waiting times, called *decision points* (DPs), to reduce computation. When the decision tree posterior exceeds a threshold or a maximum pause duration is reached, the system outputs the decision that an EOU was found. A refinement of this algorithm combines the prosodic score with a language model score, as described below. Figure 1 shows a flowchart of the algorithm.

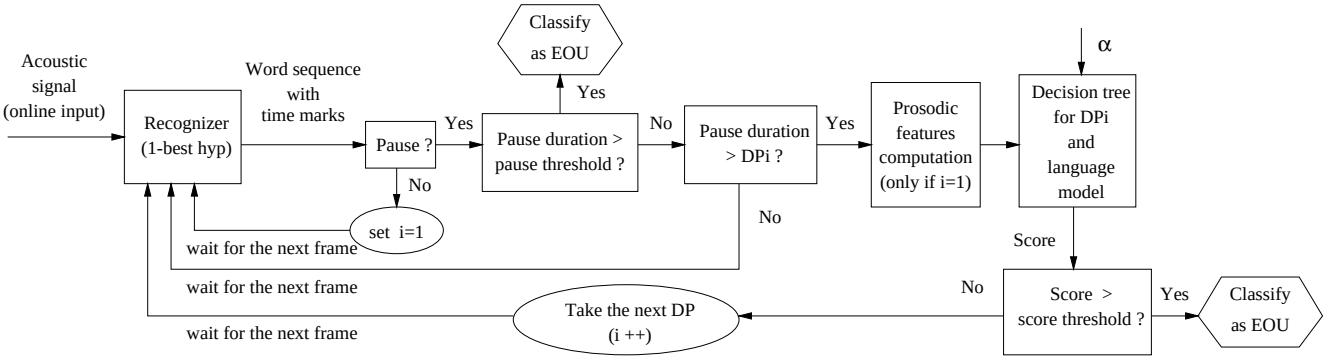

Fig. 1. Diagram of the proposed system. Each time the recognizer finds a pause longer than the first decision point (DP), the prosodic features are computed and a score is obtained using the decision tree for that DP and the LM. If that score is lower than the score threshold, the system waits for a 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 duration threshold, or the pause ends (i.e., the speaker resumes speaking).

2.1. Prosodic model decision trees

As in prior work on disfluency and sentence boundary detection [9], we trained CART-style decision trees [10] to predict EOUs from 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 are computed based on duration and pitch (fundamental frequency). As seen in Figure 1, these features are computed only for points at which the recognizer finds a pause (that is, only at word boundaries with an interword pause of at least 30 ms, the minimum duration of the recognizer's pause model). Note that although pause durations are technically prosodic features, we do not use the pause duration as a prosodic feature in our model, since our decision trees for the remaining prosodic features are already conditioned upon that feature.

Duration features are computed from the time marks output by the recognizer. Pitch features are extracted directly from the signal and then post-processed using an approach developed at SRI [11] and recently significantly improved. Pitch contours are "stylized," octave errors are estimated, and, importantly, 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. Using these prosodic features, a tree is trained for each chosen DP, using only the sets of features corresponding to boundaries with a pause duration larger than that DP. In this way, we do not split our data, since 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.

2.2. Language model

As we will see in Section 3, a significant improvement is achieved by the use of prosody alone to aid EOU detection. Prosody by itself does not consider the word identity, yet word identity is clearly important too. Some words are unlikely to occur as the last word of a sentence, while others are more likely. In addition, entire word phrases can serve similarly as cues. This suggests using an N-gram language model to improve the predictions made by the decision trees.

This 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 N-gram probability of <end_of_sentence> given the previous N-1 words. Note that, if it were not for the requirement of online processing, we could also use the words following the pause to improve the prediction of EOU, by using the LM as a hidden Markov model (HMM) where the word/event (word/non-EOU or word/EOU) pairs correspond to states and the words to observations, with the transition probabilities given by the N-gram language model. Then, the forward-backward probabilities could be used to obtain the probability of EOU in the current boundary [12]. In our case, however, we are restricted to using only past information. Therefore, we use only N-gram probabilities conditioned on the word history.

2.3. Knowledge source combination

To make use of both prosodic and word information, we compute the following score at each boundary for each applicable DP:

$$Sc(DP) = P^{\alpha}_{DT(DP)} P_{LM} \quad (1)$$

where $P_{DT(DP)}$ is the probability given by the tree trained for the decision point DP, P_{LM} is the probability given by the EOU language model, and α is a prosody weight parameter that is empirically optimized. This combination method represents a simple log-linear interpolation of the two predictors. Other combination approaches are clearly possible (such as training a single classifier that uses both prosodic and word features), but have not been examined yet. The score $Sc(DP)$ will be compared with the score threshold for each applicable DP as shown in Figure 1.

3. EXPERIMENTS

3.1. Methodology

We tested our approach on the ATIS (Air Travel Information System) corpus [13]. The training set consisted of 15,843 utterances and their word transcriptions. We consider each sentence (as indicated by standard punctuation in the transcripts) as one utterance unit. Very few waveforms in the database contain more than one sentence (fewer than 1%). In these cases, we treat each sentence end as its own EOU to be detected.

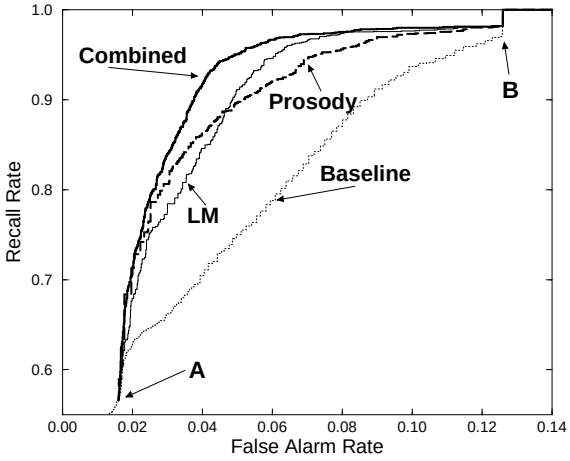

Fig. 2. ROCs for the four systems (only the portion of the graph were the systems differ is shown).

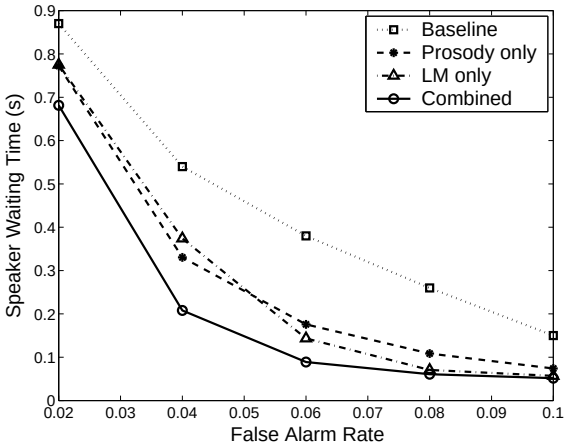

Fig. 3. Speaker waiting times for the four systems.

	EER	SWT (s)
Baseline (pause threshold)	8.9%	0.200
Prosodic decision trees only	6.7%	0.152
Language model only	5.8%	0.159
Prosody + LM combined	4.9%	0.135

Table 1. Equal error rates and speaker waiting times for the four systems.

Experiments used the SRI Decipher^{TM} recognition engine with acoustic models trained for the December 1994 ATIS evaluation [14]. The recognizer language model and the EOU language model were trained using the transcription of the training set, representing a total of 164,000 words. Both models made use of hand-defined, task-specific word classes for airline names, cities, etc., to improve generalization.

We generated forced alignments for the training set, and derived the prosodic features from the resulting phone-level time marks and the acoustic signal. Decision trees were trained using those features. We used 80% of the training set for tree induction. The remaining 20% were used as a tuning set for choosing the best set of features for each DP via an automatic feature subset selection algorithm [9]. This tuning set was also used to optimize the α weight for the combined system.

A separate set of 1,976 utterances was used for testing. For expediency, 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 computed from phone-level alignments and pitch tracks, as in training.

The overall accuracy of the recognizer on our 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.

Our proposed EOU detection system was compared with a baseline system. The baseline system classifies a boundary as an EOU whenever the pause duration found by the recognizer for that boundary is greater than a given pause duration threshold. To test our systems we set decision points at 30, 60, 90, 150, 250, 500, and 800 ms into a pause. The maximum pause duration, beyond which a boundary is always deemed to be an EOU, was set at 1 second.

3.2. Results

Figure 2 shows the receiver operating characteristic (ROC) plot for each of the four systems: baseline, prosody only, LM only, and the combined system, with an optimized $\alpha = 0.8$. For the baseline, the ROC curve is obtained by varying the pause duration threshold, while for the other systems the curve is obtained by varying the score threshold and keeping the maximum pause duration threshold at 1 second. In this way, when the score threshold reaches the maximum value, no sample will be classified as an EOU unless its pause duration exceeds 1 second. The four curves meet at that point ("A" in the figure). The curves also meet at the other end (point "B"), corresponding to the case where the score threshold for the proposed systems and the pause duration threshold for the baseline tend toward zero, classifying all pauses as EOUs.

Clearly, at operating points between 2% and 12% false alarm rate, all proposed systems consistently outperform the baseline system. Furthermore, and interestingly, *the prosody and LM systems are complementary*: the prosody-only system performs better than the LM-only at lower false alarm rates (around 4%), while the LM-only system is better at higher false alarm rates (around 7%). The combined system consistently outperforms both individual knowledge sources.

Our other performance measure of interest is the "speaker waiting time" (SWT), that 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. As shown in Figure 3, the SWT for every presented system is substantially shorter than for the baseline at every false alarm level between 2% and 10%. On this measure, too, the combined system is better than the prosody-only or the LM-only systems.

Table 1 shows the equal error rates (EER) for each of the four systems, along with the speaker waiting times (SWT) for corresponding rates of false alarms. (The EER is the false alarm rate at the operating points where it equals the rate of missed EOUs ($= 1-$ recall rate), and is often used to summarize detection performance in a single number.)

These results show that both the prosodic decision trees alone and the LM alone beat the baseline system in accuracy and speed. The combined system yields an even bigger improvement in both performance measures, with a relative reduction of 45% in the error rate.

4. FEATURE USAGE AND EXAMPLES

Due to lack of space, we cannot report on the features used in all of the decision trees at various DPs, but fortunately, since trees were similar across DPs, we can provide a good picture by reporting feature usage for the tree used at the earliest DP (thus affecting the most decisions). We measure feature usage as the percentage of decision tree tests involving a certain feature, averaged over the entire training data. Results for this tree show that, of all feature tests, 72% involved three features that all capture speaker-normalized pitch range. When speakers are near the bottom of their pitch range, they are more likely to have finished speaking than if they are somewhere in the middle of their range. One of the three features is independent of any alignment information, and all are independent of word identities, suggesting that the features could be used in the absence of speech recognition. Duration features played a smaller role, contributing 21% of the total feature usage. The useful features captured normalized durations of syllable rhymes, and thus did make use of hypothesis alignment information. Since our DPs are conditioned on the presence of a pause, we expect preboundary syllable lengthening (relative to rhymes not followed by a pause). Hesitation boundaries show even *more* lengthening than do EOU boundaries, making duration a useful feature for discriminating the two cases.

The examples below are prototypical cases that show how the prosodic decision trees and the language model can correct EOU detection errors. They were obtained from systems configured to have an average false alarm rate of 4%.

The following is an example for which the baseline system finds a false EOU where the speaker made a long pause (longer than the baseline threshold), while the prosody-only system correctly classifies the pause as non-EOU.

> **August $\uparrow$ twenty six**

where $\uparrow$ marks the point at which the baseline system mistakenly detected an EOU. When listening to the acoustic signal, one notices that the pitch before this long pause does not fall as it usually does in actual EOUs. That is why the prosody-only system, using pitch information, does not confuse this boundary with an EOU.

The second example shows a case where the prosody-only system gives a false alarm, while the combined system (prosody with LM) does not.

> **What time is your flight from Atlanta $\uparrow$ to Boston?**

where $\uparrow$ marks the point at which the prosody-only system mistakenly found an EOU. In this case the word "Atlanta" is uttered with a pitch fall similar to those encountered at the end of utterances, leading to a false EOU detection. When using the combined system, the LM gives a very low probability to "from Atlanta" being at the end of a sentence (in ATIS, phrases like these are usually followed by "to . . . "), thereby correctly classifying the boundary as non-EOU.

5. CONCLUSIONS

We have shown a combined approach for the online detection of ends of utterances in speech transcribed by an automatic recognizer. The system combines prosodic and language model knowledge sources, modeled by decision trees and N-grams respectively.

The equal error rate for the combined prosody/LM system is 4.9%, representing a 45% relative improvement compared to a baseline system that uses only pause duration for end-of-utterance detection. In addition, the speaker waiting time is substantially shortened in the proposed system; for example, at a false alarm rate of 6% the average speaker waiting time is reduced from 380 ms for the baseline system to 90 ms for the combined system.

6. ACKNOWLEDGMENTS

We thank Harry Bratt and Kemal Sönmez for developing the pitch stylizer used in feature computation, and for significant helpful discussions. Kristin Precoda gave valuable comments on a draft of the paper. We also acknowledge Patti Price for an early suggestion to use prosody for end-of-utterance detection. This work was funded by NASA under NCC 2-1256, by DARPA under contract N66001-99-D-8504, 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.

7. REFERENCES

[1] Nuance Communications, *Nuance Speech Recognition System, Version 6, Developer's Manual*, Menlo Park, CA, 1997.

[2] B. Carpenter, Personal communication, March 2002.

[3] World Wide Web Consortium, "Voice Extensible Markup Language (VoiceXML) version 2.0", http://www.w3.org/TR/2001/WD-voicexml20-20011023/, Oct. 2001.

[4] R. Hariharan, J. Häkkinen, and K. Laurila, "Robust end-of-utterance detection for real-time speech recognition applications", *in Proc. ICASSP*, vol. 1, pp. 249–252, Salt Lake City, May 2001.

[5] M. Savoji, "A robust algorithm for accurate endpointing of speech", *Speech Communication*, vol. 8, pp. 45–60, 1989.

[6] Q. Li, J. Zheng, Q. R. Zhou, and C. Lee, "A robust, real-time endpoint detector with energy normalization for ASR in adverse environments", *in Proc. ICASSP*, vol. 1, pp. 233–236, Salt Lake City, May 2001.

[7] L. Huang and C. Yang, "A novel approach to robust speech endpoint detection in car enviroments", *in Proc. ICASSP*, vol. 3, pp. 1751–1754, Istanbul, June 2000.

[8] K. Iwano and K. Hirose, "Prosodic word boundary detection using statistical modeling of moraic fundamental frequency contours and its use for continuous speech recognition", *in Proc. ICASSP*, vol. 1, pp. 133–136, Phoenix, AZ, Mar. 1999.

[9] 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.

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

[11] 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.

[12] A. Stolcke and E. Shriberg, "Automatic linguistic segmentation of conversational speech", in H. T. Bunnell and W. Idsardi, editors, *Proc. ICSLP*, vol. 2, pp. 1005–1008, Philadelphia, Oct. 1996.

[13] 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.

[14] 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.