AUTOMATIC DETECTION OF SENTENCE BOUNDARIES AND DISFLUENCIES BASED ON RECOGNIZED WORDS

Andreas Stolcke[1] *Elizabeth Shriberg*[1] *Rebecca Bates*[2] *Mari Ostendorf*[2] *Dilek Hakkani*[1]
Madelaine Plauche[1] *Gökhan Tür*[1] *Yu Lu*[1]

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

[2]Electrical and Computer Engineering Department
Boston University, Boston, MA, USA
http://raven.bu.edu/

ABSTRACT

We study the problem of detecting linguistic events at interword boundaries, such as sentence boundaries and disfluency locations, in speech transcribed by an automatic recognizer. Recovering such events is crucial to facilitate speech understanding and other natural language processing tasks. Our approach is based on a combination of prosodic cues modeled by decision trees, and word-based event N-gram language models. Several model combination approaches are investigated. The techniques are evaluated on conversational speech from the Switchboard corpus. Model combination is shown to give a significant win over individual knowledge sources.

1. INTRODUCTION

Current automatic speech recognition systems output a string of words. Most natural language understanding systems, however, require structural information such as punctuation, which is present in text but not overtly indicated in spoken language. Similarly, for speech understanding and information extraction, it is important to find the location and extent of disfluencies (including self-repairs), so that a speaker's intended meaning can be inferred. We will refer to sentence boundaries and disfluencies collectively as our target "events."

Prior work on utterance boundary detection [8, 12] as well as on disfluency detection [5, 10] has addressed this problem, but not in a completely realistic framework. Previous work has assumed either a correct word sequence, or knowledge of the word boundaries. In reality, word information is not known, but has to be hypothesized using a speech recognizer. This renders word-based cues less reliable and increases the importance of prosodic cues. It also raises the question of how the various cues are to be combined taking unreliable word information into account.

2. METHOD

2.1. Data

Speech data consisted of more than 1100 conversations from the Switchboard corpus of human-human telephone dialogs on prescribed topics [4]. The data set represents over 350 different speakers (45% male, 55% female). The corpus was partitioned into three portions: 1794 conversation sides (1.2M words) were used for model training; 436 conversation sides (231K words) were used for development and testing on data that had been transcribed by humans; and a small, 19 conversation (18K words) set was decoded with a large-vocabulary speech recognizer and used for tests that involved automatic transcription. There was no speaker overlap between the three corpus subsets.

Event class	Tag	Freq.	Example
Sentence boundary	**S**	10.8%	I haven't seen it $*$ Not sure I like it
Filled pause	**FP**	2.9%	he uh $*$ liked it
Repetition	**REP**	1.9%	he $*$ he liked it
Deletion	**DEL**	1.3%	it was $*$ he liked it
Repair	**OthDF**	1.2%	he $*$ she liked it
Else/fluent	**else**	81.8%	she $*$ liked it

Table 1: Boundary and disfluency event classes.

We prepared a speech database that combines information from various sources, and at various levels of resolution, including:

- Word transcripts
- Hand-labeled disfluency annotations and sentence segmentations prepared by the Linguistic Data Consortium [9]
- Phone-level time marks produced by forced alignment of the word transcripts using a non-speaker-adapted version of the SRI Decipher(TM) speech recognizer used in the 1997 LVCSR evaluations [7]
- Raw acoustic measurements for the prosodic features described below, such as fundamental frequency (F0) and signal-to-noise ratio (SNR) values.

The resulting database contained all information in a time-aligned format. For the automatically transcribed test set we created the alignments and derived information for the top 100 hypotheses generated by the recognizer.

For this study, we grouped the linguistic events at word boundaries into six classes, comprising the major types of boundaries and disfluencies we were interested in. These events are mutually exclusive, and exactly one event occurs at any given interword boundary. Table 1 summarizes these event classes, their relative frequencies, and gives an example for each. The distribution of event types is highly skewed; close to 82% percent of events are fluent, sentence-internal word transitions. This makes it difficult for automatic classifiers to learn the distinctive features of the much less frequent, marked event types.

2.2. Modeling

To build an automatic detector of interword events we need to model the relationships between the following entities:

- A, the acoustic features used by the recognizer
- F, the prosodic features
- W, the string of spoken words $W_1, W_2, \ldots$
- E, the sequence of interword events $E_1, E_2, \ldots$

A practical constraint for our work was to retain the standard components of a speech recognizer, i.e., the **acoustic model** $P(A|W)$ that characterizes how well an acoustic observation matches a given word sequence, and the **word language model** $P(W)$ that estimates the *a priori* probability of a word sequence. Both models are combined to give the *a posteriori* probability $P(W|A)$ of a word sequence. Acoustic and word language models were of standard varieties, i.e., state-clustered Gaussian mixtures [3] and backoff trigrams [6], respectively.

In addition to the standard models, the following statistical models capture the relationship between interword events and their cues. The **prosodic model** $P(E|F,W)$ predicts events from their prosodic correlates. Finally, the **event language model** $P(W,E)$ describes the joint distribution of words and the intervening events. The four model components are combined to estimate the posterior event probabilities $P(E|A,F)$.

All our models have the property that the posteriors of individual events E_i in E are estimated, not E as a whole. This is both convenient and legitimate, since the overall classification error is minimized by maximizing the posterior of each E_i independently.

2.3. Prosodic Model

As in prior work on disfluency and sentence boundary detection [8, 10], we trained CART-style decision trees [2] to predict event classes from local properties at the word boundary of interest. However, we use the trained tree models not simply as classifiers that output the most likely class, but as probability estimators $P_{\mathrm{T}}(E_i|F,W)$ to be combined with the other components.

We experimented with a large collection of features capturing the three major aspects of speech prosody:

- Duration: of pauses, final vowel and final rhymes, normalized both for phone durations and speaker statistics
- Pitch: F0 patterns, preceding the boundary, across the boundary, and pitch range relative to the speaker's baseline
- Energy: signal-to-noise ratio using a front-end tuned for this corpus, to capture energy fluctuations not due to channel

While the feature extraction makes extensive use of the forced alignment of words to the speech signal (e.g., to extract phone durations), it is important to note than none of our features encoded the identity of words directly, affording some degree of independence from the word-based cues. This will be important later on during model combination.

2.4. Event Language Model

The event language model describes the joint distribution of words and events, $P_{\mathrm{LM}}(W,E)$. We treated words and events as a single token stream, as described below. During testing, the model can be used as a hidden Markov model (HMM) in which the word/event pairs correspond to states and the words to observations, with the transition probabilities given by the N-gram model. The model is a generalization of the hidden segment boundary language model used in [12] where the number of events types and the context length can be arbitrary. Given a word sequence, a forward-backward dynamic programming algorithm is used to compute the posterior probability $P_{\mathrm{LM}}(E_i|W)$ of an event E_i at position i.

We trained a word/event N-gram model from 1.2M words of transcripts that had been hand-labeled for the events of interest [9]. Each event was represented by an additional non-word token, with two exceptions. First, we omitted event tags for filled pauses, since they are redundantly encoded by the preceding word ("uh" or "um"). Second, we did not represent the fluent, intra-sentence boundary events explicitly, since they are implied by the absence of any other event tag or filled pause word. These two conventions lead to a more compact encoding of events and make better use of the limited context of the N-gram model. A 4-gram model was used for all results reported here, i.e., a word or event was conditioned on no more than three preceding words and/or events.

Certain other kinds of information, such as the location of speaker changes (turn boundaries) and long pauses (where the waveforms had been cut for processing purposes) can be conveniently encoded in the language model as well, and are known to improve its quality for speech recognition purposes [13, 11]. While conceptually this information is part of the prosody, it is an empirical question whether turn and pause information is best encoded in the prosodic model, the language model, or both. Therefore, we created two versions of the event N-gram model, one containing such segmentation cues ("Seg N-gram"), and one without ("No-seg N-gram").

3. EXPERIMENTS

3.1. Methodology

We tested each of the three models (prosodic decision tree, event language model with and without segmentation) in isolation for their event detection accuracy. This was performed first on a test set with known words, and then on recognizer output. The recognizer word error on the test set was 46.8%, i.e., on average almost every other word was incorrect.[1]

To run the event detector on recognizer output, we adopted the expedient of simply conditioning the event models on the 1-best hypothesis:

$$P(E|F,A) \approx P(E|F,\operatorname*{argmax}_W P(W|A))$$

This approach is suboptimal if event detection is the overall goal, in that multiple hypotheses other than the best one might conspire to raise the overall probability of an event above the one

[1] Note that we left a number of features out of the recognizer (such as speaker adaptation) which would have either created a significant computational burden or an acoustic modeling mismatch between training and test sets. This resulted in performance somewhat below the current state-of-the-art.

Model Type	Known Words	Recognized words	
	% correct	% correct	% accuracy
Chance	81.8	72.3	69.2
Prosodic Tree	88.9	76.1	72.9
No-seg N-gram	90.0	74.4	71.1
Seg N-gram	92.7	77.0	73.8

Table 2: Event recognition performance for three knowledge sources. All score differences are significant by a Sign test ($p < .0001$).

		Detected Events					
		S	else	FP	DEL	OthDF	REP
True Events	S	16880	5065	0	111	31	33
	else	3594	162847	0	439	174	170
	FP	0	0	5879	0	0	0
	DEL	393	1524	0	660	131	48
	OthDF	218	1338	0	191	442	341
	REP	43	892	0	34	97	2856

Table 3: Confusion matrix for segment-N-gram event classifier on known words.

supported by the top hypothesis. In other words, for some applications we might want to sum event posteriors over the entire N-best list:

$$P(E|A) = \sum_W P(E|W,F)P(W|A) \quad .$$

This approach is complicated by the need to identify corresponding events in hypotheses that differ in their words; we plan to study this approach in future work.

A related problem concerns the scoring of event detection accuracy given that the number of words (and hence events) differs between reference and hypothesis. For this study, we aligned the word/event strings and then counted the number of mismatched events, as well as the number of events inserted and deleted. Similar to the scoring practice used in speech recognition, we report both the percentage of correctly identified true events, and the accuracy (1 – the number of event substitutions, deletions and insertions divided by true total). A more stringent error criterion might require the event times to match up as well.

3.2. Results

Table 2 summarizes the results obtained for the three models in each of two test conditions: known and recognized words. Chance performance (obtained by labeling each word boundary as the **else** event) is also given for reference. Results show that the N-gram with segment information performs significantly better than either the prosody model (which also has access to segment information) or the N-gram without segment boundaries.

Table 3 shows a confusion matrix for the segmentation LM on known words. The matrix indicates that the infrequent disfluency types (deletions and other repairs) are particularly difficult to detect. This could be both because of their low frequency, their lack of distinct lexical cues, or both.

When the same three models are applied to speech recognizer output, we observe a substantial degradation in event detection accuracy. As expected, the word-based models suffer most from recognition errors in relative terms. In this condition, the no-segmentation LM performs worse than the prosody model. Notice that the prosody is also negatively impacted by word recognition errors, since its input features depend on phone alignments and word boundary hypotheses. However, these seem to be more robust to errors than information based on word identity.

A general point about our paradigm is that only data based on (automatically aligned) correct words are used for model training, thus creating an inherent mismatch when testing on partially incorrect words. We made this choice because recognition of large amounts of speech data is a considerable computational task. Thus, an overall improvement is expected if we trained models on actual recognizer output, allowing the models to partially compensate for systematic recognizer errors.

4. MODEL COMBINATION

4.1. Approaches

The goal of model combination is to make the best use of all available knowledge sources while keeping the modeling computationally and statistically tractable. For example, it is not feasible to explicitly model all combinations of word identity and prosodic features because of the resulting large input feature space.

So far we have experimented with three different model combination approaches:

- **Model interpolation.** This is a weak approach that treats multiple knowledge sources as alternative estimators of the same probability distribution, which are combined by linear interpolation. In our case, we combine the prosodic posterior and the event LM posterior using an empirically optimized weighting:

  $$P(E_i|F,W) \approx \lambda P_{\mathrm{T}}(E_i|F,W) + (1-\lambda)P_{\mathrm{LM}}(E_i|W) \quad .$$

  A more refined (but as yet unimplemented) version is the *mixture of experts* where λ is replaced by a function of W and F.

- **Independent model combination.** In this approach we assume that the prosodic feature F are largely independent of the words W when conditioned on the events: $P(F|E_i,W) \approx P(F|E_i)$. This allows the following decomposition:

  $$P(E_i|F,W) = \frac{P_{\mathrm{LM}}(E_i|W)P(F|E_i)}{P(F|W)} \quad .$$

  The denominator does not depend on E_i and so can be ignored for classification purposes. $P(F|E_i)$ is proportional to $\frac{P(E_i|F)}{P(E_i)}$, and can be directly estimated by a prosodic tree that is trained on a uniform distribution of event classes. As in the previous approach we introduce an empirically determined balancing parameter λ to adjust the dynamic ranges of the two models, giving us

  $$P(E_i|F,W) \propto P_{\mathrm{LM}}(E_i|W)^{(1-\lambda)} \left(\frac{P_{\mathrm{T}}(E_i|F)}{P(E_i)} \right)^{\lambda} \quad .$$

- **Joint modeling.** Various approaches exist to allow training of a single classifier that takes both word and prosodic information as input, while avoiding the large input space if

Model Type	Known Words	Recognized words	
	% correct	% correct	% accuracy
Seg N-gram	92.7	77.0	73.8
Interpolation	93.0	78.1	74.9
Independent	93.0	77.4	74.1
Joint Tree	93.1	76.6	73.3

Table 4: Event recognition performance for various model combination strategies. All score differences are significant by a Sign test ($p < .005$).

words were encoded as atomic feature values [1, 5]. We experimented with a very simple technique where the word-based posterior probabilities are used as additional input features to the prosodic decision tree. The tree, while not having direct access to word identities, can model correlations between the word-based LM decisions and prosodic features.

4.2. Results

Table 4 shows event classification accuracies for the three model combination approaches, for both known and recognized words. For comparison, the Seg-N-gram results are repeated as a baseline. The model interpolation approach is seen to be the most robust model combination approach so far. It yielded virtually identical relative error reduction (4%) over the N-gram classifier alone, on both known and recognized words.

While the joint tree seems to have a slight edge on known words, it predictably fares poorly on recognized words. A likely explanation is that the posterior LM probabilities the tree is trained on come from correct words. Testing on recognized words renders these input features very noisy, creating a train/test mismatch. The approach is expected to work better if large amounts of recognizer output were available for training the joint model.

The independent combination approach performs reasonably well, though not as well as interpolation on recognized words. The results given here actually used the Seg-N-gram in the combination, which violates the independence assumption of the approach, since the LM makes use of some of the same turn and pause-related information as the prosodic model. However, when using the No-seg N-gram in the combination instead, accuracy went down by about 1-2%. The likely reason is that the N-gram makes more effective use of turn and pause information; thus, omitting it hurts the overall model more than the independence violation.

5. CONCLUSIONS

We have demonstrated a combined approach for the detection of interword events (sentence boundaries and four classes of disfluencies) on spontaneous speech transcribed by an automatic recognizer. The system combines prosodic and language model knowledge sources, modeled by decision trees and N-grams, respectively. Event detection accuracy is about 75% (78% correct) on recognizer output with 46.8% word error, as compared to 93% correct on human transcripts. In both test conditions, the combination of prosodic and word N-gram models gives a 4% relative error reduction over the most powerful knowledge source, an N-gram that includes turn and pause information.

The results reported here should be regarded as a baseline for future work. For example, the overall model could be improved by including parts-of-speech (POS) or other syntactic information in the event model. (We showed in [12] that using POS improves sentence boundary detection, and [5] observed that POS modeling enhanced disfluency detection.) Other directions for future improvement include event posterior probability combination across multiple N-best hypotheses, improved prosodic feature, and more sophisticated model combination.

Acknowledgments

This research was supported by DARPA and NSF, under NSF Grants IRI-9314967, IRI-9619921 and IRI-9618926, and by the Naval Command, Control, and Ocean Surveillance Center under contract no. N66001-97-C-8544. The views herein are those of the authors and should not be interpreted as representing the policies of the funding agencies.

6. REFERENCES

1. L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. A tree-based statistical language model for natural language speech recognition. *IEEE ASSP*, 37(7):1001–1008, 1989.
2. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.
3. V. Digalakis and H. Murveit. GENONES: An algorithm for optimizing the degree of tying in a large vocabulary hidden Markov model based speech recognizer. In *Proc. ICASSP*, vol. 1, pp. 537–540, Adelaide, Australia, 1994.
4. 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.
5. P. Heeman and J. Allen. Intonational boundaries, speech repairs, and discourse markers: Modeling spoken dialog. In *Proc. ACL/EACL*, Madrid, 1997.
6. S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE ASSP*, 35(3):400–401, 1987.
7. *Conversational Speech Recognition Workshop DARPA Hub-5E Evaluation*, Baltimore, MD, 1997.
8. 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.
9. M. Meteer et al. Dysfluency annotation stylebook for the Switchboard corpus. Distributed by LDC, `ftp://ftp.cis.upenn.edu-/pub/treebank/swbd/doc/DFL-book.ps.gz`, 1995. Revised June 1995 by Ann Taylor.
10. 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.
11. A. Stolcke. Modeling linguistic segment and turn boundaries for N-best rescoring of spontaneous speech. In G. Kokkinakis, N. Fakotakis, and E. Dermatas, editors, *Proc. EUROSPEECH*, vol. 5, pp. 2779–2782, Rhodes, Greece, 1997.
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, 1996.
13. T. Zeppenfeld, M. Finke, K. Ries, M. Westphal, and A. Waibel. Recognition of conversational telephone speech using the Janus speech engine. In *Proc. ICASSP*, vol. 3, pp. 1815–1818, Munich, 1997.