# RERANKING FOR SENTENCE BOUNDARY DETECTION IN CONVERSATIONAL SPEECH

*Brian Roark,[a] Yang Liu,[b] Mary Harper,[c] Robin Stewart,[d] Matthew Lease,[e] Matthew Snover,[f]*
*Izhak Shafran,[g] Bonnie Dorr,[f] John Hale,[h] Anna Krasnyanskaya,[i] and Lisa Yung,[g]*
[a] OGI/OHSU; [b] UT, Dallas; [c] Purdue; [d] Williams; [e] Brown; [f] U. of Maryland; [g] Johns Hopkins; [h] Michigan State; [i] UCLA

## ABSTRACT

We present a reranking approach to sentence-like unit (SU) boundary detection, one of the EARS metadata extraction tasks. Techniques for generating relatively small n-best lists with high oracle accuracy are presented. For each candidate, features are derived from a range of information sources, including the output of a number of parsers. Our approach yields significant improvements over the best performing system from the NIST RT-04F community evaluation[1].

## 1. INTRODUCTION

Automatic speech recognition (ASR) system quality is typically measured in terms of the accuracy of the word sequence. However, automated speech processing applications may benefit from (or sometimes even require) system output that is richer than an undelimited sequence of recognized words. For example, sentence breaks and disfluency annotations are critical for legibility [1], as well as for downstream processing algorithms with complexity that is polynomial in the length of the string, such as parsing. One aspect of the DARPA EARS program[2] was to focus on structural metadata extraction (MDE) [2], including a range of disfluency annotations and sentence-like unit (SU) boundary detection.

This paper specifically addresses the task of SU boundary detection. Previous approaches to this task have used finite-state sequence modeling approaches, including Hidden Markov Models (HMM) [3] and Conditional Random Fields (CRF) [4]. While these approaches have yielded good results, the characteristics of this task make it especially challenging for Markov models. Average SU length for conversational telephone speech is around 7; hence, most of the time the previous states will be for non-boundary positions, providing relatively impoverished state sequence information. Thus, in [4], a Maximum Entropy (MaxEnt) model that did not use state sequence information, was able to outperform an HMM by including additional rich information. Our approach is to rely upon a baseline model [5] to produce n-best lists of possible segmentations, and extract disambiguating features over entire candidate segmentations, with no Markov assumption. This paper presents an effective n-best candidate extraction algorithm, along with a detailed investigation of the utility of a range of features for improving SU boundary detection.

In the next section we provide background on the SU detection task, baseline models, and the general reranking approach. We then present our n-best extraction algorithm and the features we investigated, followed by empirical results under a variety of conditions.

## 2. BACKGROUND

In this section, we provide background on the baseline SU detection models and our reranking approach.

### 2.1. MDE tasks and baseline models

There are four tasks for structural MDE in EARS in the most recent evaluations: SU detection, speech repair detection, self-interruption point (IP) detection, and filler detection. Evaluation is conducted using human reference transcriptions (REF) and ASR output, the latter to assess the impact of recognition errors. Two corpora with different speaking styles were used in EARS: conversational telephone speech (CTS) and broadcast news. Performance is generally measured as the number of errors (insertion, deletion, and substitution when subtype of the events is considered) per reference events (e.g, SU boundaries, speech repairs), which we will refer to as NIST error. We also report F-measure accuracy[3] for SU detection.

We now briefly summarize the ICSI/SRI/UW MDE system [5], which is the baseline for the current research. The MDE tasks can be seen as classification tasks that determine whether an interword boundary is an event boundary (e.g., SU or IP) or not. To detect metadata events, multiple knowledge sources are utilized, including prosodic and textual information. Typically, at each interword boundary, prosodic features are extracted to reflect pause length, duration of words and phones, pitch contours, and energy contours. These prosodic features are modeled by a decision tree classifier, which generates a posterior probability of an event given the feature set associated with a boundary. Textual cues are captured by contextual information of words, their corresponding classes, or higher-level syntactic information.

Three different Markov modeling approaches are baselines for the MDE tasks: HMM, MaxEnt, and CRF. In all cases, there is a hidden event ($\boldsymbol{E}$) at each word, representing the segmentation decision following the word. There are also features ($\boldsymbol{F}$) corresponding to the observed input, e.g., the words and prosodic features. The HMM is a second order Markov model, the CRF model first order, and the MaxEnt model order 0. Both the MaxEnt and CRF models are trained using conditional likelihood objectives; whereas, the HMM is trained as a generative model. The CRF model has been shown to outperform the MaxEnt model, which outperforms the HMM [4]. Baseline results will be presented in section 4.

[1] http://www.nist.gov/speech/tests/rt/rt2004/fall/

[2] http://www.darpa.mil/ipto/programs/ears/

[3] If $\boldsymbol{c}$, $\boldsymbol{s}$, and $\boldsymbol{r}$ are the number of correct, system, and reference SU boundaries, respectively, then F=$\boldsymbol{2c/(s+r)}$.

| Baseline Posterior $x$ | Percent Accurate | Percent of Word Boundaries |
|---|---|---|
| $x > 0.95$ | 97.9 | 8.2 |
| $x < 0.05$ | 99.4 | 77.0 |
| $0.05 \leq x \leq 0.95$ | 78.1 | 14.8 |

**Table 1**. Accuracy of labels produced by baseline model versus posterior probability on dev2 set.

### 2.2. Maximum Entropy reranking

We used the MaxEnt reranker documented in [6], which optimizes parameter weights with respect to a regularized conditional likelihood objective. The approach maximizes the conditional probability mass[4] given to those candidates in the n-best list that have the highest accuracy[5]. A Gaussian regularizer is used to control for overtraining, and the regularizer constant is set empirically. Formally, let the training data consist of $N$ examples, let $Y_i$ be the set of candidates for example $i$, and let $\hat{Y}_i$ be the set of maximum accuracy candidates for example $i$, i.e.,

$$\hat{Y}_i = \{y \in Y_i \mid \text{ACC}(y) = \text{argmax}_{y' \in Y_i} \text{ACC}(y')\} \quad (1)$$

Then the parameter estimation minimizes the regularized negative conditional log probability

$$NLL_R(\theta) = -\sum_{i=1}^{N} \log \left( \sum_{y \in \hat{Y}_i} \text{P}(y \mid Y_i) \right) + R(\theta) \quad (2)$$

where $\text{P}(y \mid Y_i)$ is the conditional likelihood of $y$ given the normalized distribution over $Y_i$, and $R(\theta)$ is the Gaussian regularizer. By normalizing the candidate probabilities over the n-best lists, this approach uses a global normalization akin to that in CRFs. It is the flexibility in extracting features over the entire sequence that differentiates the current approach from the baseline CRF model. See [6] for more details.

## 3. RERANKING FOR SU DETECTION

In this section, we present n-best list generation and feature extraction for reranking.

### 3.1. n-best candidate extraction

SU detection is done over conversation sides (see section 4 for the data description), which are far lengthier sequences than those in NLP tasks that are usually approached in a reranking paradigm, such as parsing. For the data processed in this paper, the average length of a conversation side is more than 500 words, with the maximum over 1000. Since every word boundary is a potential segmentation point, the number of possible segmentations over the conversation side is exponential in the length of the string. A brute force approach to generating the 1000 best segmentations over these conversation-side sequences, i.e., choosing the 1000 highest scoring segmentations based on the baseline posterior probabilities, yields an oracle F-measure accuracy of just a percent or so better than the 1-best. As a result, other methods for generating the n-best lists were investigated.

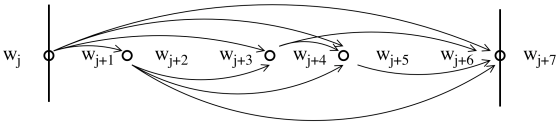


**Fig. 1**. Picture of the n-best extraction approach

The baseline classifier marks as segmentation points all positions with a posterior probability of 0.5 or higher. Table 1 shows the accuracy of the baseline model when the posterior probability of a segmentation boundary is either very high or very low, versus somewhere in the middle. For very high or very low posterior word positions, the baseline model is nearly always correct; whereas, other positions have much lower accuracy for the classifier. This suggests an approach where high and low posterior points are fixed to the classes provided by the baseline classifier.

We used a two-stage candidate generation approach. In the first stage, we fixed a subset of the word boundaries as segmentation points. The sub-sequences between fixed word boundaries we refer to as *fields*. In the second stage, we generated n-best lists of candidate segmentations for the fields. To establish the fields, we first chose all word boundaries with a posterior probability $x > p$ for some parameter $p$. Since we intend to use parsers over candidate segments for feature extraction, it is important that the longest candidate segments be of tractable lengths, given polynomial complexity of parsing. Hence, we also parameterized a maximum field length $k$. For fields of length greater than $k$, the highest posterior internal word boundary was fixed as a segmentation point. This process is continued until no fields have length greater than $k$.

In the second stage, candidate segmentations are created for each field. This is also done using two parameters. The $j$ highest posterior internal points are hypothesized as possible boundaries, although no internal point with posterior less than $q$ is allowed to be hypothesized as a segmentation point. With $j$ internal hypothesized points, there are $2^j$ candidate segmentations. Figure 1 presents a simple picture of how this candidate generation works, where three internal segmentation points are placed between the field boundaries, providing 8 candidate segmentations. For the trials presented in this paper, we chose to generate lists for all sections under the parameterization $p = 0.95$, $k = 50$, $j = 10$, and $q = 0.05$. This yields lists with an F-measure oracle accuracy of 97.4 on the reference transcribed development set (dev2).

### 3.2. Feature extraction

Features are extracted from each candidate segmentation, for use in the reranker. We will assign letters to sets of features, which will allow us to present empirical results comparing performance on the development set using subsets of features.
**a. Baseline model score:** the baseline posterior score associated with the candidate segmentation. Recall that the candidates are defined by some $j$ field-internal possible segmentation points. Each word boundary has a posterior probability $x$

[4]The probability is conditioned on the n-best list, i.e., each candidate's probability is normalized relative to the n-best list.

[5]Since the n-best list is not guaranteed to contain the truth, there may be multiple maximum accuracy candidates in the list.

of being a segmentation point (calculated using the forward-backward algorithm), and $1-x$ of not being a segmentation point. The score of each candidate is the sum of the log posterior probabilities for its label at each possible boundary.

**b. Candidate statistics:** features derived from characteristics of the candidate. These included: the number of segments in the candidate; the maximum (and minimum) segment length within the candidate; and the average segment length. We also extracted segment-length n-grams (up to 4), where each segment is assigned a bin based on the number of words in the segment, and sequences of bins were extracted as features.

**c. N-gram score:** each candidate received a trigram score, using a model trained with segmentation points as tokens, on 1500 hours of Fisher data rapidly transcribed by WordWave.

**d. Baseline model disfluency features:** whether the baseline model labeled the word just before, just after, or both just before and just after a segment boundary as a speech repair.

**e. ToBI-label features:** annotation of a reduced set of ToBI-based labels. Automatic decision-tree based classifiers were built from a subset of Switchboard that was manually annotated with ToBI labels [7]. Certain labels occurred infrequently and were collapsed to obtain three types of break indices – fluent (low), disfluent (disf) and major phrase boundaries (high). The posterior probability of these three break indices was calculated from acoustic cues of prosody such as pitch, energy and duration associated with words, syllable, rhymes and phones (see [8] for a detailed description of the features). Note, these raw acoustic cues are already present in our baseline system. In addition, however, we also explicitly combined a quantized version of posterior probabilities with the size/type of the largest syntactic constituent that begins/ends at a word, and the type/distance of the dependency. The constituents were taken from the Charniak parser output.

Other features were extracted from segments, regardless of the candidate within which they occur. Candidates then sum the feature values from the segments of which they are composed. The benefit of extracting them over segments rather than candidates is that, while the number of candidates grows exponentially with the number of internal points $j$, the number of unique segments only grows quadratically. Hence, with 10 internal points, there are 100 not 1000 parsing tasks.

**f. Segment initial and final n-grams:** for all segments in the candidate, unigrams and bigrams both sentence initial and final. Also, POS-tag unitags and bitags in the same positions.

**g. Speaker change and backchannel:** features derived from speaker change events from the two-sided conversations. The features were based on the distance of segment boundaries from both speaker changes and common backchannel words on both sides of the conversation.

We extracted features using three kinds of parsers: (1) the Charniak parser [9, 6] trained on the Switchboard treebank; (2) the Constraint-Dependency Grammar (CDG) parser [10]; and (3) the Minipar dependency tree parser [11]. Note that the CDG parses were generated for these experiments via a conversion from parses output by the Charniak parser. This conversion has been used in the past to generate CDG-derived features for language modeling [12]. Both the CDG and Minipar parses represent dependencies between words and their governors. In addition, the CDG parser represents constraints on required dependents, e.g., verb subcategorization requirements, as well as a variety of lexical features such as agreement and case. We refer readers to the citations for details on these parsers and representations.

**h. Charniak parser features:** each segment in the candidate is assigned features extracted from the Charniak parser output. These include a language model score by the Charniak parser for each segment. Nonterminal labels taken from the 1-best tree returned by the Charniak parser, e.g., S, NP, FRAG, etc., were extracted as features, along with an indication of whether they were at the root of the tree or not. In addition, a second feature combined each root or non-root non-terminal label with the number of children of the node.

**i. CDG-derived dependency features:** generated using the dependency feature template tool described below.

**j. Other dependency features:** generated using the dependency feature template tool described below, based on Minipar and Charniak-derived dependency trees[6].

Features were extracted from Minipar, Charniak-derived, and CDG dependency trees using a feature extraction tool and an associated feature template language. The template language allows for features relating sentence position, lexical identity, part of speech tags, governor relationship, governor types, and other features generated by the particular parser. Some of the features used were:

- Whether word $X$ was the root of the dependency tree
- The count or percentage of governor relationships of type $X$
- The part-of-speech $Y$ and lexical features $F$ of root word $X$
- Whether the root was the first or last word in the sentence
- The number of times word $X$ with part-of-speech $Y$ and governor relationship $Z$ was in the sentence.
- The number of times that a word with part-of-speech $X$ governed a word with part-of-speech $Y$.

## 4. EMPIRICAL RESULTS

The baseline MDE system was trained on roughly 400,000 words of MDE annotated Switchboard transcripts. Three additional sections of largely Fisher (with some Switchboard) data were RT04F SimpleMDE annotated by the LDC: a 75,000 word development set (dev1); a second 35,000 word development set (dev2); and a 35,000 word evaluation set (eval). In addition, we made use of treebanks of these three sections (LDC2005E15). For all three of these sets, both reference transcripts and 1-best ASR[7] transcripts were available.

Table 2 shows results on the dev2 development set, under both reference and ASR 1-best transcript conditions. The two baseline results demonstrate that the combination of the

[6] To derive a dependency tree directly from Charniak parses, we used standard head-percolation techniques to define governor dependencies.

[7] The ASR system output came from [13] and achieved WER of 11.7% on dev2 and 14.9% on eval.

| System | Reference | | ASR | |
|---|---|---|---|---|
| | Feats | NIST | Feats | NIST |
| Baseline CRF | | 25.9 | | 35.8 |
| Baseline HMM+MaxEnt | a | 26.1 | a | 35.1 |
| Reranked | a-e | 25.9 | a,c-e | 34.6 |
| Reranked | a-h | 24.5 | a,c-h | 33.8 |
| Reranked | a-i | 23.2 | a,c-i | **33.5** |
| Reranked | a-j | **23.0** | a,c-j | 33.9 |

**Table 2**. NIST error results with different feature sets on dev2, both reference and ASR 1-best transcripts.

HMM and MaxEnt models, from which we had access to forward-backward calculated posterior probabilities, perform competitively with the CRF model, from which we did not have access to the posteriors, due to the lack of such capability in the third-party toolkit being used [4]. Using the HMM+MaxEnt posteriors as feature **a** from our set, we then trained a reranker on dev1 using subsets of the features defined in the last section[8].

As can be seen from the table, feature sets **a-e** provide small improvements over the baseline for the reference condition, though more for the ASR condition. This difference is due to the ToBI-based features, which were generally found to be more useful in the ASR condition, when transcript-based features were less reliable. Feature set **b** was not found to be useful in the ASR condition, and hence was omitted. Adding features **f-h** provided large improvements under both conditions. Adding the CDG derived features (**i**) provided large improvements in the reference condition, but more modest gains with ASR transcripts. Finally, adding other dependency features (**j**) provided small additional improvements in the reference condition, but no gain with ASR.

The best performing reranker models for the reference and ASR conditions were applied to the eval set, using either dev1 as training, or dev1+dev2 together as training. The results are shown in table 3. This resulted in gains of 2.6 percent for the reference condition, which is statistically significant at $p < 0.00001$; and 1.1 percent for the ASR condition, which is significant at $p < 0.02$, using a matched pair test on segments defined by long pauses [14].

Note that these gains are not simply due to the use of extra training data in dev1 and dev2. Adding these sets to the training data for the baseline model yielded just 0.7 percent NIST error rate reduction on the eval set for the reference condition, and 0.1 percent for the ASR condition.

## 5. CONCLUSION

In summary, we have presented an approach for recasting SU detection as an n-best reranking problem with a relatively small $n$. Using reranking techniques with these n-best lists, we have demonstrated significant improvements over very strong baselines. Future work will include using reranking to optimize segmentation for other objectives, such as downstream processing metrics, e.g., parsing accuracy.

[8]Note that the order in which features are introduced may obscure their contribution to the final full-feature system, which is difficult to tease apart.

| System | Reference | | ASR | |
|---|---|---|---|---|
| | Feats | NIST | Feats | NIST |
| Baseline CRF | | 26.5 | | 37.2 |
| Baseline HMM+MaxEnt | a | 27.0 | a | 36.7 |
| Reranked, dev1 training | a-j | 24.9 | a,c-i | 36.4 |
| Reranked, dev1+2 training | a-j | 24.4 | a,c-i | 35.6 |

**Table 3**. NIST error results of baseline and reranker trained on dev1 and on dev1+dev2 applied to the eval set, both reference and ASR 1-best transcripts

## 6. REFERENCES


[1] D. Jones, F. Wolf, E. Gibson, E. Williams, E. Fedorenko, D. Reynolds, and M. Zissman, "Measuring the readability of automatic speech-to-text transcripts," in *Proc. of Eurospeech*, 2003.

[2] "Simple metadata annotation specification ver. 6.2," Tech. Rep., Linguistic Data Consortium, 2004, www.ldc.upenn.edu/Projects/MDE/Guidelines/SimpleMDE_V6.2.pdf.

[3] E. Shriberg, A. Stolcke, D. Hakkani-Tur, and G. Tur, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech Communication*, pp. 127–154, 2000.

[4] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Using conditional random fields for sentence boundary detection in speech," in *Proc. of ACL-05*, 2005, pp. 451–458.

[5] Y. Liu, E. Shriberg, A. Stolcke, B. Peskin, and M. Harper, "The ICSI/SRI/UW RT-04 structural metadata extraction system," in *Proc. of EARS RT-04 Workshop*, 2004.

[6] E. Charniak and M. Johnson, "Coarse-to-fine *n*-best parsing and maxent discriminative reranking," in *Proc. of ACL-05*, 2005, pp. 173–180.

[7] M. Ostendorf, I. Shafran, S. Shattuck-Hufnagel, L. Carmichael, and W. Byrne, "A prosodically labeled database of spontaneous speech," in *Proc. of the ISCA Workshop on Prosody in Speech Recognition and Understanding*, 2001.

[8] E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke, "Modeling prosodic feature sequences for speaker recognition," *Speech Communication*, vol. 46, no. 3-4, pp. 455–472, 2005.

[9] E. Charniak, "A maximum-entropy-inspired parser," in *Proc. of NAACL-00*, 2000, pp. 132–139.

[10] W. Wang and M. P. Harper, "A statistical constraint dependency grammar CDG parser," in *Proc. of the ACL-04 Workshop on Incremental Parsing*, 2004.

[11] D. Lin, "Dependency-based evaluation of minipar," in *Workshop on the Evaluation of Parsing Systems, Granada, Spain*, 1998.

[12] W. Wang, A. Stolcke, and M. Harper, "The use of a linguistically motivated language model in conversational speech recognition," in *Proc. of ICASSP*, 2004.

[13] B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, G. Zweig, A. Stolcke, R. Gadde, W. Wang, J. Zheng, B. Chen, and Q. Zhu, "The bicoastal IBM/SRI CTS STT system," in *Rich Transcription (RT-04F) Workshop*, 2004.

[14] D. Hillard and M. Ostendorf, "Scoring structural mde: towards more meaningful error rates," in *Proc. of EARS RT-04 Workshop*, 2004.


## Acknowledgments


The authors would like to thank CLSP at Johns Hopkins, for hosting the summer workshop where much of this work was done. Thanks also to Eugene Charniak, Dustin Hillard, Mark Johnson, Jeremy Kahn, Mari Ostendorf and Wen Wang for software used (and advice given) during the course of this project. This material is based upon work supported by DARPA under contract MDA972-02-C-0038 and the NSF under Grant No. 0121285. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or DARPA.