

Sentence Unit Detection without an Audio Signal

William Morgan*

1 Introduction and motivation

Sentence unit (SU) detection is the task of dividing a sequence of words into individual sentences. SU detection is a close relative of sentence boundary detection, which has been a topic of study in the computational linguistics community for over a decade. (Palmer and Hearst, 1994; Reynar and Ratnaparkhi, 1997)

SU detection is specific to the context of automatic speech recognition (ASR) systems, which typically produce an unstructured sequence of words from an audio signal, and must then recover latent structural features in the signal such as word case (“true-casing”) and sentence boundaries (“SU detection”) in order for the output to be ready for human consumption. Recent efforts by the DARPA EARS program (Office, 2003) to improve ASR quality has renewed interest in this problem.

Work on SU detection in modern ASR systems typically takes in to account the full set of features available from the audio signal. Features like prosody, voice quality, and even (in the case of multi-modal systems) visual cues like gestures have all been shown to be informative in deciding on sentence boundaries. (Liu et al., 2005; Stolcke et al., 2004)

In this study, we apply conditional random fields (CRFs) to examine the feasibility of detecting sentence boundaries directly from text, i.e. without the corresponding audio signal. These results act as a baseline, suggesting the true usefulness of features extracted from the audio and video for SU detection.

2 CRFs

Much work on sentence unit detection has used Hidden Markov Models (HMMs) as the underlying model. (Shriberg et al., 2000; Renals and Gottoh, 2000; Christensen et al., 2001; Kim and Woodland, 2001) Recent experiments with CRFs, however, have shown they can exhibit better performance on the SU detection task than HMMs or maximum entropy approaches. (Liu et al., 2005)

A CRF is an undirected graphical model of representing an event sequence E globally conditioned on an observation sequence O . CRFs are naturally applicable to many problems to which HMMs have traditionally been applied, but unlike HMMs, which maximize the joint distribution $P(E, O)$, CRFs directly maximize the posterior event probabilities $P(E|O)$.

The most likely sequence of events in a CRF is given by

$$E^* = \arg \max_E \frac{\exp(\sum_k \lambda_k G_k(E, O))}{Z_\lambda(O)}$$

where $G_k(E, O)$ are potential functions over the events and observations, and Z_λ a normalization term. In general the G_k can be any functions, but in many cases (including ours) it is computationally beneficial to restrict oneself to a first-order CRFs, as exemplified by Figure 1.

In this paper we use the Stanford NLP CRF implementation.

3 Data

Our data was drawn from the NIST RT-03 evaluation, LDC publication LDC2004T12. (NIST, 2003)

No collaborators or advisors.

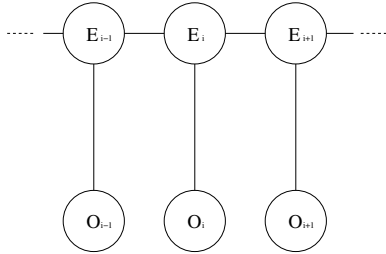


Figure 1: A first-order CRF. E represents the events (sentence boundary or not) and O represent the observations (the words of ASR output).

This corpus consists of 474 human-edited transcriptions, 172 drawn from broadcast news and 302 drawn from conversational speech, altogether comprising of 661k words. Of this, approximately 20% was set aside as test data. The remainder was used as training data.¹

4 Feature extraction and modeling

The CRF “event” for SU detection was encoded as a boolean value for each word in the training data, specifying whether that word was at the beginning of a sentence or not.

The transcript data contained, for each word of speech, the lexeme (written representation), start time, duration, and speaker identification. From these attributes we extracted the following features: the lexeme (lower-cased, to prevent confounding from case information), the duration, the delay since the prior word (if any), whether a speaker change had occurred.

Two additional sources of information were added. Part-of-speech tags for each word were determined by the Stanford NLP part-of-speech tagger, a bi-directional CMM tagger.

Additionally, a bigram language model was built from a portion of the the LDC English Gigaword corpus (LDC publication LDC2003T05). The language model was based on 211 million words of English text, and simply estimated, for each word w_i , the probability $P(w_i \text{ is the first word in a sentence} | w_{i+1})$.

¹Broadcast news and conversational speech have different characteristics; we mainly ignore this, but do try to control for it by drawing samples from the training corpus proportionately from both types.

5 Metrics

In evaluating a system for SU detection, it is likely that one is concerned both with type I and type II errors. We measured system performance using precision and recall, which capture both these types of errors. In our case, they are defined as

$$P = \frac{\# \text{ correctly marked as SU}}{\# \text{ marked as SU}}$$

and

$$R = \frac{\# \text{ correctly marked as SU}}{\# \text{ SUs in the corpus}}.$$

One pleasing feature of these two metrics is that while either may be gamed individually, it is impossible to game both simultaneously. In our case, precision may be gamed by tagging only the first word of the test set as a SU, and not tagging anything else, and recall may be gamed by tagging every word as an SU, but having high scores in both requires a tagger of both high accuracy and large coverage.

It is often convenient to combine these two scores into a single number. Typically this is done by the harmonic mean, or f-measure:

$$F = \frac{2PR}{P + R}.$$

Intuitively, as P approaches R , F approaches $\frac{P+R}{2}$, and as they diverge, F approaches 0.

6 Results

Table 1 gives the result of the experiments. Somewhat surprisingly, neither the part-of-speech tags (“+pos”) nor the language model (“+lm”) significantly affected the performance of the system as compared to the baseline. For comparison, three systems which randomly assign SU or not-SU tags to each word are given. It is clear that while the trained systems far exceed these in precision, the difference in recall scores is much lower.

Figure 2 shows the f-measure of the baseline system on both training and test sets as a function of training set size. Near-maximal scores are obtained with fairly small training set sizes (50 documents), suggesting that more work on feature extraction is needed to improve scores.

Figures 3 and 4 show the precision, recall and f-measure of the baseline system on the test and training sets respectively. Again, we see that by the time

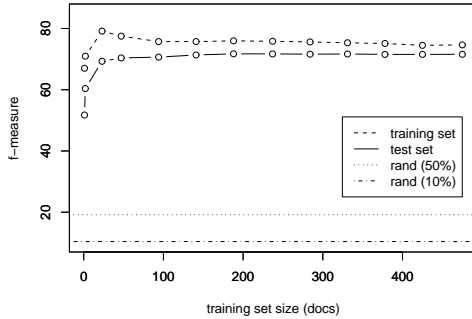


Figure 2: Training vs test set performance (f-measure) of the baseline system.

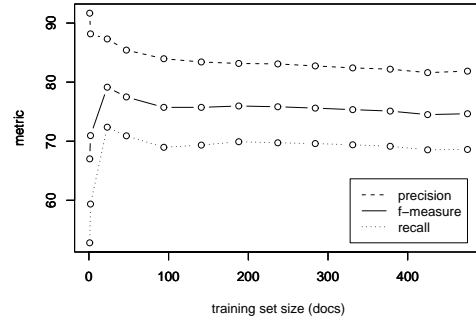


Figure 4: Precision, recall and f-measure of the baseline system on the training set.

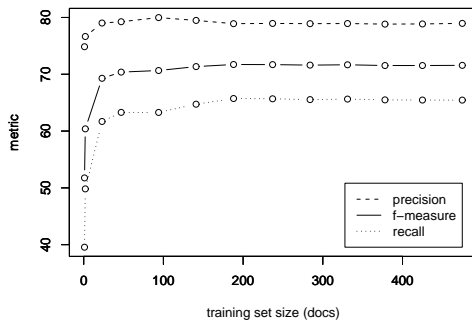


Figure 3: Precision, recall and f-measure of the baseline system on the test set.

the training corpus contains 50 documents, we have achieved the vast majority of our final performance. These graphs also highlight the fact that the system’s errors are primarily those of omission, not of incorrect labelling.

For comparison purposes, Liu et al. (2005) report a “boundary classification” error rate of 5.43% when training a CRF without using only textual features; i.e., the system correctly labelled 94.57% of the words as either SU or non-SU. Our error rate by the same metric is 5.67%, which is roughly comparable (they trained on the same corpus). They used several other textual features (such as automatically-derived word classes) that may account for the difference. When they added prosodic features, their error rate went down to 3.47%.

7 Conclusion and Future Work

Baseline system performance was a respectable f-measure of 71; specifically, approximately 80% of the labels the system made were correct, and it de-

System	Precision	Recall	F-measure
baseline	79.10	64.49	71.05
+pos	78.96	65.45	71.57
+lm	78.74	65.76	71.67
just 1 doc	76.65	49.82	60.39
random 50%	11.89	49.40	19.16
random 10%	11.87	9.80	10.74
random 1%	11.40	0.95	1.75

Table 1: Results of training the CRF to perform SU detection. The baseline system has neither part-of-speech tags nor language model probabilities. “Just 1 doc” is identical to the baseline system but is only trained on one document. The “random” systems simply assign a SU tag to each word by flipping a coin weighted with the respective probability.

tected about 65% of the possible SUs. However, the fact that test set performance reached most of its maximal value with only 50 documents suggests a need for better modeling and feature extraction. It is likely that features from the audio stream itself would improve these scores significantly.

It is somewhat surprising that neither the language model nor the part-of-speech tagger had any effect on the system performance. More work is needed to understand this. One approach would be to examine individual errors and compare the features and feature weights involved to see if the source of the errors can be pinpointed.

One interesting question still unanswered is the performance of the system on actual ASR output. The data used in this study was exclusively human-annotated and thus did not have the characteristic er-

rors of ASR output, which will likely have a negative effect on system performance.

References

- Heidi Christensen, Steve Renals, and Yoshihiko Gotoh. 2001. Punctuation annotation using statistical prosody models. In *ISCA Workshop on Prosody in Speech Recognition and Understanding*, September 10.
- Ji-hwan Kim and P. C. Woodland. 2001. The use of prosody in A combined system for punctuation generation and speech recognition. In *Proceedings of Eurospeech 2001*, pages 2757–2760, November 22.
- Yang Liu, Andreas Stolcke, Elizabeth Shriberg, and Mary Harper. 2005. Using conditional random fields for sentence boundary detection in speech. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 451–458, June 12.
- NIST. 2003. RT-03F workshop agenda and presentations. <http://www.nist.gov/speech/tests/rt/rt2003/fall/presentations/>, November.
- DARPA Information Processing Office. 2003. Effective, affordable, reusable speech-to-text (EARS). <http://www.darpa.mil/ipto/programs/ears/>.
- D. D. Palmer and M. A. Hearst. 1994. Adaptive sentence boundary disambiguation. In *Proceedings of the Fourth Applied Conference on NLP*, pages 78–83.
- Steve Renals and Yoshihiko Gotoh. 2000. Sentence boundary detection in broadcast speech transcripts. In *Proceedings of ISCA Workshop: ASR: Challenges for the New Millenium ASR-2000*, pages 228–235, July 19.
- J. Reynar and A. Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Applied Conference on NLP*, pages 16–19.
- Elizabeth Shriberg, Andreas Stolcke, Dilek Z. Hakkani-Tür, and Gökan Tür. 2000. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communications* 32, pages 127–154.
- Andreas Stolcke, Elizabeth Shriberg, Mary Harper, and Yang Liu. 2004. Comparing and combining generative and posterior probability models: Some advances in sentence boundary detection in speech. In *Proceedings of the Conference on Empirical Methods in NLP*, June 12.