SENTENCE BOUNDARY DETECTION ON BROADCAST NEWS

Dustin Hillard
University of Washington, EE
hillard@ee.washington.edu

Abstract

Sentence boundary detection is an important task that augments automatic speech recognition word output with syntactic structure to allow for ease in reading and facilitate natural language processing tasks. This paper's approach recovers setence boundaries based on a combination of prosodic and word-based features. Sentence boundaries are modeled as hidden events, predicted by language models supplemented by prosodic likelihoods computed from a decision tree. A typical language model is utilized and a part of speech class based model is integrated as well. The methods are then tested on the Broadacst News corpus.

1 Introduction

While most automatic speech recognition (ASR) systems output a string of words, many natural language processing tasks require a sentence structure. This stucture is hidden in normal ASR output, but it is important for further analysis such as topic detection, summarization, and more. The effectiveness of other tasks can depend on first attaining an accurate segmentation of speech into sentences. The quality of a language model was shown to improve when based on sentence segmentation, rather than acoustic segmentation, in (Meteer and Iyer, 1996).

While a text based approach for these types of tasks is common and productive (J. Yamron et al., 1998), prosodic features can also provide strong predictive features. Prosodic features include information about speaker pitch (F0), speaker timing and tempo (duration), and pauses. In typical text based analysis these features are not accounted for. Incorporating these features in the model has been shown to increase detection when combined with lexical features, and perform comparably by themselves (Shriberg et al., 2000). This paper expands on that work by training a part of speech language model, and integrating it into the detection approach.

2 Approach

Features from two different sources are extracted for use. Word based features are computed with language modeling, while prosodic features are evaluated with decision trees.

2.1 Language Modeling

The intention of the language modeling is to encode information about sentence boundaries. Language models are trained in a standard way, but with a special token included at sentence boundaries. The approach is to model the boundaries as hidden events in a hidden Markov model (HMM). The states of the HMM consist of whether or not the current word is a boundary, plus preceding preceding words (or boundary tokens) up to the length of the N-gram, where $N = 4$ in these experiments. The transition probabilties of the HMM are N-gram probabilitites estimated from training, where the HMM is trained in a supervised fashion (on hand labeled data), with Kneser-Ney backoff (Kneser and Ney, 1995).

The boundary classification is denoted here by $B = B, ..., B_K$ and the words by $W = W, ..., W_N$, where the hidden varible is B_i. Evaluation of sentences is based on the number of boundaries marked correctly, so the forward-backward algorithm maximizes the posterior probability at each boundary B_i (L. Baum et al., 1970; Dermatas and Kokkinakis, 1995).

$$\operatorname*{argmax}_{B_i} P(B_i \mid W) \tag{1}$$

In addition, to attempt to provide another source of information, a part of speech class language model is built. The training data set is tagged with a maximum entropy part of speech tagger (Ratnaparkhi, 1996). A simple mapping of words to classes (only one class per word) is used to reduce the search space of detection.

Finally, the two language models can be combined with a language model mixture or by interpolation. The language model mixture is a linear combination, where each token w is a word or a boundary token (boundaries occur only in the training set), and c_i is a class:

$$P(W) = \prod_i \Big[\lambda p(w_i \mid w_{i-1}, w_{i-2}, w_{i-3}) + (1-\lambda) p(w_i \mid c_i) p(c_i \mid c_{i-1}, c_{i-2}, c_{i-3})\Big] \quad (2)$$

Alternatively, the models may be combined in a log linear fashion (multiplicative), as represented below in equation (3).

$$P(W) = \prod_i \Big[\lambda \log(p(w_i \mid w_{i-1}, w_{i-2}, w_{i-3})) + (1-\lambda) \log(p(w_i \mid c_i) p(c_i \mid c_{i-1}, c_{i-2}, c_{i-3}))\Big] \quad (3)$$

2.2 Prosodic Features

Prosodic features for each inter-word boundary are related to the word before and after a boundary. While additional features from larger regions could be used, the smaller span is used to allow for simplicity and computational feasability.

Prosodic features extracted were pause durations, phone durations, and F0. The pause features are at inter-word boundaries, while duration and F0 are extracted from the two adjacent words. Most features focus on the word previous to a possible boundary because previous work found these words to carry more prosodic information (Shriberg et al., 1997). Also included are pitch features reflecting the difference across the boundary. Amplitude or energy-based features are not employed because previous work has shown that they are largely redundant, and less reliable due to large channel variability in the corpus. Features are the same set as in (Shriberg et al., 2000).

Classification with prosodic features is accomplished using CART-style decision trees (L. Breiman et al., 1984) implemented with the IND package. IND handles missing features, which is helpful for F0 features that are sometimes undefined. In the case of missing features, a test sample is sent down each branch with the proportion that samples in the training set at that node were split, then the two results are averaged. The tree predicts a class trained on prosodic features F_i at a possible boundary B_i. The tree prediction is also weakly conditioned on words, because the phone alignment is based on the word models. The probability is then:

$$P(B_i \mid F_i, W) \quad (4)$$

In order to facilitate learning of sentences boundaries, which occur at about $1/10$ the rate of non-boundaries, the data was downsampled so that there were the same number of decision tree training points per class. The decision tree size was determined using error-based cost-complexity pruning with 4-fold cross validation. Cost-complexity measures the resubstitution error of a tree, further penalized by the size of the tree. To reduce the initial candidate feature set to a smaller, optimal set, an iterative feature selection algorithm that involved running multiple decision trees was used (Shriberg et al., 2000). The algorithm combines elements of brute-force search (in a leave-one-out paradigm) with previously determined heuristics for narrowing the search space. Entropy reduction of the overall tree after cross-validation was used as a criterion for selecting the best subtree.

2.3 Model combination

Prosodic and lexical features provide different sources of information, so the combination should yeild increased accuracy. To combine the models, independance is assumed (although some gain might be had from capturing their statistical interaction, past work has shown this simpler approach is competitive in practice).

To combine the two feature types, the HMM model from the lexical detection can be extended to utilize word and prosodic features. The new model (which is not exactly an HMM) is a joint distribuition $P(W, F, B)$ of words W, prosodic features F, and boundaries B. In this framework prosodic features are included by making the assumption that each prosodic observation F_i is conditionally independant of all others, given the boundary types B_i and words W. So that a complete path through the model is the total probability:

$$P(W, F, B) = P(W, B) \prod_i P(F_i \mid B_i, W) \quad (5)$$

After making these assumptions, the prosodic liklihood $P(F_i \mid B_i, W)$ needs to be computed. Using Bayes' rule and the posteriors available from the decision tree $P_{DT}(B_i \mid F_i, W)$, the likihood is obtained:

$$P(F_i \mid B_i, W) = \frac{P(F_i \mid W) P_{DT}(B_i \mid F_i, W)}{P(B_i \mid W)} \quad (6)$$

Further simplification is possible given that $P(F_i \mid W)$ terms are constant for all terms, so they can be ignored. Further, $P(B_i \mid W) \approx P(B_i)$ because the features are weakly conditioned on the words (only in the sense of time alignment). Finally, because the tree was downsampled to have unifrom priors, the $P(B_i)$ term is equal to $\frac{1}{2}$ for both boundary types, so it can be ignored. Lastly, the prosodic model is given a weight term MCW, so that its effect on the lexical model can be adjusted and empircally optimized, with $P_{DT}(B_i \mid F_i, W)^{MCW}$.

Finally, sentence boundaries probabilities can maximized using the forward-backward algorithm, which optimizes for this paper's error metrics (as mentioned for the lexical case).

$$\underset{B_i}{\text{argmax}}\ P(B_i \mid W, F) \tag{7}$$

3 Results and Discussion

The training set consisted of 93 broadcasts with about 700K words, while testing was performed on a heldout set of 5 broadcasts, about 25K words. An analysis of prosodic features used, followed by an assesment of N-gram length, gives an understanding of how the features contribute.

The prosodic feature usage is given in Table 1 below. Pause feaures provide the strongest cues to sentence boundaries, while F0 and duration are also helpful.

Prosodic Feature	Frequency Used in Tree
Pause duration	66%
Phone duration	24%
F0 boundary change	10%

Table 1: Prosodic feature use

Detection results with prosodic features are shown in Table 2 for the downsampled data, with chance at 50%. Accuracy is much greater than chance, and good reductions in perplexity are obtained.

Detection	Perplexity by class	
Accuracy	Sentence	No Sent.
89.61%	2 ⇝ 1.4	2 ⇝ 1.37

Table 2: Prosodic feature use

Language modeling features are compared by evaluating different length N-grams. Chance for this detection is 93.8%, when all boundaries are left as non-sentence. Table 3 below shows the 4-gram outperforms the tri-gram in detection, and exhibits a greater reduction in perplexity (language complexity).

	Detection	Perplexity by class	
N-gram	Accuracy	Sentence	No Sent.
4-gram	95.85%	16.1⇝ 3.18	1.07⇝ 1.05
3-gram	95.79%	16.1⇝ 3.38	1.07⇝ 1.04

Table 3: Language Model Results

Results on the part of speech language model in Table 4 below show that there was little gain on chance by using the class language model as compared to its typical 4-gram. While accuracy is better than chance for the downsampled data set which has an even distribution over boundary types, the perplexity increases significantly.

Accuracy on Dist.		Perplexity by class	
Actual	Even	Sentence	No Sent.
90.8%	54.9%	16.1⇝ 81	1.07⇝ 1.16

Table 4: Language Model Results

The better than chance performance by the part of speech model on an evenly distributed test set shows that some sentence structure information is encoded in the class language model. Although detection is only somewhat better than chance, an attempt to incorporate it with the main model is reasonable to assess whether the class system will provide a separate source of knowledge. Unfortunately neither of the combination methods discussed earlier (a linear combination mixture model, or a log-linear interpolation) produced an accuracy greater than the standard 4-gram by its self.

Finally, results for model combination are given in Table 5 below. The baseline for this experiment is the 95.85% accuracy and 3.18 sentence class perplexity obtained with the 4-gram language model on its own.

	Detection	Perplexity by class	
MCW weight	Accuracy	Sentence	No Sent.
0.5	96.34%	16.1⇝ 2.31	1.07⇝ 1.04
0.6	96.38%	16.1⇝ 2.21	1.07⇝ 1.05
0.7	96.38%	16.1⇝ 2.13	1.07⇝ 1.05
0.8	96.39%	16.1⇝ 2.06	1.07⇝ 1.05
0.9	96.28%	16.1⇝ 2.00	1.07⇝ 1.06
1.0	96.20%	16.1⇝ 1.95	1.07⇝ 1.06
1.1	96.13%	16.1⇝ 1.91	1.07⇝ 1.07
1.2	96.05%	16.1⇝ 1.89	1.07⇝ 1.07
1.3	95.96%	16.1⇝ 1.86	1.07⇝ 1.08
1.4	95.83%	16.1⇝ 1.85	1.07⇝ 1.08

Table 5: Mixture Model Results

Results show a significant gain in detection accuracy, as well as perplexity reduction. The optimal mixture wieght for accuracy is 0.8, but increasing the weight further produces additional perplexity reduction at sentence boundaries. At the same time though, non-sentence boundary perplexity increases, so this causes overall accuracy to fall because the non-boundary cases are more frequent by roughly a factor of ten (so even the small rise in non-boundary perplexity overwhlems the consistent downward trend in the sentence class perplexity). Increasing wieghts for the prosodic model begins to bias the classification towards the uniform priors condition that the prosodic features are trained on, so there is over-detection of boundaries.

4 Conclusion

Both prosodic and lexical features are to shown to have stong predicting capabilities for sentence boundary detection. Each feature set provides reductions in perplexity, and the sets provide complimentary information sources because model combination provides further gains. Although a part of speech class language model shows some signs of ability to predict boundaries, perplexity is very high and hurts the other models when combined. A more sophisticated approach to building the part of speech language model may helpful. Using a factored language model such as in (K. Kirchhoff et al., 2003) could be a helpful approach to integrating the part of speech information with the typical word N-gram. Additional further work could integrate word confidence with the language model so that words from automatic speech recognition will be considered according to the confidence of the recognizer when the word is output.

References

E. Dermatas and G. Kokkinakis. 1995. Automatic stochastic tagging of natural language texts. *Computational LinguisticsS*, 21(2):137–164.

J. Yamron et al. 1998. A hidden markov model approach to text segmentation and event tracking. In *Proc. ICSLP*, pages 333–336.

K. Kirchhoff et al. 2003. Novel approaches to arabic speech recognition: Report from the 2002 johns-hopkins summer worksho. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Hong Kong, China.

R. Kneser and H. Ney. 1995. Improved backing-off for n-gram language modeling. In *Proc. IEEE ICAASP-95*, pages 181–184.

L. Baum et al. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Statist.*, 41(1):164–171.

L. Breiman et al. 1984. *Classification And Regression Trees*. Wadsworth International Group, Belmont, CA.

M. Meteer and R. Iyer. 1996. Modeling conversational speech for speech recognition. In *Proceedings of the Conference on Emphirical Methods in Natural Language Processing*.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Proce ssing Conference*, pages 133–141.

Elizabeth Shriberg, Rebecca Bates, and Andreas Stolcke. 1997. A prosody-only decision-tree model for disfluency detection. In *Proc. Eurospeech '97*, pages 2383–2386, Rhodes, Greece.

E. Shriberg et al. 2000. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32(1-2):127–154, September.