A Maximum Entropy Approach to Identifying Sentence Boundaries

Jeffrey C. Reynar and Adwait Ratnaparkhi*
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, Pennsylvania, USA
{jcreynar, adwait}@unagi.cis.upenn.edu

Abstract

We present a trainable model for identifying sentence boundaries in raw text. Given a corpus annotated with sentence boundaries, our model learns to classify each occurrence of *.*, *?*, and *!* as either a valid or invalid sentence boundary. The training procedure requires no hand-crafted rules, lexica, part-of-speech tags, or domain-specific information. The model can therefore be trained easily on any genre of English, and should be trainable on any other Roman-alphabet language. Performance is comparable to or better than the performance of similar systems, but we emphasize the simplicity of retraining for new domains.

1 Introduction

The task of identifying sentence boundaries in text has not received as much attention as it deserves. Many freely available natural language processing tools require their input to be divided into sentences, but make no mention of how to accomplish this (e.g. (Brill, 1994; Collins, 1996)). Others perform the division implicitly without discussing performance (e.g. (Cutting et al., 1992)).

On first glance, it may appear that using a short list of sentence-final punctuation marks, such as *.*, *?*, and *!*, is sufficient. However, these punctuation marks are not used exclusively to mark sentence breaks. For example, embedded quotations may contain any of the sentence-ending punctuation marks and *.* is used as a decimal point, in e-mail addresses, to indicate ellipsis and in abbreviations. Both *!* and *?* are somewhat less ambiguous but appear in proper names and may be used multiple times for emphasis to mark a single sentence boundary.

Lexically-based rules could be written and exception lists used to disambiguate the difficult cases described above. However, the lists will never be exhaustive, and multiple rules may interact badly since punctuation marks exhibit absorption properties. Sites which logically should be marked with multiple punctuation marks will often only have one ((Nunberg, 1990) as summarized in (White, 1995)). For example, a sentence-ending abbreviation will most likely not be followed by an additional period if the abbreviation already contains one (e.g. note that *D.C* is followed by only a single *.* in *The president lives in Washington, D.C.*).

As a result, we believe that manually writing rules is not a good approach. Instead, we present a solution based on a maximum entropy model which requires a few hints about what information to use and a corpus annotated with sentence boundaries. The model trains easily and performs comparably to systems that require vastly more information. Training on 39441 sentences takes 18 minutes on a Sun Ultra Sparc and disambiguating the boundaries in a single *Wall Street Journal* article requires only 1.4 seconds.

2 Previous Work

To our knowledge, there have been few papers about identifying sentence boundaries. The most recent work will be described in (Palmer and Hearst, To appear). There is also a less detailed description of Palmer and Hearst's system, SATZ, in (Palmer and Hearst, 1994).[1] The SATZ architecture uses either a decision tree or a neural network to disambiguate sentence boundaries. The neural network achieves 98.5% accuracy on a corpus of *Wall Street Journal*

*The authors would like to acknowledge the support of ARPA grant N66001-94-C-6043, ARO grant DAAH04-94-G-0426 and NSF grant SBR89-20230.

[1] We recommend these articles for a more comprehensive review of sentence-boundary identification work than we will be able to provide here.

articles using a lexicon which includes part-of-speech (POS) tag information. By increasing the quantity of training data and decreasing the size of their test corpus, Palmer and Hearst achieved performance of 98.9% with the neural network. They obtained similar results using the decision tree. All the results we will present for our algorithms are on their initial, larger test corpus.

In (Riley, 1989), Riley describes a decision-tree based approach to the problem. His performance on the Brown corpus is 99.8%, using a model learned from a corpus of 25 million words. Liberman and Church suggest in (Liberman and Church, 1992) that a system could be quickly built to divide newswire text into sentences with a nearly negligible error rate, but do not actually build such a system.

3 Our Approach

We present two systems for identifying sentence boundaries. One is targeted at high performance and uses some knowledge about the structure of English financial newspaper text which may not be applicable to text from other genres or in other languages. The other system uses no domain-specific knowledge and is aimed at being portable across English text genres and Roman alphabet languages.

Potential sentence boundaries are identified by scanning the text for sequences of characters separated by whitespace (tokens) containing one of the symbols *!*, *.* or *?*. We use information about the token containing the potential sentence boundary, as well as contextual information about the tokens immediately to the left and to the right. We also conducted tests using wider contexts, but performance did not improve.

We call the token containing the symbol which marks a putative sentence boundary the Candidate. The portion of the Candidate preceding the potential sentence boundary is called the Prefix and the portion following it is called the Suffix. The system that focused on maximizing performance used the following hints, or contextual "templates":

- The Prefix
- The Suffix
- The presence of particular characters in the Prefix or Suffix
- Whether the Candidate is an honorific (e.g. *Ms., Dr., Gen.*)
- Whether the Candidate is a corporate designator (e.g. *Corp., S.p.A., L.L.C.*)
- Features of the word left of the Candidate
- Features of the word right of the Candidate

The templates specify only the form of the information. The *exact* information used by the maximum entropy model for the potential sentence boundary marked by . in *Corp.* in Example 1 would be: PreviousWordIsCapitalized, Prefix=*Corp*, Suffix=NULL, PrefixFeature=CorporateDesignator.

(1) ANLP Corp. chairman Dr. Smith resigned.

The highly portable system uses only the identity of the Candidate and its neighboring words, and a list of abbreviations induced from the training data.[2] Specifically, the "templates" used are:

- The Prefix
- The Suffix
- Whether the Prefix or Suffix is on the list of induced abbreviations
- The word left of the Candidate
- The word right of the Candidate
- Whether the word to the left or right of the Candidate is on the list of induced abbreviations

The information this model would use for Example 1 would be: PreviousWord=*ANLP*, FollowingWord=*chairman*, Prefix=*Corp*, Suffix=NULL, PrefixFeature=InducedAbbreviation.

The abbreviation list is automatically produced from the training data, and the contextual questions are also automatically generated by scanning the training data with question templates. As a result, no hand-crafted rules or lists are required by the highly portable system and it can be easily retrained for other languages or text genres.

4 Maximum Entropy

The model used here for sentence-boundary detection is based on the maximum entropy model used for POS tagging in (Ratnaparkhi, 1996). For each potential sentence boundary token (*.*, *?*, and *!*), we estimate a joint probability distribution p of the token and its surrounding context, both of which are denoted by c, occurring as an actual sentence boundary. The distribution is given by: $p(b, c) = \pi \prod_{j=1}^{k} \alpha_j^{f_j(b,c)}$, where $b \in \{\texttt{no}, \texttt{yes}\}$, where

[2] A token in the training data is considered an abbreviation if it is preceded and followed by whitespace, and it contains a *.* that is *not* a sentence boundary.

the α_j's are the unknown parameters of the model, and where each α_j corresponds to a f_j, or a *feature*. Thus the probability of seeing an actual sentence boundary in the context c is given by $p(\texttt{yes}, c)$.

The contextual information deemed useful for sentence-boundary detection, which we described earlier, must be encoded using features. For example, a useful feature might be:

$$f_j(b,c) = \begin{cases} 1 & \text{if Prefix}(c) = Mr \ \& \ b = \texttt{no} \\ 0 & \text{otherwise} \end{cases}$$

This feature will allow the model to discover that the period at the end of the word *Mr.* seldom occurs as a sentence boundary. Therefore the parameter corresponding to this feature will hopefully boost the probability $p(\texttt{no}, c)$ if the Prefix is *Mr*. The parameters are chosen to maximize the likelihood of the training data using the *Generalized Iterative Scaling* (Darroch and Ratcliff, 1972) algorithm.

The model also can be viewed under the Maximum Entropy framework, in which we choose a distribution p that maximizes the entropy $H(p)$

$$H(p) = -\sum p(b,c) \log p(b,c)$$

under the following constraints:

$$\sum p(b,c) f_j(b,c) = \sum \tilde{p}(b,c) f_j(b,c), 1 \leq j \leq k$$

where $\tilde{p}(b,c)$ is the observed distribution of sentence-boundaries and contexts in the training data. As a result, the model in practice tends not to commit towards a particular outcome (`yes` or `no`) unless it has seen sufficient evidence for that outcome; it is maximally uncertain beyond meeting the evidence.

All experiments use a simple decision rule to classify each potential sentence boundary: a potential sentence boundary is an actual sentence boundary if and only if $p(\texttt{yes}|c) > .5$, where

$$p(\texttt{yes}|c) = \frac{p(\texttt{yes}, c)}{p(\texttt{yes}, c) + p(\texttt{no}, c)}$$

and where c is the context including the potential sentence boundary.

5 System Performance

We trained our system on 39441 sentences (898737 words) of *Wall Street Journal* text from sections 00 through 24 of the second release of the Penn Treebank[3] (Marcus, Santorini, and Marcinkiewicz, 1993). We corrected punctuation mistakes and erroneous sentence boundaries in the training data. Performance figures for our best performing system, which used a hand-crafted list of honorifics and corporate designators, are shown in Table 1. The first test set, WSJ, is Palmer and Hearst's initial test data and the second is the entire Brown corpus. We present the Brown corpus performance to show the importance of training on the genre of text on which testing will be performed. Table 1 also shows the number of sentences in each corpus, the number of candidate punctuation marks, the accuracy over potential sentence boundaries, the number of false positives and the number of false negatives. Performance on the WSJ corpus was, as we expected, higher than performance on the Brown corpus since we trained the model on financial newspaper text.

[3] We did not train on files which overlapped with Palmer and Hearst's test data, namely sections 03, 04, 05 and 06.

	WSJ	Brown
Sentences	20478	51672
Candidate P. Marks	32173	61282
Accuracy	98.8%	97.9%
False Positives	201	750
False Negatives	171	506

Table 1: Our best performance on two corpora.

Possibly more significant than the system's performance is its portability to new domains and languages. A trimmed down system which used no information except that derived from the training corpus performs nearly as well, and requires no resources other than a training corpus. Its performance on the same two corpora is shown in Table 2.

Test Corpus	Accuracy	False Positives	False Negatives
WSJ	98.0%	396	245
Brown	97.5%	1260	265

Table 2: Performance on the same two corpora using the highly portable system.

Since 39441 training sentences is considerably more than might exist in a new domain or a language other than English, we experimented with the quantity of training data required to maintain performance. Table 3 shows performance on the WSJ corpus as a function of training set size using the best performing system and the more portable system. As can seen from the table, performance degrades as the quantity of training data decreases, but even

	Number of sentences in training corpus						
	500	1000	2000	4000	8000	16000	39441
Best performing	97.6%	98.4%	98.0%	98.4%	98.3%	98.3%	98.8%
Highly portable	96.5%	97.3%	97.3%	97.6%	97.6%	97.8%	98.0%

Table 3: Performance on *Wall Street Journal* test data as a function of training set size for both systems.

with only 500 example sentences performance is better than the baselines of 64.0% if a sentence boundary is guessed at every potential site and 78.4% if only token-final instances of sentence-ending punctuation are assumed to be boundaries.

6 Conclusions

We have described an approach to identifying sentence boundaries which performs comparably to other state-of-the-art systems that require vastly more resources. For example, Riley's performance on the Brown corpus is higher than ours, but his system is trained on the Brown corpus and uses thirty times as much data as our system. Also, Palmer & Hearst's system requires POS tag information, which limits its use to those genres or languages for which there are either POS tag lexica or POS tag annotated corpora that could be used to train automatic taggers. In comparison, our system does not require POS tags or any supporting resources beyond the sentence-boundary annotated corpus. It is therefore easy and inexpensive to retrain this system for different genres of text in English and text in other Roman-alphabet languages. Furthermore, we showed that a small training corpus is sufficient for good performance, and we estimate that annotating enough data to achieve good performance would require only several hours of work, in comparison to the many hours required to generate POS tag and lexical probabilities.

7 Acknowledgments

We would like to thank David Palmer for giving us the test data he and Marti Hearst used for their sentence detection experiments. We would also like to thank the anonymous reviewers for their helpful insights.

References

[Brill1994] Brill, Eric. 1994. Some advances in transformation-based part-of-speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, volume 1, pages 722–727.

[Collins1996] Collins, Michael. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, June.

[Cutting et al.1992] Cutting, Doug, Julian Kupiec, Jan Pedersen, and Penelope Sibun. 1992. A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 133–140, Trento, Italy, April.

[Darroch and Ratcliff1972] Darroch, J. N. and D. Ratcliff. 1972. Generalized Iterative Scaling for Log-Linear Models. *The Annals of Mathematical Statistics*, 43(5):1470–1480.

[Liberman and Church1992] Liberman, Mark Y. and Kenneth W. Church. 1992. Text analysis and word pronunciation in text-to-speech synthesis. In Sadaoki Furui and M. Mohan Sondi, editors, *Advances in Speech Signal Processing*. Marcel Dekker, Incorporated, New York.

[Marcus, Santorini, and Marcinkiewicz1993] Marcus, Mitchell, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

[Nunberg1990] Nunberg, Geoffrey. 1990. *The Linguistics of Punctuation*. Number 18 in CSLI Lecture Notes. University of Chicago Press.

[Palmer and Hearst1994] Palmer, David D. and Marti A. Hearst. 1994. Adaptive sentence boundary disambiguation. In *Proceedings of the 1994 conference on Applied Natural Language Processing (ANLP)*, Stuttgart, Germany, October.

[Palmer and HearstTo appear] Palmer, David D. and Marti A. Hearst. To appear. Adaptive multilingual sentence boundary disambiguation. *Computational Linguistics*.

[Ratnaparkhi1996] Ratnaparkhi, Adwait. 1996. A maximum entropy model for part-of-speech tagging. In *Conference on Empirical Methods in Natural Language Processing*, pages 133–142, University of Pennsylvania, May 17–18.

[Riley1989] Riley, Michael D. 1989. Some applications of tree-based modelling to speech and language. In *DARPA Speech and Language Technology Workshop*, pages 339–352, Cape Cod, Massachusetts.

[White1995] White, Michael. 1995. Presenting punctuation. In *Proceedings of the Fifth European Workshop on Natural Language Generation*, pages 107–125, Leiden, The Netherlands.