Sentence Boundary Detection Using a MaxEnt Classifier

Neha Agarwal, Kelley Herndon Ford, Max Shneider
Department of Computer Science
Stanford University, Stanford, CA
nagarwal@stanford.edu, swift@stanford.edu, maxs@stanford.edu

Abstract

For the sentence boundary detection task, we applied a Maximum Entropy (MaxEnt) classifier using several different features based on the local context. We compared the performance of training our classifier on the Wall Street Journal (WSJ) corpus and testing on three data sets: the WSJ and Brown Penn Treebank corpora and the GENIA corpus. Our results indicate that our features work very well on the WSJ corpus, achieving a precision of 99.5%, a recall of 97.5%, and an F1 score of 98.5%. Moreover, the system was quite robust to testing on the Brown corpus, but less robust to testing on the GENIA corpus. Although we also tested our classifier by adding part-of-speech (POS) tags to the best feature set, the improvement was negligible.

1. Motivation

Sentence boundary detection is an important initial processing step for many applications, such as part-of-speech tagging. You can get fairly accurate results when doing simple boundary detection with '.', '?', and '!' characters. However, consider the following examples, which were taken from the WSJ test corpus:

> Rolls-Royce Motor Cars Inc. said it expects its U.S. sales to remain steady at about 1,200 cars in 1990.
>
> ``So what if you miss 50 tanks somewhere?" asks Rep. Norman Dicks (D., Wash.), a member of the House group that visited the talks in Vienna.
>
> Later, he recalls the words of his Marxist mentor: ``The people! Theft! The holy fire!"
>
> (Mr. Paul says it wasn't that high.)
>
> About 40 Italian businesses, including Fiat S.p.A. and Ing. C. Olivetti & Co., have formed a consortium to lobby for holding the expo in Venice.

Obviously, we need more complex logic to handle special cases, such as abbreviations, quotations within sentences, and parentheses. Our MaxEnt classifier is designed to improve the performance on these special cases. Other approaches to the sentence boundary detection task include rule-based systems, decision trees, neural networks, and Hidden Markov Models [1].

2. Methodology

The methodology behind our project is divided into three sections: the MaxEnt classifier, the feature extraction, and the data sets.

2.1. MaxEnt Classifier

We used a MaxEnt classifier to predict sentence boundaries. We assume that the reader is familiar with the MaxEnt algorithm, since its details require a more thorough explanation than we are able to provide here. However, as a high-level summary, MaxEnt produces a probability distribution over the possible labels based on features extracted from local contexts in the training data. These probabilities are computed using an objective function and its derivatives, which are smoothed to prevent overfitting. The test labels are then predicted using the features extracted from their contexts and the corresponding probability distribution over the labels.

The data sets, which will be described in section 2.3, exist in a variety of formats. However, they all have three common characteristics: a list of sentences, the words in each sentence, and their corresponding POS tags. In boundary prediction, we need features that straddle the sentence boundaries. So instead of feeding the list of sentences into the MaxEnt classifier directly, we concatenated them together when reading in the training, validation, and test data, resulting in one long sentence for each. In order to mark true sentence boundaries, we assign each word an end-of-sentence (EOS) label (simply Y or N) in addition to its tag. ***Table 1*** shows an example of the types of sentences that are produced.

Label	Y	N	N	N	N	Y	N
Tag	.	NNS	VBD	RB	VBD	.	DT
Word	.	Terms	were	n't	disclosed	.	The

Table 1*:* An Example Tagged and Labeled Sentence For Our System.

Next, we simply extract features from the training sentence to construct our MaxEnt probability

distributions based on the local context, and then predict labels for the words in the validation or test sentence.

2.2. Feature Extraction

We trained our MaxEnt classifier to determine sentence boundaries by extracting features from each of the trigram contexts that occurred within the WSJ training sentence. These contexts consisted of the current word (*Current*) and its tag, the words that occurred before and after it (*Previous* and *Next,* respectively) and their tags, and a label that indicated whether or not *Current* was a sentence boundary, as discussed in section 2.1.

After training, we tested the classifier by predicting the label of each trigram context in the WSJ validation sentence based on its features. We then compared our predicted labels with the true (gold) labels, as identified in the corpus (the comparison techniques will be discussed in section 3). To enable us to target specific errors, we printed *Previous*, *Current*, *Next*, the predicted label, and the gold label of each context that was guessed incorrectly.

To determine the best feature set, we performed numerous runs on the WSJ validation data using different combinations of features. ***Table 2*** describes the feature set that yielded the best results, along with example trigram contexts that each feature was designed to target. For example, an effective feature to deal with quotation marks as EOS is listed in the final row: if the previous word is a period, question mark, or exclamation point, the current word is a single or double right quote (´ or ”), and the next word is either a double left quote (“) or begins with an uppercase, then this feature is added. Some of the table entries were actually separate features, but are combined in the table for brevity.

Features	Precision
Previous/*Next* is uppercase	", 1989 The"
Previous is all uppercase	"MONEY : 9"
Previous/*Next* length	"Anita Davis Of"
Current is “:”,”--”,”…”	"Report : Dow"
	"buy ... I"
Next is “$”	"Corp. -- $"
Next is all digits	"DEPOSIT : 8.15"
Previous is an abbreviation	"S.p . A."
	"Ltd. -- Four"
Current is “.”,”?”, or “!” and *Next* is “--” or double left quote (“)	"interview . ``"
	"in ? ``"
	"stocks . ''"
Current is “.”,”?”, or “!” and *Next* is not double left quote	"kidding ! When"
	". '' ``"
Previous is “.”,”?”, or “!”, *Current* is single or double right quote (´ or ”), and *Next* is double left quote (“) or is uppercase	". " Besides"
	". ' It"

Table 2*:* The Features Included in Our Best Feature Set.

Some additional features that we tested are shown in ***Table 3***. For example, we tried adding a feature if the current word is uppercase or if the previous word is a period, question mark or exclamation point. Although these seemed like good features to include, they did not improve the performance of the MaxEnt classifier on the WSJ validation data.

Features
Current is uppercase
Previous is “.”, “?” or “!”
Current is “.”, “?”, “!”, “!!”, “??”, “-RRB-“, single quote, double quote, or double right quote
Next is “-LRB-”, single quote, double quote, or double right quote

Table 3*:* Additional Features That Were Tested But Not Included in Our Best Feature Set.

2.3. Data Sets

We examined the following data sets: WSJ text from the Penn Treebank, the Brown corpus from the Penn Treebank, and the POS-labeled GENIA corpus [2] (described in detail below). Although the code read in hand-labeled parse trees from the Penn Treebank, only the POS tags were used; phrasal information was ignored. The relative data set sizes are shown in ***Table 4***.

Corpus	Total Sentences	Training Sentences	Validation Sentences	Test Sentences
WSJ	43,948	39,832	1,700	2,416
Brown	24,243	22,778	784	681
GENIA	20,544	18,146	1,199	1,199

Table 4*:* The Number of Sentences in Our Data Sets.

Since the Brown corpus was already in the Penn Treebank labeled format, no additional processing was required. The GENIA corpus, however, consisted of hand-labeled words for each of the sentences in a single file rather than parse trees. Because of this, we first split the data set into separate files for training, validation and test data sets.

The GENIA corpus contains annotated data for abstracts from the MEDLINE database (created by the National Library of Medicine) and is therefore substantially different from the WSJ and Brown corpora. It is labeled in the following manner, where the word is separated from its POS tag using a “/”:

to/TO
involve/VB
protein/NN
tyrosine/NN
kinase/NN
activity/NN

However, "/" was also used in some sentences, such as:

sites/cell/NNS
leukemia/B/NN
and/or/CC
monocytes/macrophages/NNS
+/-/CC

This caused quite a bit of ambiguity, which we dealt with by simply ignoring the text between the first and last "/". For example, "+/-/CC" was interpreted as "+/CC".

3. Evaluation

The MaxEnt classifier was evaluated by training on the WSJ corpus and testing on each of the data sets. Additionally, we trained the classifier on the particular corpus for comparison. Finally, POS tags were added to our best feature set, and the data sets were re-evaluated.

We evaluated the classifier using metrics of precision, recall, and F1 score. Precision is defined as the number of correct EOS labels divided by the total number of guessed EOS labels, whereas recall is the number of correct EOS labels divided by the total number of gold EOS labels. The F1 score is calculated as two divided by the sum of the inverse of precision and the inverse of recall.

3.1. The Best Feature Set on WSJ Validation Data

The best feature set, described in section 2.2, was determined by training on the WSJ corpus and testing on the WSJ validation data. The results are shown in ***Table 5***.

	Precision	Recall	F1 Score
WSJ	99.5%	97.5%	98.5%

Table 5: Training the Best Feature Set on WSJ and Testing on WSJ Validation Data.

3.2. Training on WSJ Corpus and Testing on the Respective Corpora

Having attained the best feature set, we tested the classifier on the WSJ test data. To determine the robustness of our classifier, we also tested it on the Brown and GENIA test data. ***Table 6*** and ***Figure 1*** present these results.

Corpus	Precision	Recall	F1 Score
WSJ	99.2%	96.1%	97.6%
Brown	96.8%	98.8%	97.8%
GENIA	98.5%	87.8%	92.5%

Table 6: Training on the WSJ Corpus and Testing on the Respective Corpus.

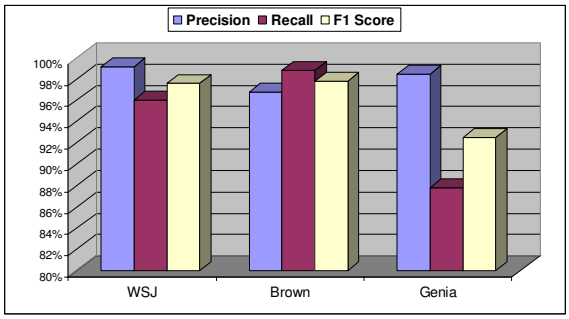

Figure 1: Training on the WSJ Corpus and Testing on the Respective Corpus.

We see a slight reduction in each metric for the WSJ test data relative to the validation data (***Table 5***). The performance on the Brown corpus has lower precision, higher recall, and a higher F1 score than the WSJ corpus. Testing on the GENIA corpus yields very good precision (98.5%), but a significantly lower recall (87.8%) and F1 score (92.5%).

3.3. Training and Testing on the Respective Corpora

To see how much improvement we could obtain by training on the same corpus that we tested on, we trained and tested our classifier on the Brown and GENIA corpora. The results are summarized in ***Table 7*** and ***Figure 2***. We see a slight improvement in the precision and F1 score for the Brown corpus. For the GENIA corpus, however, we see a slight reduction in precision (1%) and a significant improvement in recall (13%) and F1 score (6%).

Corpus	Precision	Recall	F1 Score
Brown	97.0%	98.8%	97.9%
GENIA	97.2%	98.8%	98.0%

Table 7: Training and Testing on the Brown and GENIA Corpora.

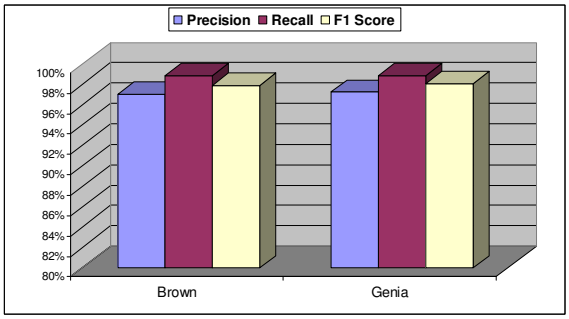

Figure 2: Training and Testing on the Brown and GENIA Corpora.

3.4. Using POS Tags: Training on WSJ Corpus and Testing on the Respective Corpora

In addition, we added POS tags to the best feature set since the data sets described above were already tagged. Although we tested various combinations of the previous tag, current tag, and next tag, the best results, shown in ***Table 8***, were obtained using features that included all three tags. Comparing these results to those in ***Table 6***, we see a decrease in precision, an increase in recall, and little change in the F1 score for each corpus.

Corpus	Precision	Recall	F1 Score
WSJ	98.8%	96.8%	97.8%
Brown	96.2%	99.3%	97.7%
GENIA	98.2%	89.5%	93.6%

Table 8: Training on the WSJ Corpus and Testing on the Respective Corpus with POS Tags.

3.5. Using POS Tags: Training and Testing on the Respective Corpora

Table 9 shows the results of training and testing on the Brown and GENIA corpora using POS tags in the features. Comparing these to the results in ***Table 7***, we see a slight decrease in precision and a slight increase in recall and F1 score for the Brown corpus when POS tags are added. For the GENIA corpora, however, we see a more substantial increase using POS tags, achieving precision of 99.6%, recall of 99.9%, and an F1 score of 99.8%.

Corpus	Precision	Recall	F1 Score
Brown	96.8%	99.1%	98.0%
GENIA	99.6%	99.9%	99.8%

Table 9: Training and Testing on the Brown and GENIA Corpora with POS Tags.

4. Discussion

The MaxEnt classifier that was trained on labeled sentence boundaries performed very well. Training and testing on the WSJ corpus, we obtained a precision of 99.2%, a recall of 96.1%, and an F1 score of 97.6%. Adding POS tags to the best feature set gave a slight decrease in precision (98.8%) and an increase in recall (96.8%). Thus, it is important to note that POS tags are generally not necessary to achieve excellent results in sentence boundary detection when training and testing on the same corpus.

The MaxEnt classifier was also reasonably robust to testing on other corpora. For example, on the Brown corpus without POS tags for features, we obtained a lower precision (96.8%) than on the WSJ corpus, but a higher recall (98.8%) and F1 score (97.8%). Testing on the GENIA corpus, which contains medical data in a different domain, yielded a comparable precision (98.5%), but a lower recall (87.8%) and F1 score (92.5%). Adding POS tags generally reduced the precision, but increased the recall slightly. Of course, training on the respective corpus rather than the WSJ corpus gave much better results, as shown in ***Table 9***.

Next, we present some common errors that we found in the model's prediction of sentence boundaries for each corpus. Each includes the original sentence followed by the error description. The errors fall into two obvious categories of either missing an EOS that the corpus had labeled or mislabeling a non-EOS word as an EOS. The cause of these errors was either that the features were not selective enough to find the true EOS, the test data was unobserved in the training data, or the corpus was inconsistently or erroneously hand-labeled.

4.1. Errors in Evaluating the WSJ Corpus

Sentence: Connecticut -- $ 100 million of general obligation capital appreciation bonds, College Savings Plan, 1989 Series B, via a Prudential-Bache Capital Funding group.
Error: In the corpus, "Connecticut" was labeled as one sentence with "--" labeled as the EOS. However, our model failed to predict this EOS and several others like it.

Sentence: FEDERAL HOME LOAN MORTGAGE CORP. Freddie Mac: Posted yields on 30-year mortgage commitments for delivery within days.
Error: This was another common source of error, where ":" was labeled as the EOS in the corpus, but not in our model.

Although the best feature set (***Table 2***) included features for both "--" and ":" as EOS, these were probably not selective enough.

Sentence: "Schwarzwaldklinik," Black Forest Clinic, a kind of German St. Elsewhere set in a health spa.
Error: This was an interesting error, where we predict the "." before the word "Elsewhere" as an EOS. Although "St." was undoubtedly an abbreviation in the training data, it most likely stood for "Street" instead of "Saint". Furthermore, since the word following "St." ("Elsewhere") was capitalized, it is not surprising that our model had difficulty.

4.2. Errors in Evaluating the Brown Corpus

Sentence: ...much more so that Allah and Jehovah and the rest. 'fess up -- don't you??
Error: Here the "." before the word "'fess" was labeled as the EOS in the corpus, but our model missed it. As seen in this example, there were a couple of things to throw us off – not a capital letter, but an apostrophe beginning the sentence.

Sentences: "What's the matter with the music"?? Moreland asked.
"Uh well, what can I do" ??, many of you will say.
He took his glass, clinked it against mine, and said "Toujors gai, what the hell"!! Borrowing a line from Don Marquis Mehitabel.
Error: There were many such instances where a quoted sentence ends with a "??" or "!!" outside the end of the quote. In each of these instances, our model predicted an EOS at "??" or "!!", whereas the true EOS is not until the period. Even though "!!" was not labeled as the EOS in the corpus, the next word ("Borrowing") began with a capital letter, making it particularly difficult to classify.

4.3. Errors in Evaluating the GENIA Corpus

Sentence: ...and to activate HIV-1 transcription. Anti-TNF-alpha antibody but not anti-IL-1 beta antibody strongly inhibited....
Error: In this example, we guess the period after "transcription" as the EOS, but the corpus does not consider this an EOS. We think that this is actually a mistake in the hand-labeled data.

Sentence: Cell proliferation of CEM-C7 cells cultured in both serum-free media has been sustained for 3 mo. with culture doubling times of about 25h for both serum-supplemented....
Error: Here we guess that mo. is not the EOS, but it actually is in the corpus. Again, it looks as though this is an erroneous EOS in the corpus, since "mo." is most likely an abbreviation for month.

4.4. Errors While Training on WSJ and Testing on GENIA

Sentence: UI-90351381
Error: There were several such instances in the GENIA corpus, which we think is probably a number for a scientific document or the name of a drug. While training on the WSJ corpus, we fail to see such sentences, so these errors are not surprising. Thus, our model does not recognize that "90351381" is the EOS.

Sentence: NF-IL6 mRNA was found in human Jurkat T cells and in the mouse Th2 clone D10, but not in the TH1 clone 29. rNF-IL6 expressed in bacteria was shown to specifically bind to PRE-I.
Error: Here our model missed the EOS before "rNF-IL6". We think that this is because it is not capitalized, as would be expected for the beginning of a sentence.

4.5. Errors While Training on WSJ and Testing on Brown

Since the WSJ and the Brown corpora have similar content, there are not many differences in the mistakes on the Brown corpus, regardless of the corpus that we train on.

4.6. Improving the MaxEnt Classifier's Performance

Despite some errors in the hand-labeled data, we believe that there are ways in which the classifier could be improved. In particular, we could include a collection of common proper nouns, adding a feature to the classifier whenever one of these proper nouns was encountered. This would increase the accuracy of the classifier, such as in the following example trigram: "Anita Davis Of". Here, "Davis" is the true EOS, since the sentence is merely the proper noun, "Anita Davis". However, our classifier misses it, since it guesses without the knowledge of proper nouns. For similar reasons, we could include a list of abbreviations induced from the training data, as in [1, 3].

Additionally, we could add features to target the particular data set, such as noticing that the WSJ data set contains newspaper headings. For example, we miss the EOS in the heading, "Reluctant Advertisers Try Soft-Sell Spots", where the next word is all uppercase, "CALL". Unless we knew to look for a sequence of capitalized words for headings, we might not be able to detect this case. Some other features that we could have tried include using prefixes or suffixes, as in [1].

5. Conclusions

Applying our MaxEnt classifier with finely-tuned features to the sentence boundary detection task was very successful. In fact, we were able to achieve excellent results without including POS tags in the features. The classifier was also quite robust to testing on the Brown corpus and reasonably robust to testing on GENIA.

By examining specific errors that occurred in the sentence boundary detection for each corpus, we were able to gain some insight as to where the MaxEnt classifier was having difficulty. By adding specific features targeted to these errors, we believe that we could obtain even better results. However, state-of-the-art sentence boundary detection is only 98.8% accurate [1], so we anticipate that the additional improvement would be small using a MaxEnt classifier.

To further assess the robustness of the MaxEnt classifier, we could test it on different languages as in [4] or on a variety of corpora, such as email or legal documents.

6. References

[1] Reynar, J.C. and A. Ratnaparkhi. "A Maximum Entropy Approach to Identifying Sentence

Boundaries", *Proc. of ANLP97,* Washington D.C., p. 16-19, 1997.

[2] GENIA corpus website: http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/.

[3] Mikheev, Andrei. "Tagging Sentence Boundaries", *Proc. Of NAACL*, p. 264-271, 2000.

[4] Palmer, D.D. and M.A. Hearst. "Adaptive Multilingual Sentence Boundary Disambiguation", *Computational Linguistics*, 23(2): 241-269, 1997.

[5] Mikheev, Andrei, Feature Lattices and Maximum Entropy Models, p.848-854, 1998.

[6] Wang, H. and Y. Huang. "Bondec—A Sentence Boundary Detector", CS224N Project, Stanford, 2003.