Period Disambiguation with Maxent Model

Chunyu Kit and Xiaoyue Liu

Department of Chinese, Translation and Linguistics,
City University of Hong Kong, 83 Tat Chee Ave., Kowloon, Hong Kong
{ctckit, xyliu0}@cityu.edu.hk

Abstract. This paper presents our recent work on period disambiguation, the kernel problem in sentence boundary identification, with the maximum entropy (Maxent) model. A number of experiments are conducted on PTB-II WSJ corpus for the investigation of how context window, feature space and lexical information such as abbreviated and sentence-initial words affect the learning performance. Such lexical information can be automatically acquired from a training corpus by a learner. Our experimental results show that extending the feature space to integrate these two kinds of lexical information can eliminate 93.52% of the remaining errors from the baseline Maxent model, achieving an F-score of 99.8227%.

1 Introduction

Sentence identification is an important issue in practical natural language processing. It looks simple at first glance since there are a very small number of punctuations, namely, period (“.”), question mark (“?”), and exclamation (“!”), to mark sentence ends in written texts. However, not all of them are consistently used as sentence ends. In particular, the use of the dot “.” is highly ambiguous in English texts. It can be a full stop, a decimal point, or a dot in an abbreviated word, a numbering item, an email address or a ULR. It may be used for other purposes too. Below are a number of examples from PTB-II WSJ Corpus to illustrate its ambiguities.

(1) `Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.`
(2) `The spinoff also will compete with International Business Machines Corp. and Japan’s Big Three -- Hitachi Ltd., NEC Corp. and Fujitsu Ltd.`
(3) `The government’s construction spending figures contrast with a report issued earlier in the week by McGraw-Hill Inc.’s F.W. Dodge Group.`

Frequently, an abbreviation dot coincides with a full stop, as exemplified by “Ltd.” in (2) above. A number followed by a dot can be a numbering item, or simply a normal number at sentence end.

In contrast to “.”, “!” and “?” are rarely ambiguous. They are seldom used for other purposes than exclamation and question marks. Thus, the focus of

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 223–232, 2005.

sentence identification is on period disambiguation to resolve the ambiguity of ".": Whenever a dot shows up in a text token, we need to determine whether or not it is a true period. It is a yes-no classification problem that is suitable for various kinds of machine learning technology to tackle.

Several approaches were developed for sentence splitting. These approaches can be categorized into three classes: (1) rule-based models consisting of manually constructed rules (e.g., in the form of regular expression), supplemented with abbreviation lists, proper names and other relevant lexical resources, as illustrated in [1]; (2) machine learning algorithms, e.g., decision tree classifiers [11], maximum entropy (Maxent) modelling [10] and neural networks [8], among many others; and (3) syntactic methods that utilize syntactic information, e.g., [6] is based on a POS tagger. The machine learning approaches are popular, for period disambiguation is a typical classification problem for machine learning, and the training data is easily available.

Our research reported in this paper explores how context length and feature space affects the performance of the Maxent model for period disambiguation. The technical details involved in this research are introduced in Section 2, with a focus on feature selection and training algorithm. Section 3 presents experiments to show the effectiveness of context length and feature selection on learning performance. Section 4 concludes the paper with our findings: putting frequent abbreviated words or sentence-initial words into the feature space significantly enhances the learning performance, and using a three-word window context gives better performance than others in terms of the F-score. The best combination of the two kinds of lexical information achieves an F-score of 99.8227%, eliminating 93.5% remaining errors from the baseline Maxent model.

2 Feature Selection

The problem of period disambiguation can be formulated as a statistical classification problem. Our research is aimed at exploring the effectiveness of Maxent model [2,12] tackling this problem when trained with various context length and feature sets.

Maxent model is intended to achieve the most unbiased probabilistic distribution on the data set for training. It is also a nice framework for integrating heterogeneous information into a model for classification purpose. It has been popular in NLP community for various language processing tasks since Berger et al. [2] and Della Pietra et al. [3] presenting its theoretical basis and basic training techniques. Ratnaparkhi [9] applied it to tackle several NL ambiguity problems, including sentence boundary detection. Wallach [14] and Malouf [4] compared the effectiveness of several training algorithms for Maxent model.

There are a number of full-fledged implementations of Maxent models available from the Web. Using the OpenNLP MAXENT package from `http://maxent.sourceforge.net/`, acknowledged here with gratitude, we are released from the technical details of its implementation and can concentrate on examining the effectiveness of context length and feature space on period disam-

biguation. Basically, our exploration is carried out along the following working procedure: (1) prepare a set of training data in terms of the feature space we choose; (2) train the Maxent model, and test its performance with a set of testing data; (3) examine the errors in the test outcomes and adjust the feature space for the next round of training and testing towards possible improvement.

2.1 Context and Features

To identify sentence boundaries, a machine learner needs to learn from the training data the knowledge whether or not a dot is a period in a given context . Classification decision is based on the available contextual information. A context is the few tokens next to the target. By "target" we refer to the "." to be determined whether or not it is a period, and by "target word" (or "dotted word") we refer to the token that carries the dot in question. The dot divides the target word into prefix and suffix, both of which can be empty. Each dot has a *true* or *false* answer for whether it is a true period in a particular context, as illustrated by the following general format.

$$[\text{ preceding-words prefix}.\text{suffix following-words }] \rightarrow \text{Answer: } true/false\ . \quad (1)$$

Contextual information comes from all context words surrounding the target dot, including its prefix and suffix. However, instead of feeding the above contextual items to a machine learner as a number of strings for training and testing, extracting special and specific features from them for the training is expected to achieve more effective results. To achieve a learning model as unbiased as possible, we try to extract as many features as possible from the context words, and let the training algorithm to determine their significance. The main cost of using a large feature set is the increase of training time. However, this may be paid off by giving the learner a better chance to achieve a better model.

Table 1. Features for a context word

Feature	Description	Example
`IsCap`	Starting with a capital letter	`On`
`IsRpunct`	Ending with a punctuation	`Calgary,`
`IsLpunct`	Starting with a punctuation	``` ``We ```
`IsRdot`	Ending with a dot	`billions.`
`IsRcomma`	Ending with a comma	`Moreover,`
`IsEword`	An English word	`street`
`IsDigit`	An numeric item	`25%, 36`
`IsAllCap`	Consisting of only capital letters (& dots)	`WASHINGTON`

The feature set for a normal context word that we have developed through several rounds of experiments along the above working procedure are presented in Table 1. Basically, we extract from a word all features that we can observe from its

Table 2. Features for a target word

Feature	Description	Example
`IsHiphenated`	Containing a dash	`non-U.S.`
`IsAllCap`	Consisting of only capital letters (& dots)	`D.C.`
`IsMultiDot`	Containing more than one dot	`N.Y.,`
`prefixIsNull`	A null prefix	`.270`
`prefixIsRdigit`	Ending with a digit	`45.6`
`prefixIsRpunct`	Ending with a punctuation	`0.2%.`
`prefixIsEword`	An English word	`slightly.`
`prefixIsCap`	Starting with a capital letter	`Co.`
`suffixIsNull`	A null suffix	`Mr.`
`suffixIsLdigit`	Starting with a digit	`78.99`
`suffixIsLpunct`	Starting with a punctuation	`Co.'s`
`suffixIsRword`	Ending with a word	`Calif.-based`
`suffixIsCap`	Starting with a capital letter	`B.A.T`

text form. For feature extraction, this set is applied equally, in a principled way, to all context words. The feature set for both parts of a target word is highly similar to that for a context word, except for a few specific to prefix and/or suffix, as given in Table 2, of 13 features in total. The data entry for a given dot, for either training or testing, consists of all such features from its target word and each of its context words. Given a context window of three tokens, among which one is target word, there are 2×8+13=29 features, plus an answer, in each data entry for training.

After feature extraction, each data entry originally in the form of (1) is turned into a more general form for machine learning, as shown in (2) below, consisting of a feature value vector and an answer.

$$f : [f_1 = v_1, f_2 = v_2, f_3 = v_3, \cdots, f_n = v_n] \ \rightarrow \ a\text{: } \textit{true/false} \ . \tag{2}$$

Accordingly, the Maxent model used in our experiments has the following distribution in the exponential form:

$$p(a|f) = \frac{1}{Z(f)} \exp(\sum_i \lambda_i \delta(f_i, a)) \ , \tag{3}$$

where λ_i is a parameter to be estimated for each i through training, the feature function $\delta_i(f_i, a) = v_i$ for the feature f_i in a data entry $f \rightarrow a$, and the normalization factor

$$Z(f) = \sum_a \exp(\sum_i \lambda_i \delta(f_i, a)) \ . \tag{4}$$

2.2 Abbreviation List and Sentence-Initial Words

In addition to the above features, other types of contextual information can be helpful too. For example, abbreviated words like "Dr.", "Mr." and "Prof."

may give a strong indication that the dot they carry is very unlikely to be a period. They may play the role of counter-examples. Another kind of useful lexical resource is sentence-initial words, e.g., "The", "That" and "But", which give a strong indication that a preceding dot is very likely to be a true period.

In order to integrate these two kinds of lexical resource into the Maxent model, we introduce two multi-valued features, namely, `isAbbr` and `isSentInit`, for the target word and its following word, respectively. They are both multi-valued feature function. A list of abbreviated words and a list of sentence-initial words can be easily compiled from a training corpus. Theoretically, the larger the lists are, the better the learning performance could be. Our experiments, to be reported in the next section, show, however, that this is not true, although using the most frequent words in the two lists up to a certain number does lead to a significant improvement.

3 Experiments and Results

3.1 Corpus

The corpus used for our experiments is the PTB-II WSJ corpus, a refined version of PTB [5]. It is particularly suitable for our research purpose. In contrast to BNC and Brown corpus, the WSJ corpus indeed contains many more dots used in different ways for various purposes. Sentence ends are clearly marked in its POS tagged version, although a few mistakes need manual correction. Among 53K sentences from the corpus, 49K end with ".". This set of data is divided into two for training and testing by the ratio of 2:1. The baseline performance by brute-force guess of any dot as a period is 65.02% over the entire set of data.

3.2 Baseline Learning Performance

Our first experiment is to train a Maxent model on the training set with a three-word context window in terms of the features in Tables 1 and 2 above. The performance on the open test is presented in Table 3. It is the baseline performance of the Maxent model.

Table 3. Baseline learning performance of Maxent model

Precision (%)	Recall (%)	F-score (%)
97.55	96.97	97.26

3.3 Effectiveness of Context Window

To examine how context words affect the learning performance, we carry out a number of experiments with context windows of various size. The experimental results are presented in Fig. 1, where `x` stands for the position of target word and

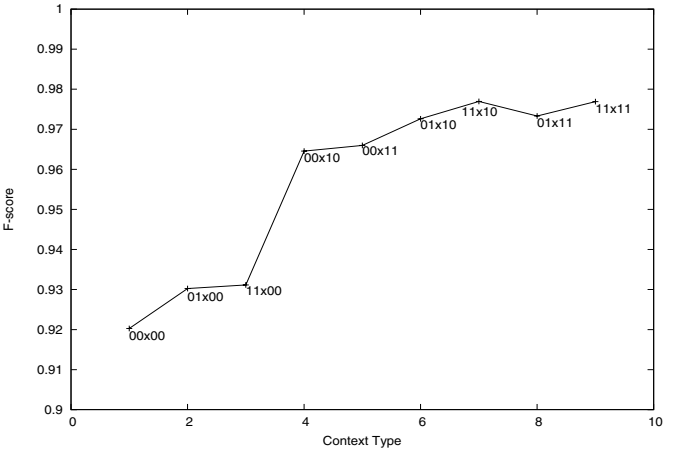

Fig. 1. Effectiveness of context window

1 for a context word in use. For example, `01x10` represents a context window consisting of a target word, its preceding and following words. Each such window is itself a context type.

We can observe from the results that (1) the features extracted from the target word itself already lead the Maxent model to an F-score beyond 92%, (2) the context words preceding the target word are less effective, in general, than those following the target, and (3) combining context words on both sides outperforms those on only one side. The best three context types and the correspondent performance are presented in Table 4. Since they are more effective than others, the experiments to test the effectiveness of abbreviated words and sentence-initial words are based on them.

Table 4. Outperforming context types and their performance

Context Type	`01x10`	`11x10`	`11x11`
F-score (%)	97.2623	97.6949	97.6909

3.4 Effectiveness of Abbreviated Words

Information about whether a target word is an abbreviation plays a critical role in determining whether a dot is truly a period. To examine the significance of such information, an abbreviation list is acquired from the training data by dotted word collection, and sorted in terms of the difference of each item's occurrences in the middle and at the end of a sentence. It is assumed that the greater this difference is, the more significant a dotted word would be as a counter-example. In total, 469 such words are acquired, among which many are not really abbreviated words. A series of experiments are then conducted by adding the next 50 most frequent dotted words to the abbreviation list for model training each time. To utilize such

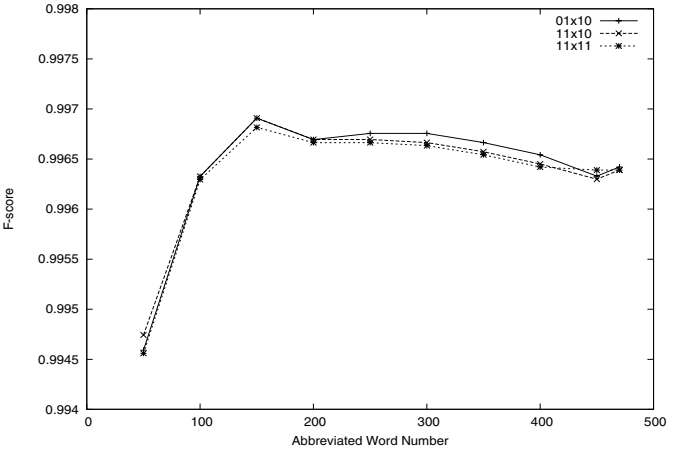

Fig. 2. Effectiveness of abbreviation list

Table 5. Effectiveness of abbreviation list

Context Type	`01x10`	`11x10`	`11x11`
F-score (%)	99.6908	99.6908	99.6815
Increase	+2.4285	+1.9959	+1.9906

lexical resource, a multi-valued feature `isAbbr` is introduced to the feature set to indicate whether a target word is in the abbreviation list and what it is. That is, all words in the list actually play a role equivalent to individual bi-valued features, under the umbrella of this new feature.

The outcomes from the experiments are presented in Fig. 2, showing that performance enhancement reaches rapidly to the top around 150. The performance of the three best context types at this point is given in Table 5, indicating that an abbreviation list of 150 words leads to an enhancement of 1.99–2.43 percentage points, in comparison to Table 4. This enhancement is very significant at this performance level. Beyond this point, the performance goes down slightly.

3.5 Effectiveness of Sentence-Initial Words

In a similar way, we carry out a series of experiments to test the effectiveness of sentence-initial words. In total, 4190 such words (word types) are collected from the beginning of all sentences in the training corpus. Every time the next 200 most frequent words are added to the sentence-initial word list for training, with the aid of another multi-valued feature `isSentInit` for the context word immediately following the target word.

Experimental outcomes are presented in Fig. 3, showing that the performance maintains roughly at the same level when the list grows. Until the very end,

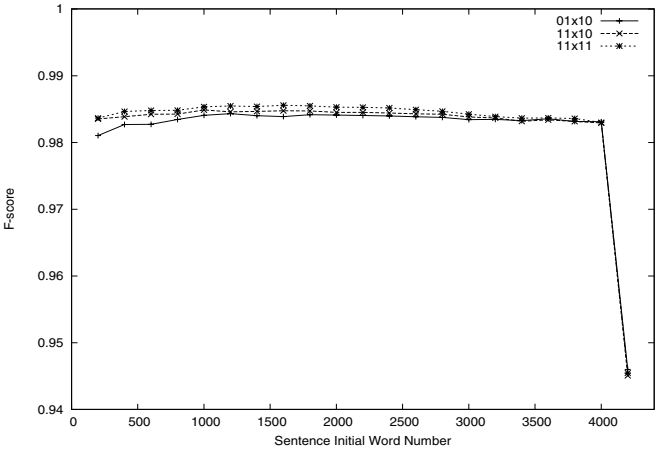

Fig. 3. Effectiveness of sentence-initial words

Table 6. Performance enhancement by sentence-initial words

Context Type	01x10	11x10	11x11
List size	1200	1000	1200
F-score (%)	98.4307	98.4868	98.5463
Increase	+1.1784	+0.7919	+0.8554

when those most infrequent (or untypical) sentence-initial words are added, the performance drops rapidly. The numbers of sentence-initial words leading to the best performance with various context types are presented in Table 6. This list of words lead to a significant performance enhancement of 0.79–1.18 percentage points, in comparison to Table 4.

3.6 Combination of Two Lists

Through the experiments reported above we find the optimal size of abbreviation list and sentence-initial words, both in the order of their frequency ranks, in each context type of our interests. The straightforward combination of these two lists in terms of these optimal sizes leads to almost no difference from using abbreviation list only, as presented in Table 7.

Table 7. Performance from simple combination of the two lists

Context Type	01x10	11x10	11x11
Sentence-initial words	1200	1000	1200
Abbreviation list	150	150	150
F-score (%)	99.7064	99.7156	99.6912

Table 8. Performance from various size combination of the two lists

Sentence-initial words	Abbreviation list	F-score 01x10	11x10	11x11
100	200	99.7646%	99.7738%	99.7707%
100	400	99.7125%	99.7033%	99.7002%
100	600	99.7033%	99.6971%	99.6971%
100	800	99.6788%	99.6941%	99.6911%
100	1000	99.6696%	99.6818%	99.6696%
100	1200	99.6635%	99.6574%	99.6544%
150	200	99.8013%	99.7890%	99.7921%
150	400	99.7431%	99.7339%	99.7369%
150	600	99.7431%	99.7370%	99.7370%
150	800	99.7401%	99.7309%	99.7278%
150	1000	99.7156%	99.7156%	99.7064%
150	1200	99.7064%	99.7034%	99.6912%
200	200	99.8227%	99.7890%	99.7921%
200	400	99.7584%	99.7461%	99.7339%
200	600	99.7523%	99.7431%	99.7339%
200	800	99.7462%	99.7370%	99.7340%
200	1000	99.7309%	99.7125%	99.7064%
200	1200	99.7095%	99.6973%	99.6911%

To explore the optimal combination of the two lists, a series of experiments are carried out near each list's optimal size. The results are presented in Table 8, showing that the best combination is around 200 words from each list and any deviation from this point would lead to observable performance declination. The best performance at this optimal point is 99.8227% F-score, achieved with the `01x10` context type, which is significantly better than the best performance using any single list of the two.

Comparing to the baseline performance of the Maxent model in Table 4, we can see that this improvement increases only 99.8227 - 97.2623 = 2.5604 percentage points. Notice, however, that it is achieved near the ceiling level. Its particular significance lies in the fact that $\frac{99.8227-97.2623}{100-97.2623} = 93.52\%$ remaining errors from the baseline model are further eliminated by this combination of the two lists, both of which are of a relatively small size.

4 Conclusions

We have presented in the above sections our recent investigation into how context window, feature space and simple lexical resources like abbreviation list and sentence-initial words affect the performance of the Maxent model on period disambiguation, the kernel problem in sentence identification. Our experiments on PTB-II WSJ corpus suggest the following findings: (1) the target word itself provides most useful information for identifying whether or not the dot it carries is a

true period, achieving an F-score beyond 92%; (2) unsurprisingly, the most useful context words are the two words next to the target word, and the context words to its right is more informative in general than those to its left; and (3) extending the feature space to utilize lexical information from the most frequent 200 abbreviated words and sentence-initial words, all of which can be straightforwardly collected from the training corpus, can eliminate 93.52% remaining errors from the baseline model in the open test, achieving an F-score of 99.8227%.

Acknowledgements

The work described in this paper was supported by the Research Grants Council of HKSAR, China, through the CERG grant 9040861 (CityU 1318/03H). We wish to thank Alex Fang for his help.

References

1. Aberdeen, J., Burger, J., Day, D., Hirschman, L., Robinson, P., and Vilain, M.: Mitre: Description of the alembic system used for muc-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Columbia, Maryland. Morgan Kaufmann (1995)
2. Berger, A., Pietra, S.D., and Pietra, V.D.: A maximum entropy approach to natural language processing. *Computational linguistics.* (1996) 22(1):39–71
3. Della Pietra, S., Della Pietra, V., and Lafferty, J.: Inducing features of random fields. *Transactions Pattern Analysis and Machine Intelligence.* (1997) 19(4): 380–393
4. Malouf, R.: A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of CoNLL-2002*, Taipei, Taiwan (2002) 49–55
5. Marcus, M.P., Santorini, B., and Marcinkiewicz, M.A.: Building a large annotated corpus of english: The penn treebank. *Computational Linguistics.* (1993) 19(2): 313–329
6. Mikheev, A.: Tagging sentence boundaries. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'2000).* (2000)
7. Mitchell, T.: *Machine Learning.* McGraw Hill, New York (1997)
8. Palmer, D.D. and Hearst, M.A.: Adaptive Multilingual Sentence Boundary Disambiguation. *Computational Linguistics.* (1997) 23(2):241–267
9. Ratnaparkhi, A.: *Maximum entropy models for natural language ambiguity resolution.* Ph.D. dissertation, University of Pennsylvania (1998)
10. Reynar, J.C. and Ratnaparkhi, A.: A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, D.C. (1997)
11. Riley, M.D.: Some applications of tree-based modelling to speech and language indexing. In *Proceedings of the DARPA Speech and Natural Language Workshop.* Morgan Kaufmann (1989) 339–352
12. Rosenfeld, R.: *Adaptive statistical language modeling: A Maximum Entropy Approach.* PhD thesis CMU-CS-94. (1994)
13. Van Rijsbergen, C.J.: *Information Retrieval.* Butterworths, London (1979)
14. Wallach, H.M.: *Efficient training of conditional random fields.* Master's thesis, University of Edinburgh (2002)