Chinese Unknown Word Extraction by Mining Maximized Substrings

Mo Shen, Daisuke Kawahara, and Sadao Kurohashi
Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku,
Kyoto, 606-8501, Japan
shen@nlp.ist.i.kyoto-u.ac.jp {dk,kuro}@i.kyoto-u.ac.jp

Abstract

The issue of identifying out-of-vocabulary (OOV) words is a major difficulty in Chinese word segmentation. We address this issue by applying a very efficient algorithm for extracting *maximized substrings* (Shen et al., 2013) from a large-scale raw text, which form a list of unknown word candidates. We then apply techniques such as Short-term Store and Lexicon-based Voting to reduce the noises in the extracted list of unknown words. We demonstrate that our method outperforms previous studies in both accuracy and efficiency.

1. Introduction

Chinese sentences are written without explicit word boundaries, which makes Chinese word segmentation (CWS) an initial and important step in Chinese language processing. As the Chinese language continually and rapidly evolves particularly with the today's rapid growth of the internet, the lack of knowledge of vocabulary presents the biggest challenge in Chinese word segmentation. In fact, previous studies have shown that with a comprehensive lexicon, even a simple maximum matching segmentation algorithm can yield an F-score as high as 0.99 (Sproat and Emerson, 2003). It is impossible, however, to collect a complete list of Chinese words by hand. Therefore, it is necessary to develop techniques that automatically develop vocabulary lists from large-scale web texts, which is also the task of Chinese unknown word extraction (Feng et al., 2004; Ye et al., 2013).

In this paper, we address this issue by applying the method in (Shen et al., 2013) which extracts substrings as reliable word boundary estimations. The technique uses large-scale unlabeled data, and processes it on the fly. We also introduce techniques such as Short-term Store and Lexicon-based Voting to reduce the noise in the extracted list of unknown words.

2. Approach

2.1 Maximized Substring: the Definition

To illustrate the basic idea of *maximized substrings*, we first consider a corpus which consists of only two sentences:

(1) 一部路易斯・布努埃尔的超现实主义电影
(2) 路易斯・布努埃尔是一位西班牙超现实主义者

Consider the two words "路易斯・布努埃尔" (Luis Buñuel) and "超现实主义" (Surrealism). Both words appear multiple times in this corpus, and both of them have surrounding characters different from each other, which means they are longest non-overlapping substrings in the local context. In fact, these two words are the only substring in this corpus that satisfies these conditions. On the other hand, some substrings like "布努埃" and "超现实主", although being frequent, are meaningless. These substrings are "internal": they are overlapped with other substrings and can be further extended by its surrounding characters without losing their frequencies. We are only interested in the substrings that share the properties with "路易斯・布努埃尔" (Luis Buñuel) and "超现实主义" (Surrealism), and we use the term *maximized substrings* (Shen et al., 2013) to describe these substrings.

Formally, *maximized substring* is defined as follows. Given a document D which is a collection of sentences, denote a length n substring which starts with character c_t by $s_t = [c_t c_{t+1} \dots c_{t+n-1}]$. s_t is called a *maximized substring* if:

1. It has a set of distinct occurrences M with at least two elements:

   $\mathrm{M} = \{s_{t_1}, s_{t_2}, \ldots, s_{t_m}\}$, $m > 1$, $t_1 \neq t_2 \neq \cdots \neq t_m$ s.t. $_{t_1} = {}_{t_2} = \cdots = {}_{t_m}$; and

2. $c_{t_i-1} \neq c_{t_j-1}$ and $c_{t_i+n} \neq c_{t_j+n}$ $\forall i, j = 1,2 \ldots, \mathrm{m}, i \neq j$.

2.2 Maximized Substring Extraction: Algorithm and Data Structure

To address the issue of mining *maximized substring*s, we use the substring extraction algorithm described in (Shen et al., 2013). The algorithm is illustrated in Algorithm 1. For each unreached positions in the document, it searches for the longest one of the extracted substrings. It then goes through the occurrence list of the substring found. If the condition 2 of the definition of maximized substring remains satisfied, the position is added as a new occurrence; otherwise it creates a new entry in the data structure by extending the condition-violating occurrence with the succeeding characters until the condition is met again.

2.3 Short-Term Store

Although maximized substrings extracted by Algorithm 1 provide reliable estimations of word boundaries, they do not always form single words in Chinese. Some noises are introduced during the extraction, such as the sequences of single-character words "在沪将" (in Shanghai will). This kind of noise is unpredictable and is also hard to be filtered out during the post-processing.

To address this problem, we take advantage of a linguistic phenomenon. It has been observed that a word occurring in the recent past has a much higher probability to occur again soon than its overall frequency (Kuhn and Mori, 1990). This observation is applicable to the task of maximized substring extraction in the following way. Suppose a substring is registered into the data structure. If the substring is in fact a word (especially when it is a technical term or a named entity), it is much more likely to reoccur in the next 50 to 100 sentences rather than the rest part of the corpus; otherwise the substring should have a more unified probability of reoccurrence across the entire corpus.

We thus introduce a functionality in the process of maximized substring extraction, called "Short-term Store"(STS), to analogize the cache component in speech recognition as well as the human phonological working memory in language acquisition (Shen et al., 2013). It restricts the visible context in extracting the next candidate of a registered substring; the length of the context is proportional to the current count of the substring. For a registered substring s in the data structure, the extraction algorithm scans for a certain number of sentences after the latest occurrence of the substring, where the number of sentences $\mathrm{D}(s)$ is determined as follows:

$$\mathrm{D}(s) = \begin{cases} \lambda \cdot count(s), & if\ count(s) < \theta \\ \infty, & otherwise \end{cases}$$

where $count(s)$ is the current number of occurrences of s in the data structure. With each count of occurrences, the parameter λ contributes a fixed-length distance to the visible context. The parameter θ works as a threshold of reliability, which means if s has been observed at least θ times in a short period, we can regard s as a word or a sequence of words with a high level of

Algorithm 1: Maximized Substring Extraction

```
1  procedure ExtractMaxSub(D)
2    i ← 0, H ← ∅
3    until i reaches the end of document D
4        ← longest element in H forward-
5          searchable from position i
6      if |s| = 0  ◅ empty string
7          _new ← [c_i] ◅ single-character string
8        occurList_new ← {i}
9        ◅ starting position of new occurrence
10       H.Add(⟨s_new, occurList_new⟩)
11       ◅ associate string s with its occurrence
12         list and add to data structure
13       i ← i + 1
14     else
15       (H, i) ← Maximize(H, i, s)
16   return H
17
18 procedure Maximize(H, i, s)
19   t ← 0
20   for each u in s.occurList
21     if c_{u+|s|} = c_{i+|s|}
22       while c_{u+|s|+t} = c_{i+|s|+t}
23         t ← t + 1
24       _new ← [c_i c_{i+1} ... c_{i+|s|+t−1}]
25       occurList_new ← {i, u}
26       H.Add(⟨s_new, occurList_new⟩)
27       i ← i + |s| + t
28       return (H, i)
29   s.occurList.Add(i)
30   i ← i + |s|
31   return (H, i)
```

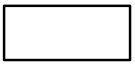
Sentence: $s = \cdots c_{i-1}c_ic_{i+1} \ldots c_{j-1}c_jc_{j+1} \ldots$		Representation
Maximized substring $m = c_ic_{i+1}c_{i+2} \ldots c_j$		
Lexicon entry $l = c_0c_1c_2 \ldots c_n$		
ID	Relative Position	
L1	$i \leq 0 < j < n$	
L2	$0 < i < n \leq j$	

Table 1. Lexicon-based voting by relative positions between a maximized substring and a lexicon entry.

confidence, thus $\mathrm{D}(s) = \infty$ means s is no longer subject to periodical decaying, and will stay in the data structure statically.

During the scanning of the $\mathrm{D}(s)$ sentences, if a new occurrence of s is found, after it is added into the data structure, $\mathrm{D}(s)$ will be re-calculated immediately to start a new scanning period; otherwise, we remove the earliest occurrence of s from the data structure, and then re-calculate $\mathrm{D}(s)$.

2.4 Lexicon-based Voting

There is a typical kind of noises in the extracted list of maximized substrings, namely, those like the substring "中美经", which is resulted from two phrases "中美经济" (China and U.S. economy) and "中美经贸" (China and U.S. economic and trade). This happens when the boundary of a maximized substring is a shared boundary character of multiple other words. As in this example, the ending character "经" of the maximized substring is a shared character at the beginning of "经济" (economy) and "经贸" (economic and trade). In other words, characteristics of this kind of noises can be captured by checking the context of maximized substrings with system's lexicon.

For each extracted maximized substring, we check its occurrences in the original document with the help of a system's lexicon. If there is any word found in the lexicon that forms the relative position L1 or L2 listed in Table 1 with this occurrence, it votes for discarding the maximized substring. If at least 50% of the occurrences vote for discarding, we remove the maximized substring from the extracted list.

Substring	Translation
温布尔登	Wimbledon
克拉玛依	Karamay
骨质疏松症	Osteoporosis
小阪善太郎	Kosaka Zentaro
普济禅院	Puji Temple
军事五项	Military pentathlon
珞巴族	Lhoba people
黄玉斌	Huang Yu-bin
利率管理体制	Interest rate regulation system
黎以和谈	Israeli-Lebanese peace talks
加快教育改革	Accelerate education reform
舍维奇	~šević (part of Milošević)

Table 2. Some good examples (upper part of the table) and bad examples (lower part of the table) of extracted unknown words.

3. Evaluation

To demonstrate the effectiveness of our method, we conducted unknown words extraction experiments on Chinese Treebank 7.0 (CTB7). It is difficult to directly evaluate the precision and recall of a list of extracted unknown words, since there is no complete list of unknown words to be compared with. Previous studies have adopted evaluation methods based on hand annotation (Feng et al., 2004). We instead used the word list of CTB7 as gold standard data for evaluation. We used the entire CTB7 dataset as an input text for maximized substring extraction, which has 51,447 sentences. We used the same lexicon that has been used in previous studies (Feng et al., 2004), which has 119,803 Chinese words of two to seven characters[1]. With the words in the lexicon being known, there are 11,722 unknown words remaining in the word list of CTB7.

Table 2 shows some examples of the extracted unknown words which correctly identify unknown words. As we can see from the table, our method is effective in identifying named entities, including names of persons, locations and technical terms. We also show some negative examples in the lower part of this table. The major types of the error include compounds, noun and verb phrases, and partial words.

In Figure 1 we show the performance of three maximized substring-based systems: "MaxSub" represents the maximized substring extraction method described in section 2.2; "MaxSub+LV" represents the previous system plus the post-processing technique of Lexicon-based Voting;

[1] http://www.mandarintools.com/segmenter.html

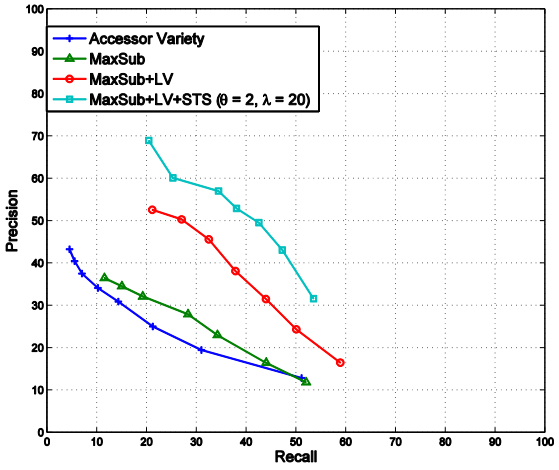


Figure 1. Precision-Recall curves of maximized substring extraction and the *accessor variety* method on CTB7.

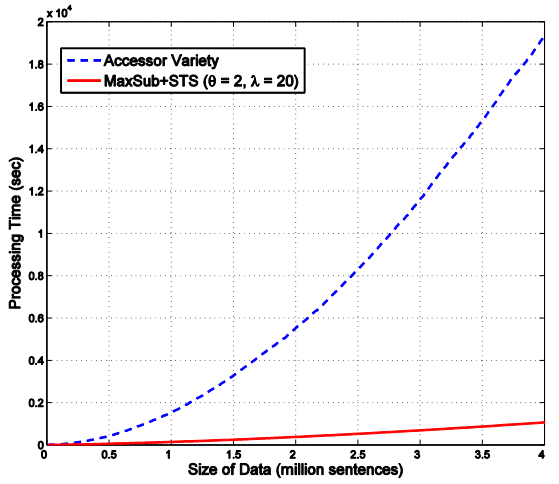


Figure 2. Processing time comparison between maximized substring extraction and the *accessor variety* method on a large-scale text.

"MaxSub+LV+STS" represents the second system with the Short-term Store utilized on during the extraction process. The parameters of Short-term Store we used are shown in the figure, which are the combination that yielded the best performance on this dataset. We also implemented the method of *accessor variety* (Feng et al., 2004) for comparison as shown in the same figure, which is one of the most widely applied Chinese word recognition method. Following the original work, we calculated the accessor variety for substrings of two to seven characters and applied their *Adhesive Judge* rules to reduce the errors. The result shows that our method substantially outperforms the method of accessor variety, and both the Short-term Store and the Lexicon-based Voting techniques significantly contribute to the overall performance of unknown word extraction.

To demonstrate the efficiency of our approach in processing large-scale data, in Figure 2 we compared the processing time of our system ("MaxSub+STS", no post-processing) against the method of *accessor variety* which considers substrings of length two to seven. The unlabeled data we used in this experiment is the first four million sentences of the Xinhua Newswire section in Chinese Gigaword Second Edition. The result shows that the difference in processing time between the two methods can be as large as more than 12 times on this dataset. In addition, our method shows a quasi-linear time complexity while the accessor variety method empirically runs in $O(n^2)$ time.

4. Conclusion

We have proposed an algorithm for unknown word extraction. The Short-term Store and Lexicon-based Voting techniques demonstrated to be effective in error reduction. Our method substantially outperformed the previous work of accessor variety in both effectiveness and efficiency in Chinese unknown word extraction experiments.

References

Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. 2004. Accessor variety criteria for Chinese word extraction. Computational Linguistics, 30(1), pages 75–93.

Roland Kuhn and Renato De Mori. 1990. A Cache-based Natural Language Model for Speech Recognition. IEEE Transaction on Pattern Analysis and Machine Intelligence, 12(6), pages 570-583.

Mo Shen, Daisuke Kawahara and Sadao Kurohashi. 2013. Chinese Word Segmentation by Mining Maximized Substrings. In Proceedings of IJCNLP2013, pages 171-179. Nagoya, Japan.

Richard Sproat and Thomas Emerson. 2003. The First International Chinese Word Segmentation Bakeoff. In Proceedings of the Second SIGHAN Workshop on Chinese Language Processing. Sapporo, Japan.

Yunming Ye, Qingyao Wu, Yan Li, K.P. Chow, Lucas C.K. Hui, and S.M. Yiu. 2013. Unknown Chinese Word Extraction Based on Variety of Overlapping Strings. Information Processing & Management, 49(2), pages 497–512.