Extracting Open Compounds from Text Corpora

Virach Sornlertlamvanich and Hozumi Tanaka
Department of Computer Science, Tokyo Institute of Technology
{virach,tanaka}@cs.titech.ac.jp

1 Introduction

This paper discusses the result of studying the automatic extraction of candidates for open compound registration. The open compound is referred to an uninterrupted sequence of words, generally functions as a single constituent of a sentence [Smadja and McKeown 1990]. We proposed our new method of extraction for the languages which have no specific use of punctuation mark, especially the use of word breaking. Our method is applied to n-gram text data using the statistical observation of the change of frequency of occurrence when the window size of string observation is extended (character) cluster by cluster. We generate both of the rightward and the leftward sorted n-gram data, then determine the left and right boundaries of a string using the methods of *competitive selection* and *unified selection* of the both sorted n-gram data. In this paper, we examine the result by applying our method to Thai text corpora and also introduce the conventional Thai spelling rules to avoid extracting the invalid strings.

Previous work of [Nagao et al.1994] showed an effective way to construct a sorted file for the efficient calculation of n-gram data. But at the same time, a large amount of invalid strings are unexpectedly extracted. Later, [Ikehara et al.1995] has extended the sorted file to avoid repeating the count of contained substring of the string that has been counted. It has concluded itself to extract only the longest strings in order of frequency of occurrence. The result of extraction has been improved but the longest strings are always determined consecutively from left to right. In case of extracting an erroneous string, its error will directly propagate through the rest of input string. It is possible that a string with the invalid starting pattern will be extracted because the previous string with too long in character length has been extracted.

In the following sections, we firstly, describe the necessity why we make this statistical observation for extracting the open compounds from Thai text corpora. Then, the methodology of data preparation and open compound extraction are explained. Finally, we discuss the result of experiment on both large and small test corpora to investigate the effectiveness of our method.

2 Problem Description

It is not an easy task to identify a word in the text of the language which has no specific use of punctuation mark, especially the use of word break. Up to present, lexicographers' efforts have been limited by the size of corpus and computational facilities. Almost all the lexicon knowledge bases are created in reliance on human encoding. In recent years, a large amount of text corpora become available, and it is now becoming possible to conduct the experiment on text corpora in more rigorous way. We address the following problems in the way that they are able to be solved in statistical observations.

1. There is no any good evidence to support the registration of a word in a dictionary. In the traditional dictionary making, lexicographers have to rely on the citations collected by human readers from the limited text corpora. Many rare words rather than common words are found even in the standard dictionaries [Church and Hanks 1990]. This is the problem in making lexical entry list for dictionary construction.
2. It is hard to decide where to separate a string into words. It is also hard to list up words from a text though it is reported that the accuracy of recent word segmentation using dictionary and some heuristic methods

is higher than 95% [Virach 1993]. The accuracy depends mostly on word entries in the dictionary and the priority for selecting words when there are more than one solution for the word segmentation. This is the problem in assigning priority information for selection.

3 Word Extraction from Text Corpora

We used the window size of 4 to 32 for n-gram data accumulation. The value is arbitrary but it is reasonable to avoid collecting illegible strings.

3.1 Algorithm

Let's

|a| number of cluster[1] in string 'a',
n(a) is number of occurrence of string 'a', and
n(a+1) is number of occurrence of string 'a' with one cluster added.

In general, when the length of a string increases the chance that the string repeats itself in the next occurrence will decrease. Therefore,

$$n(a+1) \leq n(a). \tag{1}$$

Observing string 'a', the n(a+1) decreases significantly when 'a' is a frequently used string pattern in contrast to 'a+1'. We now can assume that 'a' is a rigid expression when it agrees the condition of

$$n(a+1) \ll n(a). \tag{2}$$

This means that 'a' is considerably a rigid expression that is frequently used in the text, and 'a+1' is just a string pattern that occasionally occurs in some context.

Since we count the strings that are generated by starting from an arbitrary position in the text, with only the above observation, we can only assume the right stopped position of a string to make a rigid expression. To identify the correct starting position of a string, we then apply the same observation of counting to the leftward increasing of a cluster to the string. Therefore, we have to include the direction of the string observation.

Assuming that,

+a is right observation of string 'a', and
-a is left observation of string 'a'. Then,
n(+a+1) is number of occurrence of string 'a' with one cluster added to the right of the string 'a', and
n(-a+1) is number of occurrence of string 'a' with one cluster added to the left of the string 'a'.

In the same way, we will obtain,

$$n(+a+1) \leq n(a), and \tag{3}$$

$$n(-a+1) \leq n(a). \tag{4}$$

'a' could be a rigid expression when it agrees the following counting conditions,

$$n(+a+1) \ll n(a), and \tag{5}$$

$$n(-a+1) \ll n(a). \tag{6}$$

3.2 Data preparation

Followings are the steps of creating n-gram text data according to the fundamental features of Thai text corpora. The results are shown in Table 1 and Table 2. In the table, "n" is number of the count and "d" is the difference value.

1. Tokenize the text at the position of space, tab or newline character.
2. Produce n-gram strings according to the Thai spelling rules. Only the strings that have possible break are generated and counted the occurrence through the whole text. For example, shifting string from 'a+6' to 'a+7' in the Table 1, the string at 'a+7' is 'กระทรวงการคลัง' but not 'กระทรวงการคล' though one character just after 'a+6' is 'ค'. According to Thai spelling rules, the character '์' never stands by itself. It needs both of an initial consonant and a final consonant. We call it a cluster.
3. Create both of rightward (Table 1) and leftward (Table 2) sorted strings. Number of the count of each string is the same but the strings are sorted reversely.
4. Calculate the difference of the count of adjoining strings in the sorted lists. Let's d(a) is the difference value of 'a', then

$$d(a) = n(a) - n(a+1). \tag{7}$$

[1]The smallest unit of character cluster according to the spelling rules.

The difference value (d) is generated separately for both of rightward and leftward sorted string tables.

The counts (n) in both Table 1 and Table 2 apparently support the condition (3) and (4).

String	Rightward sorted string	n	d
a	กระท	513	68
a+1	กระทร	445	0
a+2	กระทรว	445	0
a+3	กระทรวง	445	142
a+4	กระทรวงกา	303	0
a+5	กระทรวงการ	303	22
a+6	กระทรวงการค	281	0
a+7	กระทรวงการคลัง	281	274
a+8	กระทรวงการคลังกำ	7	0

Table 1: Sample of the Count of Rightward Sorted String Table

String	Leftward sorted string	n	d
-b	การกระทรวง	172	0
-b+1	ว่าการกระทรวง	172	0
-b+2	รีว่าการกระทรวง	172	42
-b+3	ตรีว่าการกระทรวง	130	9
-b+4	นตรีว่าการกระทรวง	121	0
-b+5	มนตรีว่าการกระทรวง	121	7
-b+6	รัฐมนตรีว่าการกระทรวง	114	107
-b+7	งรัฐมนตรีว่าการกระทรวง	7	0

Table 2: Sample of the Count of Leftward Sorted String Table

3.3 Extraction

3.3.1 Competitive selection

According to the condition (5) the string 'a' in Table 3 is considerably an open compound because the difference of the count between n(a) and n(a+1) is as high as 450. It looks reasonable to pick up 'กระท' to be an open compound because its occurrence is very high comparing to the next generated strings. But, 'กระท' is an illegible string and cannot be used individually in general text. Observing the same string 'a' in Table 1, the difference of the count between n(a) and n(a+1) is only 68. It is not comparably high enough to be selected. Therefore, we have to figure out the minimum value of the difference when there are more than one branch extended from a string.

String	Rightward sorted string	n	d
a	กระท	513	450
a+1	กระทบ	63	22
a+2	กระทบก	41	0
a+3	กระทบกระ	41	0
a+4	กระทบกระเทือ	41	0
a+5	กระทบกระเทือน	41	25
a+6	กระทบกระเทือนต่อ	16	11
a+7	กระทบกระเทือนต่อกา	5	0

Table 3: Another Sample of the Count of Rightward Sorted String Table

3.3.2 Unified selection

In Figure 1, we obtain the string 'รกระทรวงการคลัง' by observing the significantly change of the count just before the next string 'รกระทรวงการคลังรัก'. Reversely, we observe the count of string 'รกระทรวงการคลังรัก' when it is extended leftward, as shown in Figure 2.

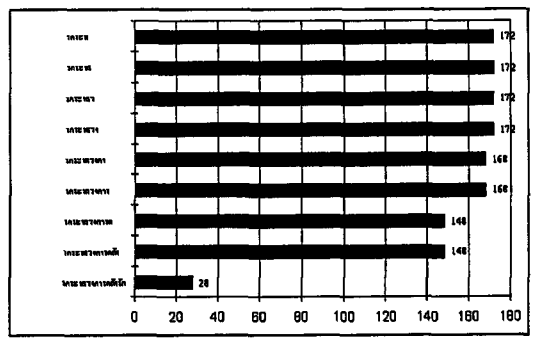

Figure 1: Rightward Sorted Strings Starting from an Arbitrary String

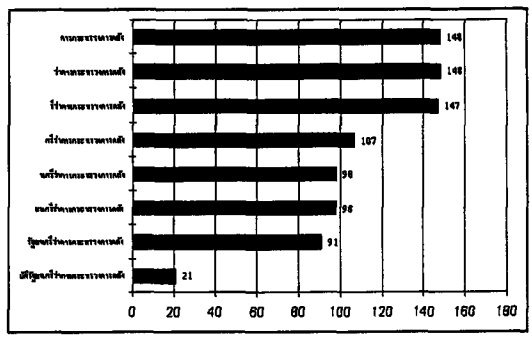

Figure 2: Leftward Sorted Strings Starting from an Arbitrary String

By unifying the results of both ways of the observation, we finally, obtain the string 'รัฐมนตรีว่าการกระทรวงการคลัง'.

4 The Experimental Results

We have applied our method to actual Thai text corpora without any preprocessing.

4.1 Natural language data

We selected 'Thai Revenue Code (1995)' which is as large as 705,513 bytes and a book of the 'Convention for Avoidance of Double Taxation between Thailand and Japan' which has a smaller size of 40,401 bytes. The purpose is to show that our method is effective in a wide range of file size.

4.2 Result of extraction

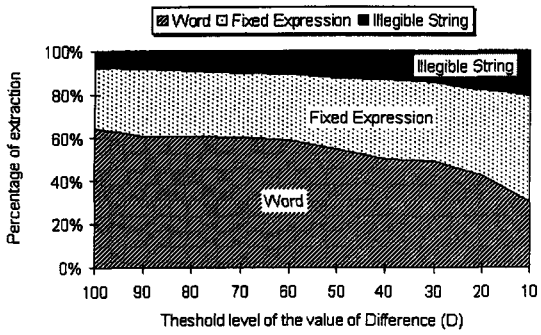

Figure 3: Result of Extraction of Thai Revenue Code (705,513 bytes)

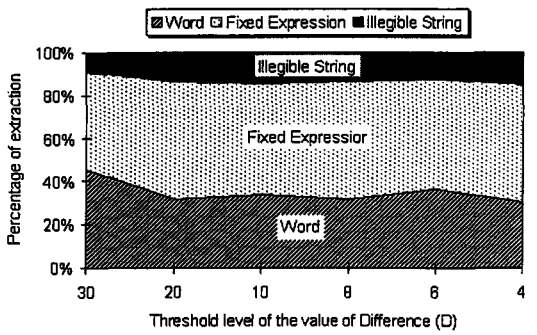

Figure 4: Result of Extraction of Convention for Thailand-Japan (40,401 bytes)

The results of extraction examined in both large and small file sizes are very satisfactory. Very few illegible strings are extracted though the threshold of the difference value is set to be as low as 10 in Figure 3 and 4 in Figure 4. The suitable threshold of difference value varies with the size of text corpus file. To obtain the more meaningful strings from the large file, we have to set up a relatively high level for the threshold of extraction. One of the advantages of our method is that we can make a trade-off between the quantity and the quality of the extracted strings. In case of Figure 3, to limit the amount of illegible strings not to exceed 90% of the total extracted strings, we set the threshold to 20. As a result, we obtained 210 words, 196 fixed expressions and together with only 89 illegible strings. Furthermore, we found that within the 210 items which are exactly words in the actual text, there are 123 items that are not found in a Thai standard dictionary.

5 Conclusion

This paper has shown the algorithm for data preparation and open compound extraction. The *competitive selection* and *unified selection* of rightward and leftward sorted strings play an important role in improving accuracy of the extraction. In the experiment, we have applied Thai spelling rules to restrict the search path for string count. Some types of the spelling irregularity can be avoided in this process. The result of the experiment is satisfied to serve lexicographers conducting the inspection of rigid expression of open compounds.

References

[Church and Hanks 1990] Church, K. W. and Hanks, P. 1990. Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, Vol.16, No.1, pages 22-29.

[Ikehara et al.1995] Ikehara, S., Shirai S. and Kawaoka, T. 1995. Automatic Extraction of Uninterrupted Collocations by n-gram Statistics. *Proceedings of The first Annual Meeting of The Association for Natural Language Processing*, pages 313-316 (in Japanese).

[Nagao et al.1994] Nagao, M. and Mori, S. 1994. A New Method of N-gram Statistics for Large Number of n and Automatic Extraction of Words and Phrases from Large Text Data of Japanese. *Proceedings of COLING 94*, Vol.1, pages 611-615.

[Smadja and McKeown 1990] Smadja, F. A. and McKeown, K. R. 1990. Automatically Extracting and Representing Collocations for Language Generation. *Proceedings of ACL-90*, pages 252-259.

[Virach 1993] Sornlertlamvanich, Virach. 1993. Word Segmentation for Thai in Machine Translation System. *Machine Translation*, National Electronics and Computer Technology Center, (in Thai).