AUTOMATIC CORPUS-BASED THAI WORD EXTRACTION

TANAPONG POTIPITI, VIRACH SORNLERTLAMVANICH, and
THATSANEE CHAROENPORN
National Electronics and Computer Technology Center (NECTEC), Thailand

The Thai language is infamous in its ambiguity. One of its important ambiguities is that there is no explicit word boundary, or in other words there is no explicit definition what words are. Traditional methods on defining words, which depend on human judgement, base on unclear criteria or procedures, and have several limitations. This paper describes an automatic statistical method Thai word extraction from plain Thai text, by employing suffix-array, mutual-information and entropy techniques. Experimental results are quite impressive; our algorithm can extract 428 acceptable words from 1 MB of plain Thai text corpus and the accuracy of extraction is about 85 per cent in both training and test corpus.

Key words: Thai word extraction, part-of-speech tagging

1. INTRODUCTION

1.1 WORD EXTRACTION AND THAI NATURAL LANGUAGE PROCESSING

In the non-word-boundary languages such as Thai, word entries are difficult to define. This causes a lot of problems in Thai language processing such as word segmentation, information retrieval, machine translation, etc. Unless there is regularity in defining word entries, the Thai language processing cannot be achieved. The existing Thai language processing employs the manually created dictionaries to determine what words are. These manual and static dictionaries have a lot of drawbacks. First, it cannot deal with words that are not in the dictionaries. Second, because these dictionaries are manually created, it will never cover all words that occur in real corpora. This paper, therefore, proposes an automatic word-extraction algorithm, which hopefully can overcome this Thai language-processing barrier.

1.2 PROBLEMS OF THAI WORD IDENTIFICATION

An essential and non-trivial task for languages that exhibit inexplicit word boundary such as Thai, Japanese, and other Asian languages undoubtedly is identifying word boundary. Word, generally, means a unit of expression which has universal intuitive recognition by native speakers. Linguistically, word can be considered as the most stable of unit which has little potential to rearrangement and is uninterrupted as well. “Uninterrupted” here attracts our lexical knowledge bases so much. Because there are a lot of uninterrupted sequences of words functioning as a single constituent of a sentence. These uninterrupted strings, of course are not the lexical entries in a dictionary, but each occurs in a very high frequency. The way to point out whether they are words or not is still not distinct even by native speakers. Actually, it depends on individual judgement. Computationally, it is also difficult to decide where to separate a string into words. Even it is reported that the accuracy of recent word segmentation using dictionary and some heuristic methods is in a high level. Currently, lexicographers can make use of large corpora and can conduct any experiment on corpora. We, therefore, introduce here a new automatic statistical method for extracting and identifying Thai words.

2 PREVIOUS WORKS

Reviewing the previous works on Thai word extraction, we found only the work of Sornlertlamvanich and Tanaka (1996). They employed the frequency of the sorted n-gram character data to extract Thai open compounds; the strings that experienced a significant change of occurrences when their length are extended were extracted as open compounds. This algorithm reports about 90 per cent accuracy of Thai-open-compound extraction. However, the algorithm had to limit the range of n-gram to 4-20 gram for the computational reason. This causes limitation in the size of corpus and efficiency in the extraction.

The other works we can find are for the Japanese language. Nagao et al. (1994) has provided an effective method to construct a sorted file that facilitates the calculation of n-gram data. But their algorithm did not yield a satisfactory accuracy; there were many invalid strings extracted. The following work (Ikehara et al., 1995) improved the sorted file to avoid repeating in counting strings. The extraction result was better, but the determination of the longest strings is always made consecutively from left to right. If an erroneous string is extracted, its errors will propagate through the rest of the input string.

3 OUR WORD-EXTRACTION APPROACH

3.1 BASIC CONCEPTS

We employ mutual-information and entropy statistics as criteria to accept a string as a word. In this section, these concepts will be introduced.

3.1.1 *Left Mutual Information and Right Mutual Information*

Mutual information (Church et al. 1991) of random variable a and b is the ratio of probability that a and b co-occur, to the independent probability that a and b co-occur. High mutual information indicates that a and b co-occur more than expected by chance. Our algorithm employs left and right mutual information as criteria in word extraction procedure. Left mutual information (*Lm*), and right mutual information (*Rm*) of string $x\mathbf{y}z$ are defined as:

$$Lm(x\mathbf{y}z) = \frac{p(x\mathbf{y}z)}{p(x)\,p(\mathbf{y}z)},$$

$$Rm(x\mathbf{y}z) = \frac{p(x\mathbf{y}z)}{p(x\mathbf{y})\,p(z)},$$

where

x is the rightmost character or string $x\mathbf{y}z$
$\mathbf{y}$ is the middle substring of $x\mathbf{y}z$
z is the leftmost character or string $x\mathbf{y}z$
$p(\ )$ is the probability function.

If $x\mathbf{y}z$ is a word, both $Lm(x\mathbf{y}z)$ and $Rm(x\mathbf{y}z)$ should be high. On the contrary, if $x\mathbf{y}z$ is a string that is not a word but consists of words and characters, its left or right mutual information will be low. For example, "กปรากฏ" which consists of character "ก" + word "ปรากฏ" will have low left mutual information.

3.1.2 *Left Entropy and Right Entropy*

Entropy (Shannon 1948) is the information measuring disorder of variables. The left and right entropy is exploited as criteria in our word extraction. Left entropy (*Le*), and right entropy (*Re*) of string $\mathbf{y}$ are defined as:

$$Le(\boldsymbol{y}) = -\sum_{\text{all } x \in \boldsymbol{A}} p(x\boldsymbol{y} \mid \boldsymbol{y}) \log_2 p(x\boldsymbol{y} \mid \boldsymbol{y}) ,$$

$$Re(\boldsymbol{y}) = -\sum_{\text{all } z \in \boldsymbol{A}} p(\boldsymbol{y}z \mid \boldsymbol{y}) \log_2 p(\boldsymbol{y}z \mid \boldsymbol{y}) ,$$

where

y is the considered string

A is the set of all alphabets

x, z is any alphabet in ***A***.

if ***y*** is a word, the alphabets that come before and after ***y*** should have varieties or high entropy. If ***y*** is not a complete word, its left or right entropy will be low. For example, "ปราก" is not a word but a substring of word "ปรากฏ", thus the alphabet right adjacent to "ปราก" should be only 'ฏ' and the right entropy of "ปราก" is low.

3.2 THE PROCEDURES

Briefly, our algorithm consists of these main parts. Firstly, all strings are extracted from the corpora. Then the statistics above are calculated for each string. Finally, these statistics are used to determine whether the considering string is a word or not.

3.2.1 Substring Extraction

First, we assume that a Thai word generally consists of less than 20 characters. Then we extract all substrings that have length less than 20 characters from the corpora. To extract all these substrings in linear time, Yamamoto and Church (1998)'s suffix-array algorithm is applied (for the naïve algorithm, it requires polynomial time for this task which is computationally impossible).

3.2.2 Computing Left Mutual Information, Right Mutual Information, Left Entropy and Right Entropy for Each Substring

These four statistics are calculated for each substring and used as criteria to accept the considered substring to be a word.

3.2.3 Substring-to-Word Criteria

Substring ***s*** will be accepted as a word if all these conditions are satisfied:

1) $Lm(\boldsymbol{s}) > Lm_0$,

2) $Rm(\boldsymbol{s}) > Rm_0$,

3) $Le(\boldsymbol{s}) > Le_0$,

4) $Re(\boldsymbol{s}) > Re_0$,

5) The number of occurrences of ***s*** in text is more than O_0 .

The first four conditions filter out the substrings that could not possibly to be candidates for words. The table below illustrates how these four conditions work.

No.	Substring	*Lm*	*Rm*	*Le*	*Re*
1	คโนโลยี	6034	8040	***0.13***	3.21
2	เทคโนโล	6203	5417	4.57	***0.12***
3	กเทคโนโลยี	***0.53***	1.48	2.48	3.67
4	เทคโนโลยีก	4609	***1.37***	3.74	2.44
5	เทคโนโลยี	6178	8337	4.57	3.21

Table 3-1: Mutual information and entropy for substring

In the five substrings above, only substring no.5: "เทคโนโลยี"(technology) is a word. We can see that the other substrings will be filtered out by one of these conditions. Substring no.1 is filtered out by condition 1. Substring no.2 is filtered out by condition 2 and so on. The substring that satisfies the first four conditions is selected as a word candidate.

The fifth condition is simply from the heuristic that a word should be a substring that is frequently used in the text.

3.2.4 Filtering the Substrings from 3.2.3 by Thai Spelling Rules

In this phase, some of Thai spelling rules are applied to re-check that these substrings can be a word in Thai or not.

4 PRELIMINARY RESULTS

4.1 HOW TO SET THE THRESHOLDS

To find the suitable thresholds, we create a training corpus in which all words are listed. By averaging the statistics described above, we now get a based-line threshold. Then, this threshold is applied to the test corpus to be evaluated. From the training corpus, we get the thresholds value as follows.

1) $Lm_0 = 6170$,
2) $Rm_0 = 4550$,
3) $Le_0 = 1.37$,
4) $Re_0 = 1.02$,
5) $Oc_0 = 7$.

4.2 THE RESULTS

We have applied the threshold chosen above for word extraction in the training corpus and another corpus, test corpus. The experimental results are as follows.

	All Substring Extracted	Words	Compounds	Errors
Training Corpus	495 (100%)	367 (74.1%)	72 (14.5%)	56 (11.4%)
Test Corpus	507 (100%)	363 (71.6%)	69 (13.6%)	75 (14.8%)

Table 4-1: The accuracy of words extracted from the algorithm

Next, we compared the correct words extracted and the words in Thai Royal Institute Dictionary (RID) that were found in the corpus. There were 4485 and 4682 words in RID that occurs in the

training and test corpus respectively. About half of the correct words extracted was in RID; and also about half of the correct words and compounds extracted was not in RID in both training and test corpus.

	Correct words + compounds extracted by the algorithm	Words extracted which is in RID	Correct extracted words which is not in RID	Words in RID which occur in the corpus
Training Corpus	439	209	220	4485
Test Corpus	431	224	207	4682

Table 4-2: The comparison between the words extracted and the words in Thai Royal Institute Dictionary.

4.3 TYPE OF ERRORS IN EXTRACTION

The error strings extracted can be categorized into two groups:

1) Non-word substrings: are the substrings that contain some parts that are not words and have no meaning such as "พฤติ" and "ครูก".
2) Incomplete meaning phrases: are the substrings that consist of words but their meanings are not complete such as "แห่งประทศไทย" (of Thailand) and "เขาจะ" (he will).

	All Errors	Type1	Type2
Training Corpus	72 (100%)	21 (29.1%)	51 (70.9%)
Test Corpus	69 (100%)	23 (33.3%)	46 (66.7%)

Table 4-3: Errors in each type

4.4 THE FURTHER STEPS TO REDUCE THESE ERRORS

If we set the value of Lm_0, Rm_0, Le_0, Re_0 and O_0 higher, Type1 errors will be diminished. However, there are always trade-off between precision and recall; the higher the thresholds are set, the less words are extracted. Most of Type2 errors contain frequently used prepositional or auxiliary words such as "ใน" (in), "แห่ง" (of) and "จะ" (will). These errors should be filtered out by some simple rules.

5 CONCLUSIONS AND FURTHER STUDIES

In this paper, we have found that the mutual information and entropy are very useful features for Thai word extraction. Applying these features to a simple AND-rule, we have got quite good results and a sound beginning step to proceed. To get more words extracted, larger corpora should be used. Furthermore, in order to employ these features more effectively, rather than a simple AND-rule, a machine learning algorithm should be employed for this task.

REFERENCES

Church, K.,W., Robert L. and Mark L.Y. (1991) A Status Report on ACL/DCL, *Proceedings of 7th Annual Conference of the UW Centre New OED and Text Research: Using Corpora*: 84-91

Ikehara, S., Shirai, and Kawaoka, T. (1995) Automatic Extraction of Uninterrupted Collocations by n-gram Statistics, *Proceeding of The first Annual Meeting of the Association for Natural Language Processing*: 313-316 (in Japanese)

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, *Proceeding of COLING 94*, Vol. 1: 611-615

Shannon, C.,E. (1948) A Mathematical Theory of Communication, *Bell System Technical Journal* 27:379-423

Sornlertlamvanich, V. and Tanaka, H. The Automatic Extraction of Open Compounds from Text, *Proceeding of COLING 96 Vol. 2*: 1143-1146

Yamamoto, M. and Church, K.(1998) Using Suffix Arrays to Compare Term Frequency and Document Frequency for All Substrings in Corpus. *Proceeding of Sixth Workshop on Very Large Corpora*