Feature-based Thai Unknown Word Boundary Identification Using Winnow

Paisarn Charoenpornsawat, Boonserm Kijsirikul
Department of Computer Engineering Chulalongkorn University
Phayathai Road, Bangkok, Thailand.
Phone: (662) 218-6976, Fax: (662) 218-6955
pcharoen@notes.nectec.or.th, fengbks@chulkn.chula.ac.th
Surapant Meknavin
National Electronics and Computer Technology Center
73/1 Rama VI Road, Rajthevi, Bangkok, Thailand.
Phone: (662) 644-8150 ext 725, Fax: (662) 644-8137
surapan@nectec.or.th

Abstract

This paper addresses the problem of Thai unknown word boundary identification. Unknown words are becoming main problem in many tasks of Natural Language Processing such as word segmentation, information retrieval and part of speech tagging etc.. In Thai, as words are written consecutively without delimiters, finding unknown word boundary is difficult. We proposed a feature-based approach to identify Thai unknown word boundary. A feature can be anything that tests for specific information in context around the target unknown words. To automatically extract features from a training corpus, we used a machine learning algorithm, namely Winnow.

1. Introduction

Unknown words are the main problem in tasks of Natural Language Processing such as word segmentation, information retrieval and part of speech tagging etc. The problem of unknown word is more serious in languages which have no explicit word boundary such as Thai, Japanese, etc. because its occurrence may make surrounding words incorrectly recognized and itself may be wrongly identified. Unknown words can be categorized into two types; *explicit unknown word* and *hidden unknown word* [6]. An explicit unknown word is an unknown word of which no any substrings are known words in a dictionary. A hidden unknown word, on the other hand, an unknown word of which some substrings are known words. In this paper we focus on unknown words which are proper nouns because most unknown words are proper nouns.

To solve the problem of identifying Thai unknown word boundary, we propose a feature-based approach. Although feature-based approaches[2,5] have already been applied in several fields of Natural Language Processing, they have not been considered in unknown word boundary identification. A feature can be anything that tests for specific information in the context around the target unknown word, such as *context words* and *collocations*. The idea is to learn several sources of features that characterize the contexts in which each unknown word tends to occur. We then combine these features to identify unknown word boundary by selecting the segmentation of known and unknown words that yields the most probable sequence of words for a given context. The Winnow algorithm is employed in our task for extracting such features.

2. Problem of Unknown Word Boundary

Various forms of unknown words are caused by the combination of known and unknown strings. For example, the various forms of unknown words are shown in Figure 1.

UKU *UKK* *KUKK* *UK* *KU* *KKK*

Figure 1 :Unknown word forms

In Figure1, *U* is an unknown string and *K* is a known string. According to unknown word forms, an unknown word is classified into two main categories.

1. **Explicit unknown word**

   An explicit unknown word is an unknown word of which no any substrings are in a dictionary. Examples of explicit unknown word are listed below.

   กทม.

   โลตัส

   สุนีย์.

2. **Hidden unknown word**

   A Hidden unknown word is composed of words in the dictionary. A hidden unknown word is classified into 2 subcategories.

   2.1 Partially hidden unknown word

   A partially hidden unknown word is composed of known words and unknown strings for example,

   สุมานี → สุ **มา** นี

   คธาพงศ์ → คธา **พงศ์**

   ไมโครซอฟต์ → ไม **โค** ร **ซอ** ฟต์

   where "→" means "is composed of",and bold strings are known words.

   2.2 Fully hidden unknown word

A fully hidden unknown word is composed of only known words such as

สมชาย →สม ชาย

กนกพร →กนก พร.

According to the characteristic of unknown words, we propose the methods to generate candidates for the explicit and hidden unknown words below.

2.1 Generating Candidates of Unknown Words:

We propose two heuristics to generate candidates for both types of unknown words. After all candidates are generated, the best candidate will be selected by Winnow. Winnow will be described in the next section.

Handling Explicit & Partially Hidden Unknown Words:

In the case that a string does not exist in the dictionary, candidates of unknown words will be created by merging words around that unknown string and the string itself into a new string. All combinations of +/- K words around the unknown string are used to generate candidates. We can define the equation for creating unknown word candidates as in Figure 2.

In case that there are many unknown strings, the nearby unknown strings will be grouped into a single unknown string for using as a candidate if they are separated by a word having less than three characters.

Sentence = $w_1 w_2 ... w_a U w_b ... w_n$

where $w_i \in$ Dictionary, $U \notin$ Dictionary

n = number of words in the sentence.

$UNK = \{ \alpha U \beta \mid \alpha \in A, \beta \in B \}$

where UNK = set of unknown word candidates.

$A = \{ w_{a-i,a}, i \in [0, K] \} \cup \{\varepsilon\}$

$B = \{ w_{b,b+i}, i \in [0, K] \} \cup \{\varepsilon\}$

$w_{i,j} = w_i \ldots w_j \quad : i < j$

ε = null string., K = constant value

Figure 2: Equation for generating explicit and partially hidden unknown word candidates

Handling Fully Hidden Unknown Words:

On the other hand, if all words are in the dictionary, it is more difficult to detect the unknown words.

Let Sentence = $w_1 w_2 \ldots w_n$ be the input sentence, w_i be a word in the sentence, and t_i be the part of speech of the word w_i. The word that will be selected as an unknown word candidate is:

Sentence = $w_1 w_2 ... w_a ... w_{n-1} w_n$

where $w_i \in$ Dictionary

n = number of words in the sentence.

w_a is the word that has probabiliy less than threshold

$UNK = \{ \alpha W \beta \mid \alpha \in A, \beta \in B \}$

where UNK = set of unknown word candidates.

$A = \{ w_{a-i,a-1}, i \in [0, K] \} \cup \{\varepsilon\}$

$B = \{ w_{a+1,a+i}, i \in [0, K] \} \cup \{\varepsilon\}$

$w_{i,j} = w_i \ldots w_j \quad : i < j$

$W = w_a \quad : P(w_a \mid t_a) < \text{threshold}$ or

$W \in \{w_{a-2}.w_{a-1}, w_a\} \quad : P(t_a \mid t_{a-1}, t_{a-2}) < \text{threshold}$

ε = null string. K = constant value

Figure 3: Equation for generating fully hidden unknown word candidates

- the word that has P($w_i|t_i$) less than a threshold, or
- the word that has P($t_i|t_{i-1},t_{i-2}$) less than a threshold.

In case that P($w_i|t_i$) is less than a threshold the w_i will be considered as an unknown word. In case that the probability P($t_i|t_{i-1},t_{i-2}$) of w_i is less than a threshold, not only w_i but also w_{i-1} and w_{i-2} must be considered as unknown words because the less-than-threshold probability of w_i may come from w_{i-1} or w_{i-2}. We can define the equation of creating unknown words as in Figure 3.

3. Winnow Algorithm

Winnow algorithm used in our experiment is the algorithm described in [1]. Winnow is a neuron-like network where several nodes are connected to a target node. Each node called specialist looks at a particular value of an attribute of the target concept, and will vote for a value of the target concept based on its specialty; i.e. based on a value of the attribute it examines. The global algorithm will then decide on weighted-majority votes receiving from those specialists. The pair of (attribute=value) that a specialist examines is a candidate of features we are trying to extract. The global algorithm updates the weight of any specialist based on the vote of that specialist. The weight of any specialist is initialized to 1. In case that the global algorithm predicts incorrectly, the weight of the specialist that predicts incorrectly is halved and the weight of the specialist that predicts correctly is multiplied by 3/2. The weight of a specialist is halved when it makes a mistake even if the global algorithm predicts correctly.

In our experiments, to train the unknown words, we select all sentences containing proper nouns and consider those proper nouns as unknown words. The context around the proper noun is used to form features in identifying that the proper noun is an unknown word. Similar process is done for known words which are all

words not being proper nouns. The features used are the context words and collocations. Context words are used to test for the presence of a particular word within +/- 10 words from the target word. Collocations are a pattern of up to 2 contiguous words and/or part-of-speech tags around the target word. After Winnow is trained, the resulting network is used to rank the score of candidates. The best score candidate will be selected as the answer.

4. An overview of the system

Our algorithm for identifying unknown words consists of four steps as follows:

1. **Word Segmentation**

For each input sentence, probabilistic trigram model [4] is applied to separate the sentence into words and assign their parts of speech, and N-best segmented sentences are then selected as candidates. The probabilistic trigram model, that generates the *N* highest probable sentences, can be described formally as following:

Let $C = c_1c_2...c_m$ *be an input character string,* $W_i = w_1w_2..w_n$ *be a possible word segmentation, and* $T_i = t_1t_2..t_n$ *be a sequence of parts of speech. Find* $W_1, W_2...W_N$ *which are the N-highest probability of sequence of words. We can compute* $P(W_i)$ *in the following fashion:*

$$P(W_i) = \sum_T P(W_i, T_i) = \sum_T \prod_i P(t_i \mid t_{i-1}, t_{i-2}) * P(w_i \mid t_i) \quad (1)$$

where $P(t_i|t_{i-1}\ t_{i-2})$ *and* $P(w_i|t_i)$ *are computed from the corpus.*

For example, let C = "ฉันไปเที่ยวน้ำตกทีลอซูกับเพื่อน". The results of our word segmentation algorithm of which the format is $w_1/t_1\ w_2/t_2\ ...\ w_n/t_n$ are shown as follows:

I. ฉัน/t_1 ไป/t_2 เที่ยว/t_3 น้ำตก/t_4 ที/t_5 **ลอซู**/t_6 กับ/t_7 เพื่อน/t_8

II. ฉัน/t_1 ไป/t_2 เที่ยว/t_3 น้ำ/t_4 ตก/t_5 ที/t_6 **ลอซู**/t_7 กับ/t_8 เพื่อน/t_9

The word **ลอซู** is an unknown string and t_n is an appropriate part-of-speech tag of each word.

2. **Generating Candidates of Unknown Words**

From the result of step 1, generate all candidates of unknown words by the explicit and hidden unknown word heuristics described in Section 2.

For example, the sentence (I) from step 1, we found the seventh word, ลอซู, is an unknown string. Therefore we use the method for handling explicit & partially hidden unknown word which is explained in section 2. Unknown word candidates are ลอซู, ทีลอซู, น้ำตกทีลอซู, ลอซูกับ, ลอซูกับเพื่อน, ทีลอซูกับ, ทีลอซูกับเพื่อน, น้ำตกทีลอซูกับ and น้ำตกทีลอซูกับเพื่อน where *K*, *U* ,*A* and *B* in Figure 1 are 2, ลอซู , {ε, ที, น้ำตกที} and {ε, กับ, กับเพื่อน} respectively. All the sentences will be processed in the same way. After that we get all the unknown word candidates for every sentence which are generated by the word segmentation.

3. **Tagging Part of Speech**

The new sentences will be formed by combining candidates that are obtained from step 2 with the rest of words in the old sentence. The part of speech of words in each sentence will be reassigned by the trigram tagger. The unknown tokens are assumed to be the proper noun. Part of speech trigram can be defined as the following:

Let W be a sequence of words $w_1..w_n$ *and* T_i *be a sequence of part-of-speech tags* $t_1..t_n$. *Find* τ *that maximizes* $P(T_i \mid W)$*:*

$$\tau = \arg\max_{T_i} P(T_i \mid W) = \arg\max_{T_i} P(t_i \mid t_{i-1}, t_{i-2}) * P(w_i \mid t_i) \quad (2)$$

For example, in sentence (I) from step 1, unknown word candidates are created by the procedure in step 2. In this step, the new sentences are shown as follows:

III. ฉัน/t_1 ไป/t_2 เที่ยว/t_3 น้ำตก/t_4 ที/t_5 **ลอซู**/*NPRP* กับ/t_7 เพื่อน/t_8

IV. ฉัน/t_1 ไป/t_2 เที่ยว/t_3 น้ำตก/t_4 **ทีลอซู**/*NPRP* กับ/t_6 เพื่อน/t_9

V. ฉัน/t_1 ไป/t_2 เที่ยว/t_3 **น้ำตกทีลอซู**/*NPRP* กับ/t_5 เพื่อน/t_6

VI. ฉัน/t_1 ไป/t_2 เที่ยว/t_3 น้ำตก/t_4 ที/t_5 **ลอซูกับ**/*NPRP* เพื่อน/t_7

VII. ฉัน/t_1 ไป/t_2 เที่ยว/t_3 น้ำตก/t_4 ที/t_5 **ลอซูกับเพื่อน**/*NPRP*

	Explicit & Partially Hidden Unknown Words		Fully Hidden Unknown Words	
	Training Set	Test Set	Training Set	Test Set
Detect	100.00 %	100.00 %	87.82 %	83.25 %
Winnow	95.26 %	92.75 %	80.68 %	74.26 %

Table 1: The results of our approach.

VIII. ฉัน/t_1 ไป/t_2 เที่ยว/t_3 น้ำตก/t_4 **ทีลอซูกับ**/*NPRP* เพื่อน/t_6

IX. ฉัน/t_1 ไป/t_2 เที่ยว/t_3 น้ำตก/t_4 **ทีลอซูกับเพื่อน**/*NPRP*

X. ฉัน/t_1 ไป/t_2 เที่ยว/t_3 **น้ำตกทีลอซูกับ**/*NPRP* เพื่อน/t_9

XI. ฉัน/t_1 ไป/t_2 เที่ยว/t_3 **น้ำตกทีลอซูกับเพื่อน**/*NPRP*

where t_n is a part-of-speech tag and *NPRP* is a proper noun tag. The sentence (II) can be processed in the similar manner.

4. Predicting by Winnow

Sentences from step 3 will be sent to Winnow to rank the sentences, and the sentence with the highest score will be selected as the answer.

For the example in step 3, Winnow will select the best- score sentence that is the sentence (IV); ฉัน ไป เที่ยว น้ำตก **ทีลอซู** กับ เพื่อน, will be selected as the answer.

5. Preliminary Result

5,000 sentence corpus were used in our experiment. Every sentence is manually separated into words and their parts of speech are tagged by linguists. The resulting corpus is divided into 2 parts; the first part about 80% of corpus is utilized for training and the rest is employed for testing. To measure the performance of our approach, we classify them into two groups. The first group is an explicit and partially hidden unknown words and the other is fully hidden unknown words. First, we count the number of unknown words in each group. Next we calculate the percentage of unknown words which can be detected by our approach. Finally, we calculate the percentage of selecting the correct answers by Winnow. The result of our experiment is shown in Table 1.

6. Conclusion

The experimental result shows that explicit and partially hidden unknown words can be completely detected but fully hidden unknown words can be detected about 83%. Context words and collocations can be effectively used to find the boundaries of unknown words, and Winnow is an efficient algorithm to apply in our task. In future work, all types of unknown words including misspelling words will be investigated and we will find the more efficient method to detect fully hidden unknown words.

Acknowledgement

We would like to thank Software and Language Engineering Laboratory (SLL) for providing the Orchid Corpus. Many thanks to Miss Virongrong Tesprasit for tagging unknown words. This work was supported by the Thai Government Research Fund.

References

[1] Blum, A. 1997. Empirial Support for Winnow and Weighted-Majority Algorithm: Results on a Calendar Scheduling Domain, Machine Learning, 26:5-23.

[2] Golding, A. R. & Roth, D. 1996. Applying Winnow to Context-Sensitive Spelling Correction. In Lorenza Saitta, editor, Machine Learning: Procs. Of the 13th International Conference, Bari, Italy.

[3] Littlestone, N. 1988. Learning Quickly when Irrelevant Attributes Bound: A New Linear-Threshold Algorithm. Machine Learning, 2:285-318.

[4] Meknavin, S., Charoenpornsawat P. & Kijsirikul, B. 1997. Feature-based Thai Word Segmentation. In proceeding of NLPRS'97.

[5] Kawtrakul, A., Kumtanode, S., Jamjanya, T. & Jewriyavech C. 1995. A Lexicon Model for Writing Production Assistant System. In Proceedings of the Symposium on Natural Language Processing in Thailand'95.

[6] Kawtrakul A., Thumkanon C., Poovorawan, Y., Varasrai P. & Suktarachan M. 1997. Automatic Thai Unknown Word Recognition. In proceeding of NLPRS'97.

[7] Rarurom, S. 1991. Dictionary-based Thai Word Seperation. Senior Project Report. (in Thai)

[8] Sornlertlamvanich, V.,Charoenporn, T. & Isahara, H. 1997. ORCHID: Thai Part-Of-Speech Tagged Corpus. In Technical Report Orchid Corpus.