Feature-based Proper Name Identification in Thai

Paisarn Charoenpornsawat, Boonserm Kijsirikul
Department of Computer Engineering, Chulalongkorn University
Phayathai Road, Bangkok, Thailand 10330

Surapant Meknavin
National Electronics and Computer Technology Center
73/1 Rama VI Road, Bangkok, Thailand 10400

Abstract

This paper addresses the problem of Thai proper name identification. To identify Thai proper names, we have to solve two problems; (1) Word boundary problem (Thai has no explicit word boundary delimiter), (2) Category problem (Find the right categories such as person, location or organization names). We propose a feature-based approach for Thai proper name identification. A feature can be anything that tests for specific information in the context around the word considered, such as context words and collocations. To automatically extract such features from a training corpus, we employ a learning algorithm, namely Winnow. Our approach works successfully with about 92% accuracy.

1. Introduction

Thai proper name identification is necessary in many tasks of Natural Language Processing such as machine translation, information retrieval etc. In this paper, we will not only locate Thai proper names, but also determine their categories, namely person names, organization names or location names. To locate proper names in Thai, we solve both word boundary problem and category problem. We propose a feature-based approach to solve both problems at the same time. Although feature-based approaches have already been applied in several fields of Natural Language Processing, they have not been employed in proper name identification. A feature can be anything that tests for specific information in the context around the target proper name, such as *context words* and *collocations*. The idea is to learn several sources of features that characterize the contexts in which each proper name tends to occur. The Winnow algorithm is employed in our task for extracting such features. We then combine these features to identify proper name boundary and proper name category by selecting the segmentation of known and unknown words that yields the most probable sequence of words for a given context.

2. Characteristics of Thai proper names

Thai proper names can be classified into many types, but we focus on three main types which are person, organization and location names. Various forms of proper names are formed by the combination of known and unknown strings. For example, (see Table 1.)

Thai proper name form	Thai proper name examples
I. Only Unknown String	แฮนเว่น ◆ แฮรี่ ◆ สุนีย์ ◆ นคเรศ
II. Unknown Strings + Known Words	สุมานี ◆ คธาพงศ์ ◆ ไมโครซอฟต์
III. Only Known Words	***สมชาย*** ◆ **กนกพร** ◆ ***รุ่งศักดิ์***

Table 1. Example of Thai proper names

In Table 1, The strings that are printed in bold or bold italic characters are known words and the others are unknown strings. Then we classify proper names into 2 types; (1) *Proper name type I* that is composed of unknown strings (I, II in Table 1), and (2*) Proper name type II* which is composed of only known words (III in Table 1). To find proper names of both types, we propose the methods to generate candidates for proper names as follows.

2.1 Generating candidates of proper names:

We propose two heuristics to generate candidates for both types of proper names. After all candidates are generated, the best candidate and its category will be selected by Winnow algorithm. Winnow will be described in the next section.

Proper name type I heuristic:

In case that a string does not exist in the dictionary, candidates of proper names will be created by merging words around that unknown string and the string itself into a new string. All combinations within +/- *K* words around the unknown strings are generated as candidates. We can define the equation for creating proper name candidates in Figure 1.

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

$$Sentence = w_1 w_2 ... w_a U w_b ... w_n \qquad w_i \in Dictionary\ ,\quad U \notin Dictionary$$

$$n = number\ of\ words\ in\ the\ sentence\ .$$

$$PN = set\ of\ proper\ name\ candidates\ .$$

$$\varepsilon = null\ string\ .$$

$$K = \text{constant value}$$

$$PN = \{\alpha U \beta \mid \alpha \in A, \beta \in B\ \} \qquad where\ A = \{w_{a-i,a}, i \in [0,K]\ \} \cup \{\varepsilon\}$$

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

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

Figure 1: Equation for generating proper name type I candidates

Proper name type II heuristic:

On the other hand, if all words are in the dictionary, the words that falls into one of the following two categories will be selected.

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 tag of the word w_i. The word that will be selected as a proper name candidate is:

- 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 a proper name. 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 proper names because the less-than-threshold probability of w_i may come from w_{i-1} or w_{i-2}. We can define the equation used to create proper name candidates as in Figure 2.

$$Sentence = w_1 w_2 ... w_a \ldots w_{n-1} w_n \qquad w_i \in Dictionary$$

$$wa\ is\ the\ word\ that\ has\ \text{probabiliy}\ less\ than\ threshold$$

$$n = number\ of\ words\ in\ the\ sentence\ .$$

$$PN = set\ of\ proper\ name\ candidates\ .$$

$$\varepsilon = null\ string\ . \quad K = \text{constant value}$$

$$PN = \{\alpha W \beta \mid \alpha \in A, \beta \in B\ \} \quad where\ A = \{w_{a-i,a-1}, i \in [0,K]\ \} \cup \{\varepsilon\}$$

$$B = \{w_{a+1,a+i}, i \in [0,K]\} \cup \{\varepsilon\} \quad ; w_{i,j} = w_i \ldots w_j \ : i < j$$

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

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

Figure 2: Equation for generating proper name type II candidates

3. Winnow Algorithm

Winnow algorithm used in our experiment bases on the algorithm described in [Blum, 1997]. Winnow is a neuron-like network (see Figure 3) 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 decides 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.

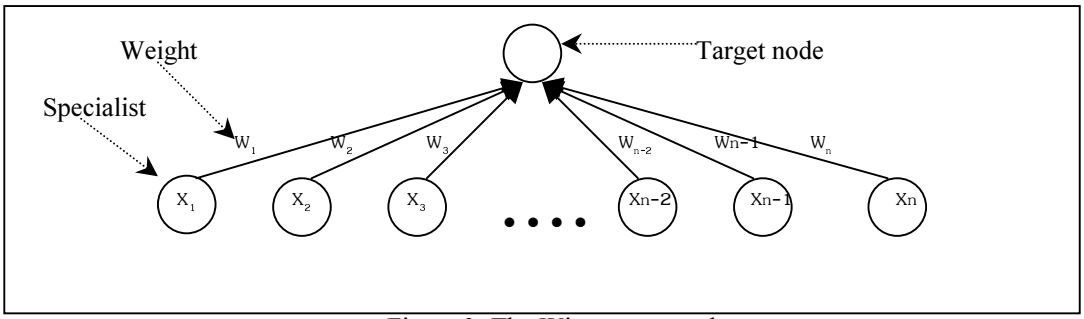

Figure 3: The Winnow network.

In our experiments, to train the proper names and their categories, we select all sentences containing proper names and assign their categories. The context around the proper name is used to form features in identifying the proper name. Similar process is done for any word, which are not proper names. The features used are the context words and collocations. Context words used in our experiment are 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 in our system. The best score candidate will be selected as the answer.

4. An overview of the system

Our algorithm for identifying proper names consists of four steps as follows: (see Figure 4)

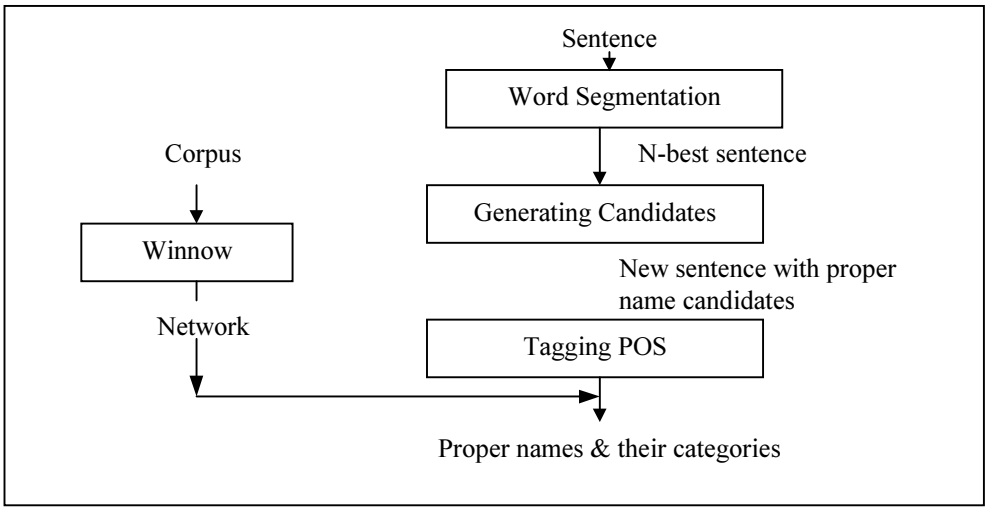

Figure 4: Proper Names Identification System

1. Word segmentation

For each input sentence, probabilistic trigram model [Kawtrakul, 1995] is applied to separate the sentence into words and to 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 follows:

Let sentence = $c_1c_2...c_m$ 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 tag. Find N-best W_i which have the highest probability

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

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

For example, C = นางเจนนีไปเดินตากลมเล่น. 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:

1. นาง/NTTL เจ/NCMN นนี/NPRP ไป/VACT เดิน/VACT ตาก/PPRS ลม/NCMN เล่น/ADVN
2. นาง/NTTL เจ/NCMN นนี/NPRP ไป/VACT เดิน/VACT ตา/CNIT กลม/VATT เล่น/ADVN

The part-of-speech tags in the example are described in [Sornlertlamvanich, 1997].

2. Generating candidates of proper name

From the result of step 1,we generate all candidates of proper names by the proper name type I and proper name type II heuristics described in Section 2.

For example, in the first sentence from step 1 the third word, นนี, is an unknown word. Therefore, we use the equation in Figure 1 to generate candidates. Then นนี will be U in that equation. If we use K=2, the A and B will be {ε, เจ, นางเจ } and {ε, ไป, ไปเดิน } respectively and all candidates are shown in Table 2.

.

No	α ($\alpha \in A$)	β ($\beta \in A$)	Candidates ($\alpha U \beta$)
1.	ε	ε	นนี
2.	ε	ไป	นนีไป
3.	ε	ไปเดิน	นนีไปเดิน
4.	เจ	ε	เจนนี
5.	เจ	ไป	เจนนีไป
6.	เจ	ไปเดิน	เจนนีไปเดิน
7.	นางเจ	ε	นางเจนนี
8.	นางเจ	ไป	นางเจนนีไป
9.	นางเจ	ไปเดิน	นางเจนนีไปเดิน

Table 2: Proper Name Candidates

3. Tagging part of speech

The new sentences will be formed by combining candidates that are obtained from step 2 with the remaining words in the original sentence. The part-of-speech tags of words in each sentence will be reassigned by the trigram tagger. The proper name tokens are assumed to be a 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 T_i that maximizes $P(T_i \mid W)$:

$$T = \arg\max_{T_i} P(Ti \mid W)$$

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

For example, the first sentence from step 1 after combining its candidates (see Table 2) and the remaining words, the results are shown in Table 3. Each word was assigned part-of-speech tag by using part-of-speech trigram model and the target word was proper noun tag (NPRP) because of our assumption. Let t_n be part-of-speech tag.

1. นาง/t_1 เจ/t_2 **นนี**/NPRP ไป/t_3 เดิน/t_4 ตาก/t_5 ลม/t_6 เล่น/t_7
2. นาง/t_1 เจ/t_2 **นนีไป**/NPRP เดิน/t_3 ตาก/t_4 ลม/t_5 เล่น/t_6
3. นาง/t_1 เจ/t_2 **นนีไปเดิน**/NPRP ตาก/t_3 ลม/t_4 เล่น/t_5
4. นาง/t_1 **เจนนี**/NPRP ไป/t_2 เดิน/t_3 ตาก/t_4 ลม/t_5 เล่น/t_6
5. นาง/t_1 **เจนนีไป**/NPRP เดิน/t_2 ตาก/t_3 ลม/t_4 เล่น/t_5

6. นาง/t_1 **เจนนีไปเดิน**/NPRP ตาก/t_2 ลม/t_3 เล่น/t_4
7. **นางเจนนี**/NPRP ไป/t_1 เดิน/t_2 ตาก/t_3 ลม/t_4 เล่น/t_5
8. **นางเจนนีไป**/NPRP เดิน/t_1 ตาก/t_2 ลม/t_3 เล่น/t_4
9. **นางเจนนีไปเดิน**/NPRP ตาก/t_1 ลม/t_2 เล่น/t_3

Table 3: The new sentences (the bold words are proper name candidates)

4. Predicting by Winnow

The sentences from step 3 will be sent to Winnow. The proper name candidates are the target words. The surrounding words and their part-of-speech tags are the context words and collocations. After that, every sentence will be scored by Winnow. Then, the sentence with the highest score will be selected as the answer.

For example,in the first sentence from Table 3, นนี is the target word. The context words are นาง ,เจ ,ไป ,เดิน ,ตาก ,ลม and เล่น. The words , นาง เจ ไป เดิน, and the part-of-speech tags ,t_1 t_2 t_3 t_4, are considered as collocations. Winnow will give the score to this sentence and predict the category of proper name candidate. Similarly, the remaining sentences will be processed. Then every sentence has a score and the highest-score sentence will be selected as the answer. In our example, the fourth sentence, นาง **เจนนี** ไป เดิน ตาก ลม เล่น, will be the answer and the category of เจนนี will be person name.

5. Preliminary results

A 5,000-sentence corpus was used in our experiment. Every sentence, which was manually separated into words and their parts of speech, was tagged by linguists. The resulting corpus is seperated into 2 parts; the first part about 80% of corpus is utilized for training and the rest is employed for testing. The accuracy on training set and test set are 96.12% and 92.17%, respectively.

6. Conclusion

Feature-based Proper Name Indentification in Thai was presented in this paper which can identify the proper names that combine various forms of known words and unknown strings. The experimental result shows that context words and collocations can be effectively used to find proper names and their categories, and Winnow is an efficient algorithm to apply in this task.

Acknowledgement

We would like to thank Software and Language Engineering Laboratory (SLL) for providing Orchid Corpus. Many thanks to Miss Virongrong Tesprasit for tagging categories of proper names.

References

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

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.

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

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

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.

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

Rarurom, S. 1991. Dictionary-based Thai Word Separation. Senior Project Report. (in Thai).

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