Automatic Sentence Break Disambiguation for Thai

Paisarn Charoenpornsawat and Virach Sornlertlamvanich

National Electronics and Computer Technology Center
22nd fl. Gypsum Metropolitan Tower, 539/2 Sriayudhaya Road,
Rajthevi, Bangkok, 10400 Thailand
Tel : (662) 642-5001 ext 303, 300 Fax: (662) 642-5015
E-mail: paisarn@nectec.or.th, virach@nectec.or.th

Abstract

Unlike English, there is no explicit sentence marker in Thai language. Conventionally, a space is placed at the end of the sentence when written in Thai. But it does not mean that a space always indicates the sentence boundary. In this paper, we propose the algorithm, which is a feature-based approach, to extract sentences from a paragraph by detecting the appropriate sentence breaking spaces. The algorithm considers the context around a space for determining the space as whether a sentence breaking space or not. The previous method, probabilistic POS trigram approach, considers only the coarse information of part-of-speech in a limited range of context whereas the feature-based approach considers as many features as possible. A feature can be anything that examines a specific information in the context around the target word sequence, such as context words and collocations. To automatically extract such features from a training corpus, we employ the learning algorithm, namely Winnow. The experimental results showed the effectiveness of Winnow comparing with POS trigram, and also demonstrated that Winnow is superior to POS trigram in our task.

Keywords: Thai sentence extraction; probabilistic part-of-speech trigram; Winnow; machine learning; sentence boundary.

1 Introduction

In Thai writing system, there is no explicit use of word and sentence boundaries. These cause difficulties in Thai language processing issues such as machine translation, information retrieval, etc. For the word boundaries, many algorithms [10, 13, 15] has been proposed and reported quite a high accuracy results. However, there are very few researches on Thai sentence boundary identification, and the previously proposed algorithms did not yield a satisfactory results.

The ambiguity level of end-sentence punctuation in English is less than the space in Thai. The period in English is used to denote a decimal point, an abbreviation and the end of sentence, but a space in Thai has more variety of usage such as determinating the end-of-sentence, the end-of-pharse/clause, or placing before/after numerals etc. Thus it makes this task attractive in using machine learning techniques.

The previous method, POS trigarm approach, consider only coarse information of parts of speech in a fix restricted range of context, some important information may be lost. In addition, another weakness of this POS trigram approach is that it does not take unordered long distance specific word collocations into account.

To overcome the disadvantages of the above mention method, we propose to use a feature-base approach to the problem of Thai sentence break disambiguation. Many feature base approaches have already been applied in several field of natural language processing and report quite a high accuracy results. A feature can be anything that tests for specific information in the context around the target space, such as context word, collocation. The idea is to learn several sources of features that characterize the contexts in which each space tends to occur. Then we can combine those features to disambiguate a space to a sentence-break space or non-sentence-break space for a given context. The new problem is then how to select and combine various kinds of features. Yarowsky [18] proposed decision lists as a way to select and combine various kinds of features, and to solve a target problem by applying the single strongest features, whatever type is. Golding [5] proposed a Bayesian hybrid method to take into account all available evidence, instead of only the strongest one. The method was applied to the task of context-sensitive spelling correction and was reported to be superior to decision lists. Golding and Roth [6] applied Winnow algorithm in the same task and found that the algorithm performs comparable to the Bayesian hybird method when using pruned feature set and is better when using unpruned sets or unfamiliar test sets. Mekanavin and etc. [10] applied Winnow and RIPPER [3] in Thai word segmentation task. From the experiment results showed

that both algorithms appeared to outperform the existing Thai word segmentation methods and Winnow is superior to RIPPER. Charoenpornsawat and etc [2] applied Winnow to the problem of identification unknown word boundary for Thai. Later, Kijsirikul and etc [7] applied Winow and RIPPER in the task of Thai name identification.

In this paper, we investigate the machine learning algorithm, Winnow, to automatic identifying Thai sentence boundaries. The advantages of our algorithm are; 1) no syntactic knowledge is required, 2) the sentence-boundary-identification information is extracted automatically from the training corpus, 3) our algorithm gives higher accuracy than those reporting in the previous works.

2 The Thai Language in Brief

This section briefly describes some Thai language features by comparing with the English language.

2.1 The use of punctuation marks

Except for the traditional punctuation marks, a space is only the marker that is only the marker that is generally used in Thai texts.

In the English language, a space is explicitly used as the word boundary marker. In Thai, there is no word-boundary marker; words are written consecutively.

Many kinds of punctuation marks are efficiently used in writing English texts. On the contrary, punctuation marks are not usually used in writing Thai. A space is only a character that is unnoticeably used to break between words, phrases, and sentences. As a result the use of a space is ambiguous whether to be a break between words, phrases or sentences.

2.2 Comparison of Thai with English in sentence boundary identification

In English, a period is used as sentence boundary marker. The task of sentence boundary identification in English is then the task of disambiguating the period whether it functions as a sentence break or not. In the same way, a space is used as a sentence boundary marker in some cases. Consequently, to identify sentence boundary in Thai is the task of disambiguating the space whether it functions as a sentence break or not. Superficially, Thai and English sentence boundary identification is quite similar. However, the degree of ambiguity of Thai sentence boundary identification is much higher. A space in Thai can be used in the following senses. [4]

1) A space is used to break between sentences.
2) A space is used to break between phrases or clauses within a sentence.
3) A space is used to break between sentences in a cohesive group of sen-tences.
4) A space is used to break before and after numerals.
5) A space is used to break between coordinate words in lists.
6) A space is used to break between the first and the second names of people.
7) A space is used to break before and after some special orthographic symbols and punctuation marks.

On the contrary, a period in English is commonly used in 3 senses, in the contexts of sentence boundary marker, abbreviation, and number. More than 85 percent of the usage the use for sentence breaking. Whereas only 30 percent of the usage of the spaces in Thai is for sentence breaking.

3 Previous Works

In English, an end-sentence marker (a period, a exclamation point and a question mark) may occur both within a sentence and at the end of sentence, therefore, there were also some works that attempt to disambiguate these markers. [14] used the CART (Classification and Regression Tree) to classify periods by using the information about one word context on either side of the punctuation mark. This approach was trained on the 25 million words of relabeled training data from a corpus of AP newswire. The result of training was the classification tree used to identify whether a word ending in a period was at the end of sentence. The error rate when testing on the Brown corpus was 0.2%. [12] applied 2 machine learning techniques: Neural Network and Decision Tree, to the sentence boundary disambiguation task. This approach estimated the parts-of-speech distribution of the tokens preceding and following each punctuation mark as the input feature to a machine learning algorithm that classified the punctuation mark. The error rate of using the Neural network which trained by the data 3,179 items was 1.3% and the decision tree method on trained data 6,373 items is 1.0%.

In the previous works about Thai sentence extraction, we found only two publications, [9] and [11]. [9] presented a method in separating Thai sentences in a paragraph. This method segmented a paragraph to morphemes and used the main verbs to estimate the number of sentences. The conjunction identified in the syntactic analysis was marked to be a sentence boundary. Basically it was a grammatical rule based approach. The accuracy of this approach was 81.18%. [11] recently proposed to use the part of speech trigram model to identify sentence the boundary. The accuracy of this approach was about 85%.

4 Winnow Algorithm

This section briefly describes learning algorithm Winnow used in our tasks, and defines the features for training the algorithm.

The advantages of Winnow, which made us decide to use for our task, are that (1) it runs fast because of its Sincremental algorithm (2) it is not sensitive to extra irrelevant features [8]

4.1 Winnow

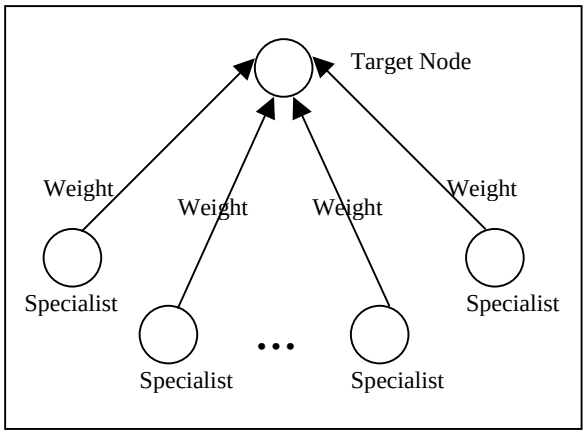


Figure 1: Winnow Network

The Winnow algorithm used in our experiment is the algorithm described in [1, 6, 8]. Winnow is shown in Figure 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 each 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 experiment, we have each specialist examine one or two attributes. For example, a specialist may predict the value of the target concept by checking for (attrib1=value1) and (attribute2=value2). A specialist only makes a prediction if its condition is true (in case of two attributes), and in that case it predicts the most popular outcome out of the last *k* times it had the chance to predict. In fact, we may have each specialist examine more than two attributes, but for the sake of simplification of preliminary experiment, let us assume that two attributes for each specialist are enough to learn the target concept.

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 in correctly 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. This weight updating method is the same as the one used in [1]. A specialist may choose to abstain instead of giving a prediction on any given example in case that it did not see the same values of an attribute in the example.

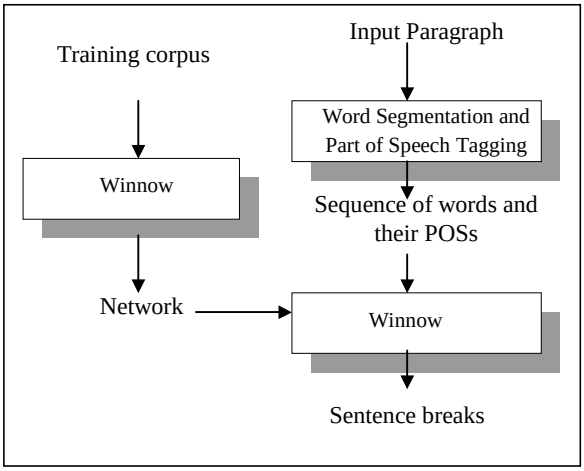
4.2 Features

To train the algorithm for a sentence break space, the context around a space which is a sentence break space or non-sentence-break space is used to form the features. The features used in the approach are the collocations and the number of words in the left and right of the target space. Collocations are patterns of up to 2 contiguous words and part-of-speech tags around the target space. Therefore the total number of features is 10; two features for number of words, and eight features for collocations.

5 An Overview of the System

We construct our sentence break disambiguation system based on Winnow as shown in Figure 2. The training set consists of paragraphs which markup sentence boundaries. Each sentence is segmented into words with appropriate part-of-speech tags. In training mode, a training set composed of segmented paragraph and segmented and tagged sentence is passed to Winnow algorithm to learn a network by using the algorithm described above. After a network is learned, a test set is fed to the system for evaluation the performance.

Figure 2: The sentence break disambiguation system



Identifying a sentence-break space in our approach consists of two steps as follows:

5.1 Word Segmentation and Part-Of-Speech Tagging

For each input paragraph, probabilistic trigram model [10] is applied to separate the paragraph into a sequence of words including spaces and assign their part of speech.

Let $C = c_1c_2\ldots c_m$ *be an input character string,* $W = w_1w_2..w_n$ *be a possible word segmentation, and* $T = t_1t_2..t_n$ *be a sequence of parts of speech. Find* $w_1,w_2...w_N$ *and* $t_1,t_2...t_N$ *which are the highest probability of sequence of*

words and sequence of part-of-speech tag. We compute the highest probability of $P(W_j, T_i)$ in the following fashion:

$$\tau = \arg\max_{W,T} P(W,T \mid C)$$
$$= \arg\max_{W,T} P(W,T)P(C \mid W,T) / P(C) \text{.....(1)}$$
$$= \arg\max_{W,T} \prod_k P(t_k \mid t_{k-1}, t_{k-2}) * P(w_k \mid t_k)$$

where $P(t_k|t_{k-1},t_{k-2})$ and $P(w_k|t_k)$ are computed from the corpus. $P(C)$ is a fixed constant for every candidate so it can be eliminated.

5.2 Space determination by Winnow

From the result of word segmentation and part-of-speech tagger. The collocation, number of words before and after the spaces are extracted and sent to Winnow. The Winnow algorithm determines each space as either a sentence-break space based on its trained features.

An example of space determination by Winnow is shown as follows:

ลำดับที่ <SPACE>1

(Number)<SPACE>1

The features of this example are "ลำดับที่" (Number), "1" and their part of speech. Winnow determines the space as either sentence-break space or non-sentence-break space by the surrounding context. In this case, the space is correctly determined as a non sentence-break space.

6 Preliminary Experiment

6.1 Corpus

We used the ORCHID corpus [16] to evaluate our approach. The ORCHID corpus is a part-of-speech tagged corpus containing 10,864 sentences. In the corpus, every paragraph is manually tagged in both sentence and words levels. Each word is assigned an appropriate part of speech manually by linguists. The corpus is divided into two parts. The first part, 90 percent of the corpus, is used for training and the rest is used for testing.

For Thai, a space can be a separator for sentences, pharases, clauses and the like. To discriminate the types of spaces, we defined a non-sentence-break space (NSBR) tag for the spaces occurred in the sentence, and a sentence-break space (SBR) tag for the space between sentence. There two new part of speech tags for the space are applied in the corpus.

6.2 Training Methodology

We train Winnow by the following processes.

Construct positive examples. The sentence break spaces are considered as target spaces. The collocations and number of words before and after the spaces are sent to Winnow.

Construct negative example. The non-sentence-break spaces are considered as target spaces. The collocations and number of words before and after the spaces are sent to Winnow.

6.3 Evaluation and Results

To evaluate our approach, we use the same data in [11] and compare it with the result from the part-of-speech trigram approach, reported in [11].

The performance criteria that used in this work are derived from [17]. We measure the performance of our approach in terms of the percentage of break-correct, space-correct and false-break. These measures are explained as follows:

Break-correct = (CB / RB) x 100%

Space-correct = (CS / RS) x 100%

False-break = (FB / RS) x 100%

where

CB is the number of correct classified sentence-break from the test set.

FB is the number of false classified sentence-break from the test set.

CS is the number of correct classified sentence-break and non-sentence-break from test set.

RB is the number of sentence-break spaces from the reference.

RS is the number of sentence-break and non-sentence-break spaces from the reference.

The difference between the break-correct and space-correct is whether the non-sentence-break spaces are included in calculation. The space-correct score gives credit when both the test and reference sentence are a non-sentence-break at the same space, while the break-correct score only considers the break space. Both break-correct and spaces-correct score are essential in indicating the accuracy of extraction. In our test set, the ratio of the number of non-sentence-break space by the number of sentence-break space is about 5:2. The measures are used to evaluate on the sentence-break accuracy only. Therefore, if an algorithm classifies all spaces as sentence-space then the break-correct is 100%, and the space-correct is about 30%. false-break is the assessment of the insertions, this score indicate how often an algorithm returns the unreliable break space. In conclusion, the good algorithm must yield high space-correct and break-correct scores, and low false-break scores. Table 1 shows the comparison results between the part-of-speech trigram approach and our approach.

	Our algorithm(%)	Trigram(%)
Space-correct	89.13	84.57
Break-correct	77.27	75.97
False-break	1.74	8.13

Table 1: The results of classifying spaces.

Our approach shows a significant improvement to the part-of-speech trigram approach. In case of our approach, the scores of space-correct, break-correct are as high as 89.13%, 77.27% and the score of false-break is as low as 1.74%.

7 Conclusion

This paper presents the feature-based approach for identifying sentence break. Our approach is to classify any spaces in the paragraph to be a sentence-break or non-sentence-break space. The experiment result shows the significant improvement of our approach to the previous reported trigram approach. However, the context that we used as the features for the Winnow is very limited because of the simplicity of using punctuation in Thai. Though, there is no obvious concept for writing Thai in sentences, the experiment result, however, that people write a text with a hidden information of the structure of sentence. The result also confirms the possibility in breaking Thai text into sentences.

Our future works include in-depth investigation various kinds of features and another machine learning techniques such as ID3, C4.5, RIPPER etc.

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] Charoenpornsawat, P., Kijsirikul, B. and Meknavin, S. 1998. Feature-based Thai Unknown Word Boundary Identification Using Winnow. *In Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS'98).*

[3] Cohen W. 1995 Fast effective rule induction. *In Proceedings of the Twelfth International Conference on Machine Learning.*

[4] Danvivathana, N. 1987, The Thai Writing System, *Forum Phoneticum 39*, Helmut Buske Verlag Hamburg.

[5] Golding, A. R. 1995. A Bayesian Hybrid Mehod for Context-sentitive Spelling Correction. *In Proceedings of the Third Workshop on Very Large Corpora.*

[6] Golding, A. R. & Roth, D. 1996. Applying Winnow to Context-Sensitive Spelling Correction. In Lorenza Saitta, editor, Machine Learning*: Proceedings of the 13th International Conference on Machine Learning*.

[7] Kijsirikul, B., Charoenpornsawat, P. and Meknavin, S. 1999. *Comparing Winnow and RIPPER in Thai Named-Entity Identification. In Proceedings of the Natural Language Processing Pacific Rim Symposium 1999(NLPRS'99).*

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

[9] Longchupole, S. 1995. Thai Syntactical Analysis system by method of splitting sentences from paragraph for machine translation. *Master Thesis.* King Mongkut's Institute of technology Ladkrabang (in Thai).

[10] Meknavin, S., Charoenpornsawat P. and Kijsirikul, B. 1997. Feature-based Thai Word Segmentation. *Proceeding of NLPRS'97*.

[11] Mitrapiyanurak, P. and Sornlertlamvanich, V. 2000. The Automatic Thai Sentence Extraction. *Proceedings of the Simposium on Natural Language Processing in Thailand*.

[12] Palmer, David D., Hearst, Mati A. 1997. Adaptive Multilingual Sentence Boundary Disambiguation, *Computational Linguistics Volumn 23 No. 2.*

[13] Rarunrom, S. 1991. Dictionary-based Thai Word Separation. *Senior Project Report.* Chulalongkorn University (in Thai).

[14] Riley, Michale D. 1989. Some applications of Tree-based modeling to speech and language indexing. *Proceeding of the DARPA Speech and Natural Language Workshop*.

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

[16] Sornlertlamvanich, V., Charoenporn, T. and Isahara, H. 1997. ORCHID: Thai Part-Of-Speech Tagged Corpus. *Technical Report Orchid Corpus.* National Electronics and Computer Technology Center.

[17] Taylor, P. and Black, A. 1998. Assigning Phrase Breaks from part-of-speech Sequences. *Computer Speech and Language 12*

[18] Yarowsky, D. 1994. Decision Lists for Lexical Ambiguity Resolution. *In Proceedings of 32nd Annual Meeting of the Association for Computational Linguistics.*