THE AUTOMATIC THAI SENTENCE EXTRACTION

PRADIT MITTRAPIYANURUK and VIRACH SORNLERTLAMVANICH
National Electronics and Computer Technology Center (NECTEC), Thailand

Unlike English, there is no explicit sentence marker in the Thai language. Conventionally, space is placed at the end of sentence in Thai writing. But it does not mean that space always indicates the sentence boundary. It is also used as other purposes [Danvivathana 1987]. This paper presents an algorithm to extract sentences from paragraph by detecting the true sentence breaking spaces, by applying the statistical part-of-speech (POS) tagging technique to the space classification problem. The algorithm considers 2 consequent strings with a space in between each time for determining the space as whether a true sentence breaking space or not. We divided the ORCHID Thai POS tagged corpus into 10 portions for cross-validation test. The evaluation result shows that the average accuracy of space classification and break-space detection are 85.26% and 79.82% respectively and the average of false-break rate is 8.75%. Our approach also shows a significant improvement to the traditional statistical POS tagging technique. The average of POS tagging error rate reduction is as high as 11.3%.

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

1. INTRODUCTION

One of the crucial problems in a Thai text analysis algorithm is how to tokenize a paragraph into sentences. Thai is an agglutinative language without an explicit punctuation mark to determine the end of sentences in a paragraph. A Thai text processing handles whole paragraph as the input text. Therefore the paragraph size is limited due to larger memory space and longer processing time are required when the paragraph size is large. Fortunately, in Thai writing [Danvivathana 1987], spaces are normally used at the end of sentences. But not all spaces are the end of sentence marker, they also have been used as other purposes such as phrases/clause break in a sentence, place before and after numerals etc. This point is where Thai sentence extraction comes into play by detecting the true sentence breaking spaces in the paragraph.

This paper presents the algorithm of sentence extraction from Thai text paragraph by applying the statistical part-of-speech (POS) tagging technique to classify spaces into 2 types: non-sentence-break space (NSBS) or sentence-break space (SBS). Paragraph can then be separated into sentences by sentence-break space in between. The following sections discuss the previous work-related to the sentence segmentation problem, describe our algorithm in details and present the experimental results of training/testing on the prelabel "ORCHID" Thai text corpus, respectively.

2. PREVIOUS WORK

For the previous work about Thai sentence extraction, we found only one publication, [Longchupole 1995] presents the method of splitting Thai sentences from paragraph. This method segments a paragraph to morphemes and uses the main verbs to estimate the number of sentences. The conjunctions of sentences are marked to be the sentence boundary and identified by the syntactic analysis of Thai sentence. The accuracy of this apporach is 81.18%. However, the disadvantage is that it requires analyzing an entired paragraph thus limiting the paragraph size.

In English, the end-sentence markers (the period, the exclamation point and the question mark) may occur both within sentences and at the end of sentence then there were also some works that attemp to disambiguate these markers. [Riley 1989] uses 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 is trained on the 25 million words of prelabeled training data from a corpus of AP newswire. The result of training is the classification tree used to identify whether a

word ending in a period is at the end of sentence. The error rate when testing on the Brown corpus is 0.2%. [Palmer 1997] applies 2 machine learning techniques: Neural Network and Decision Tree, to the sentence boundary disambiguation task. This approach estimates the parts-of-speech distribution of the tokens preceding and following each punctuation mark as the input feature to a machine learning algorithm that classifies the punctuation mark. The error rate of using the Neural network which trained by the data 3,179 items is 1.3% and the decision tree method on trained data 6,373 items is 1.0%. 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 space in Thai has more variety of usages such as the end-of-sentence, the end-of-pharse/clause and place before/after numerals etc. Then it requires further study in details before being able to apply these approaches to Thai.

3. OUR ALGORITHM

This paper assumed that the input paragraph consists of many sentences separated by spaces in the right manner without considering the effect of mis-editing from human error. It is conventional to use space in Thai writing in the following purposes: [Danvivathana 1987], [Thavaranon 1978]

- used between sentences
- used between phrases or cluase within a sentence
- used between sentences in a cohesive group of sentences
- used before and after numerals
- used between coordinate words in lists
- used between the first and the second names of people
- used before and after some special orthographic symbols and punctuation marks

For more details and examples of the general characteristics of space in Thai writing can see in the above references. We divide the space by its function into 2 different types: sentence breaking and non-sentence breaking space. The first list of above space's function that used between sentences is considered to be the sentence breaking type. The rest of them are the non-sentence breaking one. After processing in our system, by classifying all spaces in a paragraph into break or non-break space, the sequence of expected sentences which are the text that reside between any detected break space were returned.

The block diagram of our system is shown in Figure 1. The tokenization/word segmentation stage breaks the paragraph into tokens. The token means the group of connected characters enclosing by space. Because the Thai language has no explicit word delimiter, the token that consists of Thai character stream is splitted into sequence of words by the word segmentation function [Sornletlamvanich 1993].

The two adjacent tokens, one from previous token and one from the current token, are reconstructed to the word sequence with a space in between. Any spaces in this word sequence are classified to be one of two possible classes, break or non-break space. We define this classification problem in terms of statistical POS tagging. The most probable sequence of POSs and individual word-level POS assignments determines the most probable POS assignment of any word sequence.

Therefore the classification task is only to determine whether the POS of any spaces in the most probable sequence of POS is break or not. We use the part-of-speech trigram model [Sornlertlamvanich 1999] as shown in Equation (1) to compute the POS sequence probabilities and introduce the viterbi algorithm for computing the most probable sequence of POSs.

$$\tau = \max_{t_1, t_2 .., t_n} \arg \prod_{i=1}^{n} P(t_i \mid t_{i-1}, t_{i-2}) \bullet P(w_i \mid t_i) \qquad \cdots (1)$$

where τ is the sequence of POS $\{t_1,t_2,\ldots,t_n\}$ that maximize the POS sequence probabilities of the associated sequence of word $\{w_1,w_2,\ldots,w_n\}$ to be tagged.

For clarity in this work, we define 2 possible POS tags of the space: SBS (sentence-break-space) and NSBS (non-sentence-break space) for using in our system. Noted that it is possible that the space in the word sequence that used to be the non-breaking space in the previous token can be changed to the breaking space when concatenate with the current token. Therefore we must scan the space between the current and previous token as well as the spaces within previous token. If there is no space that is POS of SBS then all of this word sequence will be used as the previous token in the next iteration. But if SBS space is found then the output sentence is the first word until the word before the SBS space of word sequence. The rest words after this space are used as the previous token in the next iteration. It is obvious that this algorithm can solve the limitation on memory and processing time because it scans tokens instead of whole paragraph.

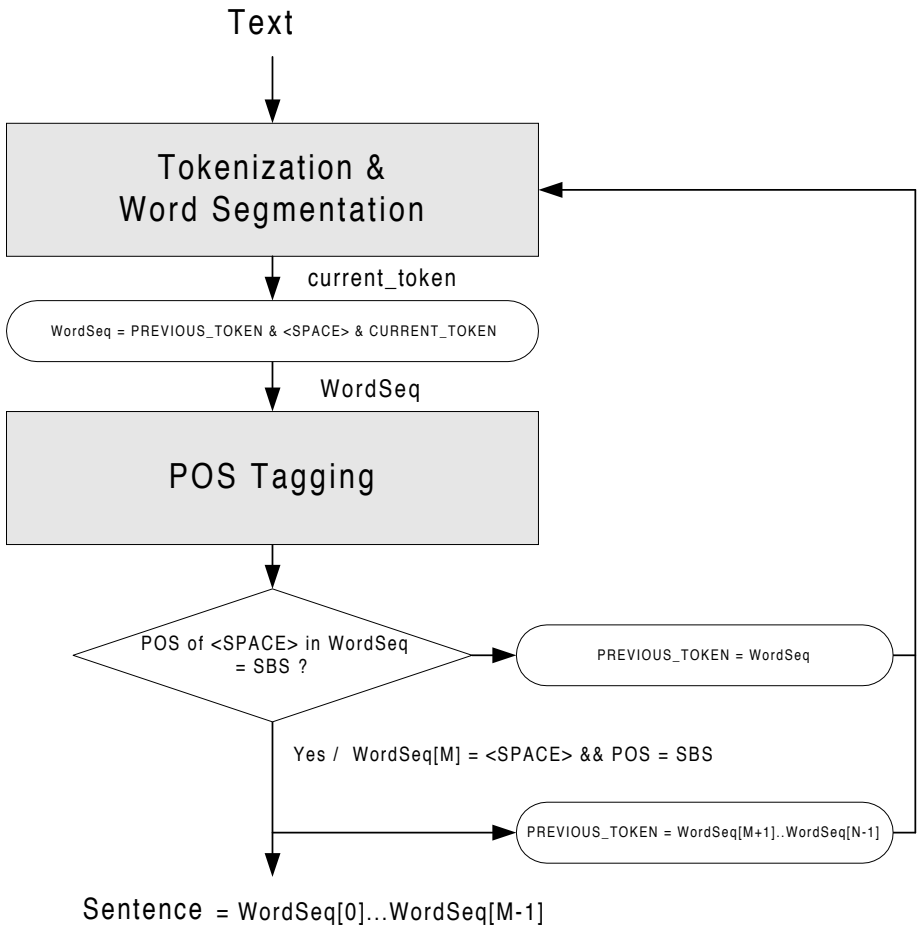

Figure 1. The block diagram of our system

4. EVALUATION

4.1 Data preparation

For testing our algorithm, we use ORCHID Thai part-of-speech tagged corpus [Sornlertlamvanich 1999] which is annotated into three levels: paragraph, sentence, and word level. Each paragraph is manually separated into sentences, then into words and each word is assigned an appropriate part-of-speech tag from 47 different tags. For simplicity, we select the paragraph that has more than 3 sentences in the training and testing. There are totally 1,330 paragraphs with 9,528 sentences after filtered. From the ORCHID corpus, the POS of space is originally punctuation (PUNC). We convert the POS of all spaces in any sentences into NSBS (non-sentence-break space) and insert the virtual space that has POS of SBS (sentence-break space) between sentence and in the first word of each paragraph. For cross-validation test, we divide the data into 10 sets, each set is equal to the number of paragraph. We evaluate the accuracy of sentence extraction in each set and the rest 9 sets that have not been evaluated are used to train the POS trigram probabilities by using the CMU-Cambridge statistical language modeling toolkit (see more details at http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html).

4.2 Experimental Results

The performance criteria that used in this work are derived from [Taylor 1998]. We measure the performance of our algorithm in term of the percentage of break-correct, spaces-correct and false-break. These measures are explained below

$$\text{Break-correct} = (CB / RB) \times 100\%$$
$$\text{Space-correct} = (CS / RS) \times 100\%$$
$$\text{False-break} = (FB / RS) \times 100\%$$

where

CB is the number of correct classified break spaces from test set.
FB is the number of false classified break spaces from test set.
CS is the number of correct classified spaces (break and non-break space) from test set.
RB is the number of break spaces from reference.
RS is the number of spaces (break and non-break space) from reference.

The difference between the break-correct and space-correct is whether the non-break spaces are included in calculation. The spaces-correct score give credit when both the test and reference sentence is a non-break at the same space, while the break-correct score only looks at the break space. Both break-correct and spaces-correct score are essential to indicate the accuracy of extraction. In our test data, each set has the ratio of the number of non-break space by the number of break spaces about 2.5:1, if an algorithm which classify all spaces as break one will has 100% break-correct score but has space-correct score about 30%. The false-break score is the assessment of the insertions, this score indicate how often an algorithm return the unreliable break space. Then the good algorithm must has the space-correct and break correct score high but the false-break score is low. By using these measures in evaluation, we get the results of each test set shown in the Table 1. We found that the average percentage of space-correct, break-correct and false-break is 85.26%, 79.82% and 8.75% respectively.

Because of the limitation on the memory resource and processing time, common POS tagging approach usually processes each token by token that separated by space and then tags the space as the punctuation. Trade-off of this scheme may worsen the accuracy of tagging because the relationship between words across the tokens is not taking into account in determining the probable POS sequence. Whereas our algorithm use the POS tagging to classify the type of space then we also get the POS value of any words by product. Therefore, in this paper we also evaluate the

accuracy of POS tagging of our algorithm comparing with the algorithm that works on token-by-token. The result is shown in Table 2. The average of error rate is reduced by 11.3%

Table 1. Experimental Results for each test set.

Test set	Number of spaces		%Space-correct	%Break-correct	%False-break
	SBS	NSBS			
1	920	2115	84.57	75.97	8.13
2	1004	3003	86.62	80.27	8.43
3	853	2364	84.39	82.88	11.06
4	911	2233	84.54	80.68	9.86
5	871	2222	82.76	76.00	10.47
6	915	2698	85.16	76.17	8.80
7	1001	2368	85.54	81.71	9.20
8	1045	1888	86.29	77.99	5.86
9	1044	1700	86.18	84.00	7.70
10	1964	2022	86.36	81.95	7.80
Average			85.26	79.82	8.75

Table 2. POS tagging accuracy of our algorithm VS the token-by-token tagging scheme.

Test set	Number of words	POS tagging accuracy		%Error rate reduction
		Token-by-token tagging	Our algorithm	
1	17530	94.1%	94.8%	11.8%
2	19959	94.19%	95.0%	13.9%
3	17347	93.73%	94.79%	16.9%
4	18174	93.56%	94.0%	6.83%
5	17068	93.76%	94.12%	5.77%
6	17288	94.02%	94.5%	8.02%
7	18824	93.58%	94.39%	12.62%
8	17656	93.49%	94.09%	9.21%
9	19371	94.42%	95.15%	13.08%
10	20039	94.75%	95.54%	15.04%
Average		93.96%	94.64%	11.3%

4.3 Discussion

Table 2 shows that the POS tagging accuracy is high average on 94.64% but why the accuracy of space detection in Table 1 is not as high as we expected. By human inspection, we found that many of these false break positions can be accepted to a correct break. It is the fact that most Thai people do not have the sense of sentence breaking in writing. This causes the writer to ignore the subject of sentence and use many conjunction words to link between the phrases and sentences. Then the sentence break positions in a paragraph are so ambiguous that Thai people feel complicated to classify them. In ORCHID corpus construction, the sentence segmentation step that done manually has the criteria of “short and acceptable” in the ambiguous case [Sornlertlamvanich 1999]. This scheme may cause variations in the sentence segmentation caused by corpus developers. They do not always place breaks in the same place: some positions can be either breaks or non-break without meaning lost, while other positions must be only one type. However if the corpus allows both sentence break and non-break at the ambiguous place, we can estimate the probability and evaluate the result more accurately.

5. CONCLUSION

This paper presents the algorithm for extracting the sentence from the Thai text paragraph. Our approach is to classify any spaces in the paragraph to be break or non-break space. The output sentence is the text between two break sentence spaces. We define this classification problem in term of POS tagging. By using the part-of-speech trigram model to determine the most probable POS sequences of each sequence of words that have one or more spaces. The output POS of the space that may be sentence-break-space (SBS) or non-sentence-break-space (NSBS) is the classified result. The average accuracy of space classification, break-space detection and false-break rate tested on the ORCHID corpus are 85.26%, 79.82% and 8.75% respectively. Furthermore, we found that the error rate of POS tagging in the sentence that achieved by the product in our algorithm comparing with the tagger that works on token-by-token is reduced by average 11.3%

REFERENCES

Danvivathana, Nantana, 1987, The Thai Writing System, Forum Phoneticum 39, Helmut Buske Verlag Hamburg.

Thavaranon, Kobkul , 1978 , Spacing in Thai Writing, M.A.Thesis Department of Thai Chulalongkorn University (in Thai)

Longchupole, Sungkornsarun, 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)

Riley,Michale D. 1989. Some applications of Tree-based modeling to speech and language indexing. Proceeding of the DARPA Speech and Natural Language Workshop, pages 339 – 352. Morgan Kaufmann

David D.Palmer, Matri A.Hearst, Adaptive Multilingual Sentence Boundary Disambiguation , Computational Linguistics Volum 23 No.2 June 1997 pp.241-267

Virach Sornlertlamvanich , Naoto Takahashi and Hitoshi Isahara , Building a Thai paprt-of-speech tagged corpus (ORCHID) , Journal of the Acoustical of Society of Japan , Volume 20 No.3 , pp 189-198, May 1999

Sornlertlamvanich,V. 1993, Word segmentation for Thai in machine translation system, Machine Translation, NECTEC pp 556-561 (in Thai)

Taylor,P. and Black, A. (1998). Assigning Phrase Breaks from part-of-speech Sequences, Computer Speech and Language 12, 99-117.