A Comparative Study on Thai Word Segmentation Approaches

Choochart Haruechaiyasak*, Sarawoot Kongyoung*† and Matthew N. Dailey†

*Human Language Technology Laboratory (HLT)
National Electronics and Computer Technology Center (NECTEC)
Thailand Science Park, Klong Luang, Pathumthani 12120, Thailand
Email: [choochart.haruechaiyasak, sarawoot.kongyoung]@nectec.or.th

†Computer Science and Information Management
Asian Institute of Technology
P.O. Box 4, Klong Luang, Pathumthani 12120, Thailand
Email: mdailey@ait.ac.th

***Abstract*—In this paper, we analyze and compare various approaches to Thai word segmentation. Word segmentation approaches can be classified into two distinct types: dictionary-based (DCB) and machine learning-based (MLB). DCB approaches rely on sets of stored terms to parse and segment input text. MLB approaches, on the other hand, rely on statistical models estimated from training corpora using machine learning techniques. We compare two algorithms based on the DCB approach: longest matching and maximal matching. We compare four algorithms based on the MLB approach: Naive Bayes (NB), decision tree, Support Vector Machine (SVM), and Conditional Random Field (CRF). In a series of experiments, the DCB approach performed better than the NB, decision tree and SVM algorithms from the MLB approach. However, the best performing algorithm was the CRF algorithm, with precision and recall of *95.79%* and *94.98%*, respectively. We believe that the CRF is the best existing statistical model for problems like Thai word segmentation.**

Keywords: Word segmentation, tokenization, morphological analysis, dictionary-based algorithms, machine learning-based algorithms.

I. Introduction

Text segmentation or term tokenization is one of the fundamental tasks in natural language processing (NLP). Most NLP applications require input text to be tokenized into individual terms or words before being processed further. For example, in machine translation, text must first be tokenized into a series of terms before it can be further analyzed grammatically and translated into another language. For information retrieval systems, in which the inputs are text documents and text queries, text is first tokenized into individual terms. The processed terms are then organized into an inverted file index data structure for fast retrieval. In speech synthesis applications, the tokenized terms are segmented further into syllables, which are then mapped into phoneme units.

Like Chinese, Japanese, and Korean, the Thai written language is unsegmented, i.e., it is written continuously without the use of word delimiters. This means Thai text tokenization is not as simple as it is for Latin-based languages such as English, French, and Spanish. With Latin-based languages, text is easily tokenized into terms by observing the word delimiting characters such as spaces, semicolons, commas, quotes, and periods. Unsegmented languages like Thai, on the other hand, require specialized algorithms to find word boundaries prior to tokenization.

Virtually all previously proposed techniques for word segmentation in unsegmented languages can be classified into two distinct categories: dictionary-based (DCB) approaches and machine learning-based (MLB) approaches.

DCB approaches use a set of terms from a dictionary for parsing and segmenting input text into word tokens. During the parsing process, we look up series of characters in the dictionary to find matches. The performance of DCB approaches depends critically on the quality and size of the the dictionary. DCB approaches are relatively simple and straightforward. However, there are two problems with the approach. The first is the unknown word problem. Unknown words are words in the input text that are not in the dictionary [**?**]. The second problem is parsing ambiguity. Ambiguity occurs when there is more than one way to segment a given character sequence. Ambiguity can be addressed via selection heuristics such as selecting the longest possible term (*longest matching* [**?**]) or selecting the segmentation yielding the minimum number of word tokens (*maximal matching* [**?**]).

MLB techniques aim to address the drawbacks of DCB approaches. Using a tagged corpus in which word boundaries are explicitly marked with special annotations, machine learning algorithms build statistical models based on the *features* of the characters surrounding the boundaries. The most common features for Thai word segmentation models are the identities and categories of characters within an n-gram of characters surrounding a candidate word boundary. Character types are quite diagnostic for word segmentation; for example, certain leading vowels often appear at the beginning of a word, whereas tone marking characters can never begin

a word. In MLB approaches, the word segmentation problem is formulated as a binary classification task in which each character in the text string is predicted to be a member of one of two classes: the beginning of a word (labeled as class "B" in our corpus) and intra-word characters (labeled as class "I" in our corpus). The main advantage of MLB approaches is that they do not require dictionaries. The unknown word and ambiguity problems are handled in principle by extracting sufficiently rich contextual information from the n-gram and by providing a sufficiently large set of training examples to enable accurate classification. The main disadvantage of MLB approaches is that their performance depends critically on the characteristics of the document domain and the size of the training corpus. For example, if a model is constructed based on a corpus from one specific domain, it might not perform well on documents from other domains.

In this paper, we compare several DCB and MLB algorithms. We first evaluate two algorithms, longest matching (LM) and maximal matching (MM), that use the DCB approach. As a dictionary, we use either the domain-specific words occurring in the training corpus, or the more general LEXiTRON dictionary, which contains approximately 30,000 words [**?**]. We then evaluate four MLB algorithms: Naive Bayes (NB), decision tree (DT), Support Vector Machine (SVM), and Conditional Random Field (CRF). For these algorithms, we transform input text into feature vectors based on the character types occurring within the n-gram of characters surrounding candidate word boundaries.

The remainder of this paper is organized as follows. In the next section, we review previous work on Thai word segmentation. In Section III, we give the details of the DCB and MLB approaches used for the performance evaluation. Section IV presents the experimental results and discussion. Section 5 gives the conclusion.

II. Related Work

There are numerous works related to word segmentation tasks. Many techniques for word segmentation and morphological analysis have been reported for languages such as Chinese and Japanese [**?**], [**?**], [**?**].

For Thai word segmentation, Charoenpornsawat (1999) presented some good reviews of previous works in his master thesis [**?**]. The early works for Thai word segmentation started in 1980's. For example, Poowarawan (1986) introduced the longest matching algorithm for dictionary based approach [**?**] . Sornlertlamvanich (1993) introduced the maximum matching algorithm that splits a sequence of characters into all possibilities of segmentation based on a word set. The algorithm selects the segmentation path with the lowest number of segmented tokens [**?**]. Kawtrakul et al. (1997) proposed a language modeling technique based on a tri-gram markov model to select the optimal segmentation path [**?**]. Meknavin et al. (1997) constructed a machine learning model by using the Part-Of-Speech (POS) features.

Recent work by Kruengkrai et al. (2006) used the Conditional Random Field (CRF) algorithm for training a word segmentation model. The CRF is a recent novel approach which has been shown to perform better than other machine learning algorithms for the task of labeling and segmenting sequence data [**?**], [**?**]. This work focused mainly on solving the ambiguity problem in word segmentation. Two path selection schemes based on confidence estimation and Viterbi were proposed. The feature set used in their model required the POS tagging information. Therefore, if the POS tagging is inaccurate, the performance of the word segmentation could be effected. In this paper, we construct the feature set based on the character types of the n-gram characters surrounding the word boundary. As shown from the experiments, the character types in Thai language provide enough effective information for classifying the character into either the word beginning or word ending class.

III. Thai Word Segmentation Approaches

In this section, we give the details of two main approaches for word segmentation: dictionary based (DCB) and machine learning based (MLB). The DCB approach is based on the string parsing technique in which series of input characters are scanned and matched against the word set from a dictionary. In this paper, we evaluate two selection algorithms for solving the ambiguity problem. The first algorithm is by selecting the longest possible term, i.e., longest matching (LM). The second algorithm is by selecting the segmented series which yields the minimum number of word tokens, i.e., maximal matching (MM).

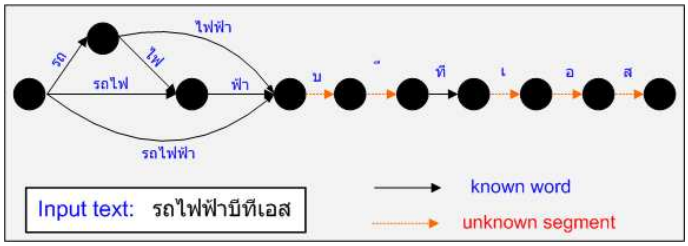

Fig. 1. Example of word segmentation using the DCB approach

Figure **??** illustrates an example of the word segmentation process. The given text can be translated to an English phrase, "BTS sky train". In the Thai language, a new word is often formed by combining a few words or lexemes into a compound word. As a result, an ambiguity problem occurs when there is more than one way to segment the text. From the example, the word "sky train" could be segmented four different ways. Both LM and MM algorithms will select the bottommost path of the parse tree which has only one segment. The reasons are that LM prefers the longest matching words and MM prefers the minimum number of segments. Another problem which is illustrated in this example is the unknown word problem. The word "BTS" is not a Thai word and therefore, not stored in the dictionary. Thai people usually transliterate foreign words phonetically. As a result, the transliterated Thai word of "BTS" is incorrectly segmented into single characters and small lexemes. In summary, the DCB approach is fast

and simple. The performance of DCB could be improved by adding new unknown words or domain-specific vocabularies into the dictionary used for segmentation process.

MLB approaches characterize word segmentation as a binary classification problem and adopt machine learning algorithms to build accurate classifiers. Each character in the input is predicted to be one of two classes: the beginning of a word (labeled as class "B" in our corpus) and intra-word characters (labeled as class "I" in our corpus). Figure **??** illustrates an example of a text string where each character is tagged with either word-beginning or intra-word characters. Using the tagged corpus in which word boundaries are explicitly marked with a special character, a machine learning algorithm can be applied to train a model based on the features surrounding these boundaries. For the Thai language, we can distinguish the character types for segmentation task into ten different types as shown in Figure **??**.

ท	c	B
่	t	I
า	v	I
น	c	I
ผ	n	B
ู	v	I
้	t	I
ฟ	c	I
ั	v	I
ง	c	I
ค	c	B
ะ	v	I

Fig. 2. Example of a string of characters tagged as word-beginning (B) or intra-word (I) characters

The set of character types is designed based on linguistic knowledge that is different across different languages. Machine learning algorithms can build classification models by extracting patterns of character types. For example, the algorithm could learn that the character whose type is vowel that can begin a word will most likely begin a word, i.e., class B. The final feature set for constructing a model is the n-gram of characters preceding and following the word boundary with their character types. In the experiments, we evaluate the algorithms by varying n from 3 to 11. In summary, MLB approaches are independent of any dictionary. The unknown word and ambiguity problems are handled by the algorithm which learns various character patterns inside a tagged corpus.

IV. Experiments and Discussion

We used the ORCHID corpus to evaluate the performance of the DCB and MLB approaches previously described. The ORCHID corpus contains 113,404 manually tagged words [**?**]. We transformed the ORCHID corpus using the n-grams of different character types as explained in the previous section. Each character was assigned to one of the following classes: beginning of a word characters (class B) and intra-word characters (class I). To evaluate the performance of all

Tag	Type	Example
c	Character that can be the final consonant in a word	กขฃคฅฆงจฉชซฌญฎฏฐฑฒฑฒณดตถทธนบปพฟภมยรลวศษสห์อ
n	Character that cannot be the final consonant in a word	ค ฉ ผ ฝ ฌ ห ฮ
v	Vowel that cannot begin a word	ะ ิ ี ึ ื ุ ู ั า ำ ๅ
w	Vowel that can begin a word	เ แ โ ใ ไ
t	Tonal character	่ ้ ๊ ๋
s	Symbol	ฯ ๆ ฯ.
d	Digit character	0-9
q	Quote character	'-' "-"
p	Space character inside a word	_
o	Other character	A-z

Fig. 3. Character types for building a feature set used by the MLB approach

approaches, we performed 10-fold cross validation on the data set.

We first evaluated four algorithms from MLB approach. For the naive Bayes and decision tree algorithms, we used the WEKA tool, which provides these two algorithms among many other machine learning algorithms [**?**]. Naive Bayes is denoted with *NB* and decision tree is denoted with *J48*. For the support vector machine algorithm, we used the LIBSVM, a library for training and testing support vector machines [**?**]. We set the kernel function to be polynomial with the power of 3 during the training process. For the conditional random field algorithm, we used CRF++, an open source implementation of the CRF algorithm [**?**]. We set the frequency flag equal to 3 which ignores the features whose frequency values are less than 3.

Table **??** presents the performance evaluation based on precision, recall and F1 measure. We tested with feature sets based on n-grams of size 3, 5, 7, 9 and 11. Figure **??** plots the F1 measures for all algorithms based on different n-grams. We observe that the performance of all the algorithms improves as n increases, since larger n-grams means the learner has more contextual information to base its classification on. We see that the CRF yields the best performance. This is consistent with previous work finding that the CRF is ideal for segmenting and labeling sequence data.

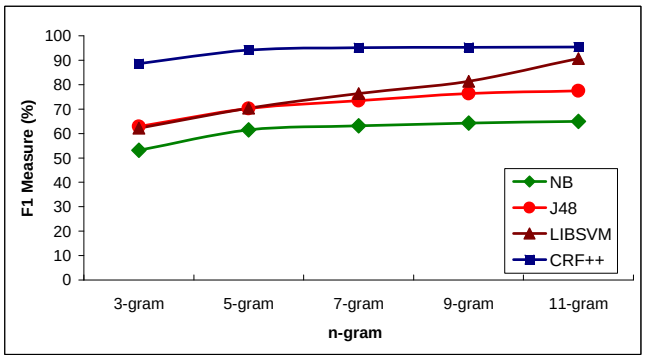

Fig. 4. MLB performance with different number of n-grams

TABLE I
EVALUATION RESULTS FOR MLB APPROACH

	3-gram			5-gram			7-gram			9-gram			11-gram		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
NB	74.60	41.20	**53.10**	68.20	56.10	**61.50**	70.70	57.00	**63.10**	69.50	59.70	**64.20**	69.70	60.60	**64.90**
J48	73.90	54.60	**62.80**	73.40	67.30	**70.30**	77.90	69.50	**73.50**	79.00	73.90	**76.40**	80.10	75.10	**77.50**
LIBSVM	75.91	52.59	**62.14**	73.48	67.45	**70.33**	80.11	72.88	**76.32**	86.49	76.93	**81.43**	92.87	88.71	**90.74**
CRF++	89.92	87.28	**88.58**	95.05	93.40	**94.22**	95.59	94.63	**95.10**	95.61	94.84	**95.23**	95.79	94.98	**95.38**

For the DCB approach, we used SWATH, which provides both LM and MM algorithms [**?**]. We used LEX*i*TRON, which contains approximately *30,000* words, as the main dictionary [**?**]. To observe the effect of using a domain-specific word set, we compared the performance of LM and MM using both the general word set from LEX*i*TRON (denoted by *-Lexitron*) and by using the word set obtained from the training corpus (denoted by *-Domain*). Figure **??** shows the results of the DCB approaches and the MLB approaches. We observe that the MM algorithm yields slightly better performance than the LM algorithm. Another interesting observation is using the word set obtained from the training corpus significantly improves the performance of both algorithms. Therefore, to get the best performance from DCB approaches, we can conclude that the dictionary should have large coverage and contain the domain-specific words related to the text being segmented.

In summary, both LM and MM from the DCB approach achieved better performance than the naive Bayes and decision tree algorithms from the MLB approach. However, the best overall performance was obtained with the CRF algorithm, with precision of 95.79% and recall of 94.98%.

TABLE II
RESULT COMPARISON BETWEEN DLB AND MLB APPROACHES

Approach	Algorithm	P	R	F1
DCB	LM-Lexitron	88.21	86.91	**87.55**
	LM-Domain	95.20	88.55	**91.75**
	MM-Lexitron	88.34	87.39	**87.86**
	MM-Domain	95.27	88.92	**91.98**
MLB	NB	69.70	60.60	**64.90**
	J48	80.10	75.10	**77.50**
	SVM	92.87	88.71	**90.74**
	CRF	95.79	94.98	**95.38**

V. CONCLUSION AND FUTURE WORK

We performed a comparative study of dictionary based and machine learning based approaches to Thai word segmentation. We evaluated two dictionary based algorithms, longest matching and maximal matching, and four machine learning based algorithms, Naive Bayes (NB), decision tree, Support Vector Machine (SVM), and Conditional Random Field (CRF), in the experiments. Using the ORCHID corpus as the data set, we measured the algorithms' performance in terms of precision, recall and F1 measure. The dictionary based approaches using domain-specific dictionaries derived from the training corpus performed better than the NB, decision tree and SVM algorithms. However, the best result overall was obtained from the CRF algorithm, with a precision and recall of 95.79% and 94.98%, respectively.

In future work, we plan to achieve better performance by integrating both dictionary based and machine learning based approaches. Dictionary based approaches perform well for known words, but machine learning methods are better when unknown words and ambiguous segmentations exist. In principle, by combining these two approaches, we may achieve better word segmentation results.

REFERENCES

[1] P. Charoenpornsawat, "Feature-based Thai Word Segmentation," Masters Thesis, Computer Engineering, Chulalongkorn University, 1999.

[2] C. Haruechaiyasak et al., "A collaborative framework for collecting Thai unknown words from the web," *Proc. of the COLING/ACL on Main Conference Poster Sessions*, pp. 345–352, 2006.

[3] A. Kawtrakul and C. Thumkanon, "A Statistical Approach to Thai Morphological Analyzer," *Proc. of the 5th Workshop on Very Large Corpora,* pp. 289–286, 1997.

[4] C. S. G. Khoo and T. E. Loh, "Using statistical and contextual information to identify two-and three-character words in Chinese text," *J. of the American Society for Information Science and Technology*, 53(5):365-377, 2002.

[5] C. Kruengkrai and H. Isahara, "A conditional random field framework for thai morphological analysis," *Proc. of the Fifth Int. Conf. on Language Resources and Evaluation (LREC-2006)*, 2006.

[6] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to japanese morphological analysis," *Proc. of EMNLP*, 2004.

[7] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *Proc. of the Eighteenth Int. Conf. on Machine Learning (ICML)*, pp. 282–289, 2001.

[8] S. Meknavin, P. Charoenpornsawat, and B. Kijsirikul, "Feature-Based Thai Word Segmentation," *Proc. of NLPRS'97*, pp. 289–296, 1997.

[9] F. Peng, F. Feng, and A. McCallum, "Chinese segmentation and new word detection using conditional random fields," *In Proc. of the 20th Int. Conf. on Computational Linguistics(COLING)*, 2004.

[10] Yuen Poowarawan, "Dictionary-based Thai Syllable Separation," *Proceedings of the Ninth Electronics Engineering Conference*, 1986.

[11] V. Sornlertlamvanich, "Word Segmentation for Thai in Machine Translation System," *Machine Translation*, National Electronics and Computer Technology Center, Bangkok.

[12] "CRF++, Yet Another CRF Toolkit",
Available at: `http://crfpp.sourceforge.net`

[13] "LEXiTRON Thai-English Dictionary",
Available at: `http://lexitron.nectec.or.th`

[14] "LIBSVM: A Library for Support Vector Machines",
Available at: `http://www.csie.ntu.edu.tw/ cjlin/libsvm`

[15] "ORCHID Corpus",
Available at: `http://links.nectec.or.th/orchid`

[16] "SWATH: Smart Word Analysis for THai",
Available at: `http://links.nectec.or.th/download.php`

[17] "WEKA: Waikato Environment for Knowledge Analysis",
Available at: `http://www.cs.waikato.ac.nz/ml/weka/`