WORD SEGMENTATION AND PART-OF-SPEECH TAGGING FOR THAI LANGUAGE USING MINIMUM TEXT AND CONDITIONAL RANDOM FIELD

Kannikar Paripremkul

A Dissertation Submitted in Partial
Fulfillment of the Requirements for the Degree of
Doctor of Philosophy (Computer Science and Information Systems)
School of Applied Statistics
National Institute of Development Administration
2020

WORD SEGMENTATION AND PART-OF-SPEECH TAGGING FOR THAI LANGUAGE USING MINIMUM TEXT AND CONDITIONAL RANDOM FIELD

Kannikar Paripremkul
School of Applied Statistics

..... Major Advisor
(Associate Professor Ohm Sornil, Ph.D.)

The Examining Committee Approved This Dissertation Submitted in Partial Fulfillment of Requirements for the Degree of Doctor of Philosophy (Computer Science and Information Systems).

..... Committee Chairperson
(Associate Professor Surapong Auwatanamongkol, Ph.D.)

..... Committee
(Associate Professor Nithinant Thammakoranonta, Ph.D.)

..... Committee
(Associate Professor Ohm Sornil, Ph.D.)

..... Committee
(Assistant Professor Orawan Chaowalit, Ph.D.)

..... Dean
(Assistant Professor Pramote Luenam, Ph.D.)

_____/_____/_____

ABSTRACT

Title of Dissertation WORD SEGMENTATION AND PART-OF-SPEECH TAGGING FOR THAI LANGUAGE USING MINIMUM TEXT AND CONDITIONAL RANDOM FIELD

Author Kannikar Paripremkul

Degree Doctor of Philosophy (Computer Science and Information Systems)

Year 2020

Thai word segmentation and Part-of-Speech (POS) tagging is still a very active research area. However, previous studies mostly focus on rule-based models or generative models such as the Hidden Markov Model (HMM), which may not suitable for segmenting an unknown word. In this research, we present a novel technique to deal with the problem of word segmentation for a language without explicit word boundary delimiters, like Thai, Chinese, or Korean. This research proposes a machine learning model called the Conditional Random Field (CRF) to segment Thai formal and informal words, including unknown words, teen slang, and loanwords.

To avoid word ambiguity, the word segmentation method is separated into three parts: (1) Minimum Text Unit (MTU) segmentation (the smallest unit of a Thai word), (2) syllable segmentation, and (3) word segmentation. In word segmentation, Longest Matching with pattern rules is used to assign word units. Pattern rules that follow Thai language structure for combining characters are also created to avoid segmentation errors. In order to select features for the CRF, existing research and the Thai language system are evaluated. For the character features of the CRF, we present both a general character and more fine-grained levels of vowels—front vowels, for example, can be separated into two categories: (a) front vowels that can have other characters placed in front of them and (b) front vowel that cannot have other character placed in front of them.

In the POS tagging procedure, each word is assigned a POS tag by the CRF model. POS tags are revised from an existing corpus to reduce the complexity of

usage by grouping uncertain POS tags together. Training data from this existing corpus is re-segmented using the proposed word segmentation method, primarily focusing on the accuracy of word units according to the official Thai dictionary. For the features used in the POS tagging, we experiment with several options and chose those features that were found to be best suited for the CRF method.

The performance of the proposed techniques is evaluated using common measurements, namely precision, recall, and F-score. The results are also compared to those of other state-of-the-art methods. In word segmentation, our proposed techniques are compared to a system using a convolutional neural network (CNN) that segments text to words. In terms of POS tagging performance, we compare our techniques to a well-known open API for the Thai language called PythaiNLP, which uses a perceptron algorithm for tagging parts of speech. The approaches proposed by this research are proven successful by high scores in all test data, especially in word segmentation. Our analysis also suggests a need to collect more training data, which may improve segmentation accuracy as well as the results of POS tagging, since both parts of the model are related.

ACKNOWLEDGEMENTS

I am so grateful to many people who have been supporting me to complete my Ph.D.

First, I would like to thank my advisor who supported, motivated, and believed in me. Thank you for your endless inspiring and knowledge throughout the years. I have learnt so much. Thank you again for your patience and understanding on me. In difficult time when it is hardly to move on but with your guidance and advice made it easier. I truly appreciated with your supports.

A special thanks to Dr.Surapong Auwatanamongkol, committee chairperson and my teacher, for given many interesting comments and suggestions to fulfilled the detail of this research during the progress examinations.

My committee members, Dr.Nithinant Thamakornnonta and the external committee Dr.Orawan Chaowalit, have been inspired on me during the thesis defense examination, and I also thank them for their ongoing support and feedback.

Many thanks to my Ph.D. friends for their help, advice, and support. A huge thank you to all my close and amazing friends who have supported and always by my side during the hard times.

I would also like to thank the staffs of Graduate School of Applied Statistics who have helped me in various times of need and provided administrative support.

Finally, I extend a huge thank to my family for their infinite support and understanding throughout this journey and beyond.

Kannikar Paripremkul

June 2021

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

1.1 Statement of Problem

A sentence without word delimiters can be segmented into words in different manners, giving different meanings. In languages with no word delimiters, such as Chinese, Japanese, Korean and Thai, the problem of word boundary ambiguity may lead to incorrect segmentation. For example, “นั่นมือถืออะไร” can be segmented into “นั่น|มือ|ถือ|อะไร” (What is in your hand?) and “นั่น|มือถือ|อะไร” (What brand is that mobile phone?), while “ผ้าไหมลายสวยมาก” can be segmented into “ผ้าไหม|ลาย|สวย|มาก” (This silk has a gorgeous pattern) and “ผ้า|ไห|มลาย|สวย|มาก” (This fabric and jar have been destroyed very nicely). For the first sentence, without any preceding or following sentences, the segmentation result can be either “นั่น|มือ|ถือ|อะไร” or “นั่น|มือถือ|อะไร” because of a lack of context. For the second sentence, the result should be “ผ้าไหม|ลาย|สวย|มาก”. Since “ผ้าไหม” (Silk cloth) is a compound word composed of “ผ้า” (Fabric) and “ไหม” (Silk), if it is segmented incorrectly, then the true meaning of the sentence and words may not be reached.

This problem is even more challenging nowadays, with styles of writing in social media that contain unknown words, informal words, slang, and words adopted from other languages. These words cannot be found in dictionaries but are understood among social media users, and new words are invented in a short time. Word segmentation methods that rely on dictionaries or are trained on formal corpora will not be able to handle these words correctly.

This research proposes a novel technique for Thai word segmentation by extracting Minimum Text Units (MTU), which are the smallest units that constitute words. These units are then used by the Conditional Random Field (CRF) to identify

syllables. Finally, words are segmented by merging syllables together with a set of rules for analyzing language characteristics.

In Natural Language Processing (NLP), part-of-speech tagging plays a role insignificant NLP tasks, such as speech recognition, information retrieval, text summarization, etc. A word is a group of characters or a single character. A sentence is constructed from a group of words using the grammar of a language. The grammatical roles of word are called part-of-speech (POS). Problems with NLP in the Thai language are the ambiguity of compound words, which are words that can have more than one POS tag depending on their position in the sentence, and the problem of unknown words.

Currently, several famous Thai POS tag corpora, such as ORCHID (Sornlertlamvanich, Charoenporn, & Isahara), require linguistic expertise to comprehend the tags. ORCHID contains 14 categories with 47 subcategories. For example, the category of determiners is divided into 9 subcategories. This high number of subcategories is only suitable for experienced users. For the typical user, our work is more appropriate for general use.

In this work, we employ the CRF, a machine learning algorithm for tagging POS for Thai words when the word units from the training and test data are different. Training data was collected from the widely used corpus for word segmentation and POS tagging research called ORCHID. Test data was collected from 3 sources: first, test data A consists of words from the word segmentation described above; second, test data B is the test set from the re-segment words in ORCHID; finally, test data C is the test set from the original words in ORCHID.

1.2 Objective

This study aims to propose and design a new, effective method of Thai word segmentation and POS tagging.

1.3 Scope and Limitation

The scope of this study covers words which have meaning in themselves and combined words which have different meanings from the original component words based on vocabulary from the official Thai dictionary. The study will also include words from foreign languages and unknown words which are written in Thai script.

For POS tagging, training data and word classes are adopted from an existing Thai corpus. The results of word segmentation in formal texts are used as a test set for the POS tagging.

1.4 Approach

In order to achieve the research objective, appropriate techniques and models were designed. For MTU segmentation and syllable segmentation, this research uses a CRF. For word segmentation, it uses the Longest Matching technique. To decrease the errors of segmentation in Longest Matching, the rules of Thai small unit patterns are generated. The CRF is also used to tag POS for Thai words. POS tags are modified from the existing Thai POS corpus, ORCHID, to reduce their complexity, which can lead to misuse.

Next, data are collected and pre-processed using an existing Thai corpus consisting of news and text from social media. For word segmentation, this research uses BEST2010 (National Electronics and Computer Technology Center (NECTEC), 2010) to train the model. ORCHID is used to train the model for POS tagging. Test data for word segmentation are divided into two types: formal text and informal text. For POS tagging, the results of the word segmentation of the formal text set are used as test data. In addition, ORCHID is also split into training and test sets to compare performance with uncovered data from formal text.

Finally, this study evaluates performance using precision, recall, and F-score. Both word segmentation and POS tagging result are compared with the popular Thai word segmentation and POS tagging techniques Deepcut (Kittinaradorn, Chaovavanich, Achakulvisut, & Kaewkasi, 2019) and PythaiNLP (Phatthiyaphaibun

et al., 2016). For word segmentation, Deepcut proposed a Convolutional Neural Network (CNN) using the BEST corpus as training data, while PythaiNLP uses the perceptron algorithm with the ORCHID corpus as training data for POS tagging.

CHAPTER 2

LITERATURE REVIEW

2.1 Characteristics of the Thai Language and Its Ambiguities.

Thai characters consist of 44 consonants, 18 basic vowels, 4 tone makers, and several other symbols. Thai also has its own numerals. The Thai characters are shown in Table 2.1. Thai text does not have specific stop words or stop syllables.

Table 2.1 Thai Characters

Consonant	ก ข ฃ ค ฅ ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ
Vowel	ั ะ า ำ ๅ ิ ี ึ ื ุ ู เ แ โ ใ ไ ฺ ็ ฤ ฦ
Tone	่ ้ ๊ ๋ ์
Other symbol	ๆ ฯ ๎ ๏
Numeral	๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙

Several previous approaches studied Thai word segmentation on the word level, but the problem of ambiguity still occurs. Aroonmanakun (2002) discussed the definition of Thai words and noted that Thai words are normally created from one or more syllables. Therefore, researchers have resolved word ambiguity by employing syllable segmentation. Aroonmanakun studied Thai grammar books, revealing that words can be separated into two types. First, they can be simple words, those with one or more syllables that contain a meaning in themselves, or whose meanings are not linked, such as “นอน” (Sleep) and “นา-ฬิ-กา” (Clock). Second, they can be compound words. A compound word consists of two or more words which together constitute a

single new word with a specific meaning. Compound words may or may not be related to their component words. For example, “แม่น้ำ” (River) is related to its components “แม่” (Mother) and “น้ำ” (Water), while “ยินดี” (Glad) is not related to its components “ยิน” (Hear) and “ดี” (Good). Moreover, compound words generally comprise more than one similar word, such as “สวยงาม” (Elegant), which is a combination of “สวย” (Pretty) and “งาม” (Beautiful). For syllable segmentation, Aroonmanakun defines syllable patterns for Thai characters for matching text. With a basis in trigram analysis, the study can correct syllable segmentation up to 99.8% of the time.

Jucksriporn and Sornil (2011) proposed a syllable segmentation method, the Thai Minimum Clusters (TMCs) technique, to resolve unknown and ambiguous words. It creates units smaller than a syllable and then combines these units into syllables using a trigram statistical model. TMCs are adjusted based on the researchers’ experience and Thai writing rules. Figure 2.1 shows a diagram of the minimum clustering non-deterministic finite automata (NFA).

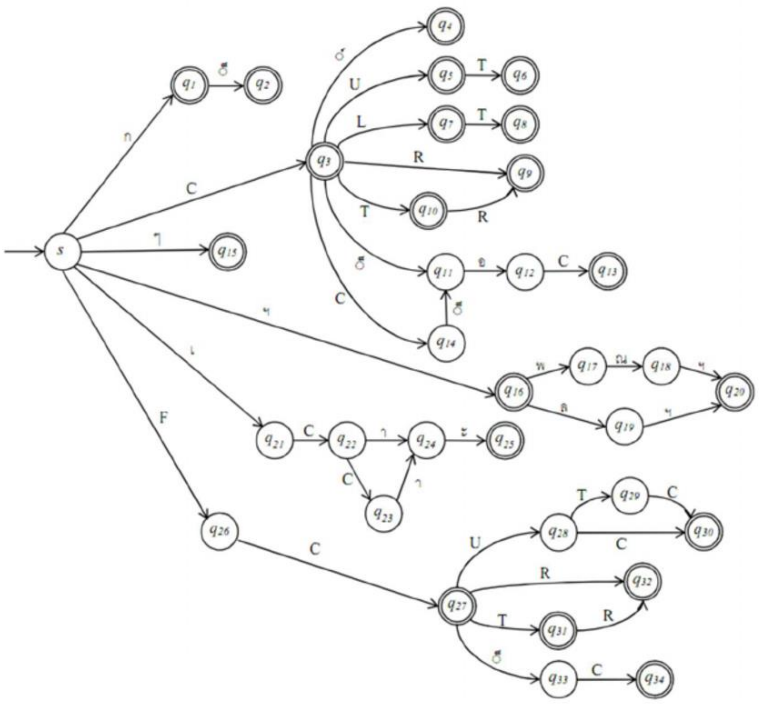

Figure 2.1 Minimum Clustering NFA
Source: Jucksriporn and Sornil (2011).

In terms of the rules for this NFA, Thai minimum clustering creates a strong sequence of units. Thus, the word “รักษา” (Heal) is segmented into “รัก - ษา” since “ ั” must always be followed by another character and “า” needs a character in front of it. This present research will modify TMCs to suit the Minimum Text Unit (MTU) segmentation and use TMCs to identify syllables.

2.2 Text Segmentation in the Thai Language

Narupiyakul, Thomas, Cercone, and Sirinaovakul (2004) developed Thai information extraction at the syllable level based on a review of many studies relating to the syllable method. It was approved that syllable segmentation provides high accuracy compared to Thai word segmentation. They devised syllable segmentation to separate the input information into small units. The syllable segmentation process follows a rule-based approach and uses statistical extraction using the Hidden Markov Method (HMM). Their conclusion showed that the tagging process, which provides prefix and suffix tags, significantly affects the accuracy of the experiment.

Aroonmanakun (2002) proposed a Thai word segmentation approach based on a trigram model and maximum collocation. This paper found that ambiguity in word segmentation can be solved by inserting syllable boundaries. The segmentation can be separated into two processes. The first is syllable segmentation, which applies a trigram model on a trained corpus which is then segmented into syllables. Aroonmanakun defined patterns for syllables and matched them against input data. However, the results were still ambiguous until the trigram model was applied. The second process is merging syllables into word units. This step uses collocation strength to merge syllables along with a dictionary to determine a sequence of syllables. Thus, if unknown words appear in the input sentence then the segmentation may be incorrect.

Theeramunkong, Sornlertlamvanich, Tanhermhong, and Chinnan (2000) presented the concept of the Thai Character Cluster (TCC) for retrieving Thai text. They argued that TCC is not perform ambiguous group of characters. TCC was created from Thai writing rules to segment a sequence of characters into inseparable

units that can be smaller than words, but it cannot divide text further than this. Experiments indicated that TCC can improve search performance. Limcharoen, Nattee, and Theeramunkong (2009) also used TCC with a two-phase GLR parsing technique. This technique was separated into candidate generation and selection processes. The first phase of GLR generates a candidate sequence of TCCs by parsing an input character using Context-Free Grammar (CFG). Next, it outputs all the word segmentation candidates. To generate a word, it uses an N-gram model to select the most suitable segmentation from among the candidates. However, the study found that this technique cannot correctly handle unknown words.

Theeramunkong and Usanavasin (2001) proposed a decision tree for word segmentation to avoid dependency on a dictionary. TCC is used as training corpus. The TCC corpus contains five sets, four for training and one for testing. Certainty factor (CF) is used to calculate numerical measures of confidence. At CF of approximately 80%, the accuracy continually drops because the chance of incorrect word segmentation has risen. The best performance is at a CF of 70%, where accuracy, precision, and recall are 87.41%, 91.92%, and 96.13%, respectively. The researchers suggested incorporating a dictionary into the method to improve accuracy when segmenting unknown words. This research also carried out preliminary experiments using Maximum Matching and Longest Matching methods for segmentation. The results showed the performance of each method against unknown words. Maximum Matching had 54.01% accuracy when the corpus contained 50% unknown words. The accuracy of Longest Matching was lower, 48.67% given 50% unknown words; otherwise, the accuracy reached 97.03%. Thus, the presence of unknown words significantly affect word segmentation using dictionary-based approaches.

2.3 Segmentation Using Conditional Random Field

Liu, Nuo, Ma, Wu, and He (2011) proposed an approach to segmenting Tibetan words. The Tibetan writing structure contains a stop maker at each syllable, called “tsheg,” and a sentence boundary marker, called “shed.” In addition, one or

more syllables may be combined into a word. The writing structure of this language is comprised of syllables and sentence separators, yet it is still difficult to define a word boundary. Figure 2.2 shows the two-syllable Tibetan word meaning “exhibition.”

Figure 2.2 Structure of a Tibetan Word
Source: Liu et al. (2011).

In the Tibetan word structure, syllables can be placed in different positions, as explained. From Figure 2.3, the researcher generated tags for each syllable position, with a total of 4 tags.

S: if the syllable forms an entire word.

B: if it is on the left side of a word.

M: if it is in the middle of a word.

E: if it is on the right side of a word.

Position	Example	Meaning	Tag
Word by itself	ཁ་	mouth	S
Begin	ཁ་གསབ་	supplement	B
Middle	མི་ཁ་ཤས་	someone	M
End	གླུ་ཁ་	ferry	E

Figure 2.3 Position of Syllable in Word
Source: Liu et al. (2011).

Template Set	Type	Feature	Function
TMPT-6	Unigram	$C_n,\ n = -1, 0, 1$	The previous, current and next syllable
	Bigram	$C_nC_{n+1},\ n = -1, 0$	The previous (next) syllable and current syllable
		$C_{-1}C_1$	The previous syllable and next syllable
TMPT-10	Unigram	$C_n,\ n = -1, 0, 1$	The previous, current and next syllable
		C_{-2}	*The syllable before the previous syllable*
		C_2	*The syllable after the next syllable*
	Bigram	$C_nC_{n+1},\ n = -1, 0$	The previous (next) syllable and current syllable
		$C_{-1}C_1$	The previous syllable and next syllable
		C_1C_2	*The next two syllables*
		$C_{-2}C_{-1}$	*The previous two syllables*

Figure 2.4 Feature Template TMPT-6 and TMPT-10
Source: Liu et al. (2011).

The tags were used in a Conditional Random Field (CRF) for word segmentation. The feature templates used in this study were TMPT-6 and TMPT-10, which are also compared against each other. This comparison revealed that the performance of TMPT-6 is better than that of TMPT-10, according to their F-scores.

In Chinese 2005, the Special Interest Group for Chinese Language Processing of the Association for Computational Linguistics (ACL) conducted the second International Chinese Word Segmentation Bakeoff, a competition in Chinese word segmentation, to update current word segmentation techniques. This then came to be the general benchmark for Chinese segmentation. The results from this competition showed that the best F-score, 0.972, came from open tests while the best score from closed tests was 0.964. (Emerson, 2005)

Zhao, Huang, Li, and Lu (2006) studied the performance of Chinese word segmentation in the Bakeoff-2003 and Bakeoff-2005 competitions. They found that the error rate was 3.9% in Bakeoff-2003 and then down to 2.8% in 2005. The research with the best results used CRF and Maximum Entropy. This work aimed to improve the performance of CRF by considering the feature template and tag set. The feature templates are unigram, bigram, punctuation and character-based which are numbers, characters that meaning are dates English letters and other characters. Tag sets were redesigned based on previous work from Bakeoff-2005. To deal with long words, 4 tag sets were revised into 5 tag sets and 5 tag sets were revised into 6 tag sets. The experiment is also performed 2 tag set and 4 tag set used in previous works. The tag sets are shown in Table 2.2. The results of word segmentation showed that CRF with

the new feature template and appropriate tag sets could improve the performance of Chinese word segmentation.

Table 2.2 The Tag Set of Chinese Word Segmentation

Tag Set	Tags
2-tag	B, E
4-tag	B, M, E, S
5-tag	B, B2, M, E, S
6-tag	B, B2, B3, M, E, S

Source: Zhao et al. (2006).

Kudo, Yamamoto, and Matsumoto (2004) studied morphological analysis in Japanese and compared the performance of Hidden Markov Models (HMMs), Maximum Entropy Markov Models (MEMMs), and Conditional Random Fields (CRFs). F-score results for word segmentation based on the Kyoto University corpus using HMMs, MEMMs, and CRFs were 96.22%, 96.44%, and 98.96%, respectively. This confirmed that CRFs can solve the problem of word boundary ambiguity. This is because CRFs can include related features but MEMMs cannot without being affected by the label bias problem.

Haruechaiyasak, Kongyoung, and Dailey (2008) compare two approaches: dictionary-based and machine learning–based. Their results showed that CRF, which is a machine learning–based algorithm, achieved the highest score in comparison with other algorithms. The input for the algorithm was a character. Taking a character as input, the algorithm predicted whether it came from the beginning of a word or from the middle. The features used for this study were 10 Thai character patterns. In addition, the authors claimed that CRF can handle unknown words and word ambiguity by learning the patterns of words from the ORCHID corpus. The F-score, evaluated on an 11-gram, was 95.38%.

Haruechaiyasak and Kongyoung (2009) proposed word segmentation to avoid the ambiguity problem and the dictionary-based approach. Their research generated three feature sets, “char,” “char-type” and “combined,” to be used in CRFs. The first

feature set, char, comprises the actual characters of the Thai script. Char-type identifies the category to which Thai characters belong. The combined feature set combines “char” and “char-type.” Examples of combined feature sets are shown in Figure 2.6.

Tag	Type	Value
c	Consonant characters which can be assigned as the word ending character	ก ข ค ม ง จ ช ซ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป พ ฟ ภ ม ย ร ล ว ศ ษ ส ฬ อ
n	Consonant characters which **cannot** be assigned as the word ending character	ค ฉ ผ ฝ ฌ ห ฮ
v	Vowel characters which are **not** allowed to begin a word	ะ ◌ั ◌ิ ◌ี ◌ึ ◌ื ◌ุ ◌ู ◌ำ ◌ๅ ◌็
w	Vowel characters which are allowed to begin a word	เ แ โ ใ ไ
t	Tonal characters	◌่ ◌้ ◌๊ ◌๋
s	Symbol characters	◌์ ๆ ฯ .
d	Digit characters	0-9
q	Quote characters	'-' "-"
p	Space character within a word	_
o	Other characters	a-z A-Z

Figure 2.5 Character Categories
Source: Haruechaiyasak and Kongyoung (2009).

As seen in Figure 2.6, the last column seeks to predict the label of a word, where “B” is a word-beginning character and “I” is intra-word character. The feature sets are trained separately in the same environment. As a result, the combined feature set has the highest F_1 value of the three feature sets, while char-type has the worst performance.

เ	w	B
ร	c	I
ี	v	I
่	t	I
อ	c	I
ง	c	I
"	q	B
บ	c	B
ท	c	I
น	c	B
ำ	v	I
:	o	B

Figure 2.6 Text String Labelled Based on Combined Feature Set
Source: Haruechaiyasak and Kongyoung (2009).

2.4 Part-of-Speech Tagging Techniques

This section discusses previous problems and techniques for Part-of-Speech (POS) tagging. While many techniques have been used for POS tagging, such as rule-based systems or machine learning, POS tagging in Thai mostly uses machine learning techniques. One famous Thai word and POS tag corpus is ORCHID. This uses a probabilistic trigram model to estimate POS tags for each word. This present work also classifies the problems of Thai POS tagging arising from the fact that Thai words may have more than one tag without changing their appearance. For example, consider "ฉันไปโรงเรียน" (I will go to school) and "พระสงฆ์ฉันเพล" (The monk is having lunch), where the POS tag of the word "ฉัน" can have different classes and meaning depending on the part it plays in a sentence, as shown in Table 2.3. This table shows the unchanging appearance of the word "ฉัน," even though the first sentence uses it as a pronoun which means "I / Me", and the second sentence turns it into a verb and the meaning is "Eat".

Table 2.3 Example of Unchanging Word Appearance in the Different Location in Sentence

First Sentence	ฉันไปโรงเรียน (I am going to school)	Second Sentence	พระสงฆ์ฉันเพล (the monk is having lunch)
	ฉัน | personal pronoun		พระสงฆ์ | personal pronoun
	ไป | verb		ฉัน | verb
	โรงเรียน | common noun		เพล | noun

Boonkwan, Supnithi, Pailai, and Kongkachandra (2013) use Support Vector Machines (SVMs) and CRFs models and show promising accuracy when correcting frequent POS tag errors using local retagging. Their experiment results, evaluated on the BEST2012 corpus, show that the local retagging technique can improve F-score accuracy from 96.46% to 97.82%. CRFs outperforms SVMs in both models (with and without the retagging technique).

Research by Pailai, Kongkachandra, Supnithi, and Boonkwan (2013) proposed SVMs and CRFs with forward trigrams, backward trigrams, and 5-grams as features. This study also used the BEST2010 corpus with 10-fold cross-validation. The results showed that the two techniques are comparably accurate. The SVMs with the forward trigram feature achieved the best score at 93.64%, while the CRFs with the forward 5-gram feature scored 93.25%. This study suggested that the size of the training data significantly effects results, since the models that achieved the highest accuracy scores for both techniques were trained on a corpus containing 100,000 tokens, the highest amount of training data.

Likewise, POS tagging has been used with three machine learning methods on the Thai language, with experiments performed on four training datasets, differing in size. The three models were a Decision List, Maximum Entropy, and an SVM. With full training data, the SVM outperformed Maximum Entropy and the Decision List in terms of precision, at 96.1%, 95.3%, and 83.6%, respectively. When the amount of training data was decreased to half, one-quarter, and one-eighth of the full dataset, the precision scores continually decreased. (Murata, Ma, & Isahara, 2002)

POS tagging has been the subject of various research about other languages which have are delimiters for word boundaries, like Thai. Many studies also proposed CRF models for tagging word classes in such language as Chinese, Arabic, and Korean. A study of Chinese POS tagging by Xiong et al. (2019) proposed CRF and Bidirectional long short-term memory (BiLSTM) with a CRF layer for word segmentation and POS tagging. The electronic health record system, excluding fragmentary clinical notes, was used as a corpus, consisting of 198,797 words. The corpus was divided into training, development, and test sets. CRF outperformed BiLSTM on both tasks. In POS tagging, the F-score of CRF was 95.11%, versus 94.77% for BiLSTM.

Darwish et al. (2018) proposed a POS tagging method for Arabic tweets in multiple dialects. The four dialects used in this work were the major dialects of Arabic, namely Egyptian, Levantine, Gulf, and Maghrebi. The dataset consisted of 350 tweets in each dialect, with the number of words in the range 6,400 - 7,481. Five-fold validation and a CRF model were used for the experiment, which also examined the performance of cross and joint dialects. There were 24 POS tags, including 4

tweet-specific tags. The results showed that training sets with the same dialect gave better scores than training sets with different dialects.

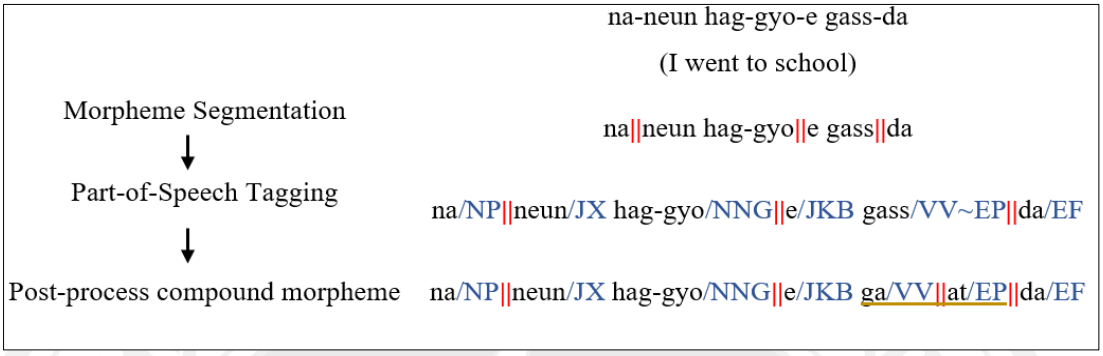

Figure 2.7 The Process of Korean Morphological Analysis and Part-of-Speech Tagging
Source: Adapted from Na (2015).

Na (2015) studied morpheme segmentation and POS tagging for the Korean language. This research presented a two-stage approach for Korean morphological analysis and POS tagging, using CRFs for three processes, as shown in Figure 2.7. In the morpheme segmentation process, compound morphemes are extracted from the original POS-tagged corpus to train a CRF model, and then segmented at syllable level. For morpheme tagging, the labels B and I are used to indicate the boundaries of morpheme segmentation, where label B refers to the beginning of a morpheme and label I refers to the inside of a morpheme. Table 2.4 shows the types of CRF features for morpheme segmentation; C_i is a syllable located at previous syllable before the current syllable C_0, S_i is a space occurs just after C_i. The feature for the POS tagging process is $[W_{-1}, W_0, W_1]$ where W is a morpheme. The POS tagging was based on the sequence of morphemes using the CRF method.

Table 2.4 Features for Morpheme Segmentation Using Conditional Random Field

Feature Description	Feature Symbol
Uni-syllable	$C_{-2}, C_{-1}, C_0, C_1, C_2$
Bi-syllable	$C_{-1}C_0, C_0C_1, C_1C_2$
Tri-syllable	$C_{-2}C_{-1}C_0, C_{-1}C_0C_1, C_0C_1C_2$
Uni-spacing	$S_{-2}, S_{-1}, S_0, S_1, S_2$
Bi-spacing	$S_{-1}S_0, S_0S_1, S_1S_2$
Tri-spacing	$S_{-2}S_{-1}S_0, S_{-1}S_0S_1, S_0S_1S_2$

Source: Na (2015).

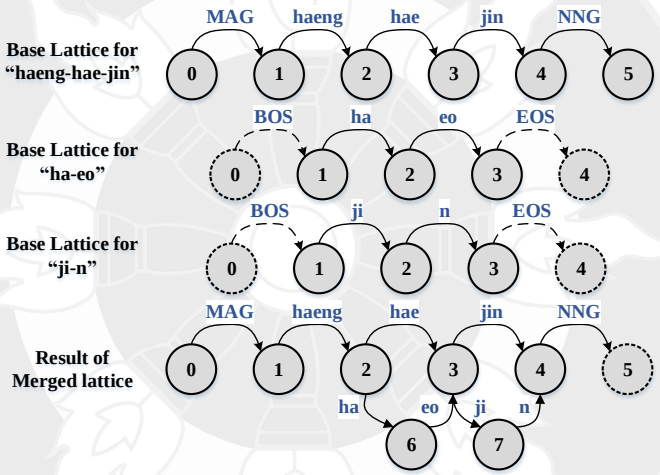

Figure 2.8 Example of a Base Lattice HMM for a Decomposed Compound Morpheme

Source: Adapted from Na (2015).

Finally, this study also decomposes compound morphemes into atomic morphemes. The post-processing of a compound morpheme consists of two steps: (1) pre-analyzed patterns of compound morphemes. This step is carried out while extracting compound morphemes from the original POS-tagged corpus during the morpheme segmentation process. (2) a lattice HMM, to handle unknown patterns of compound morphemes from the pre-analyzed step. Figure 2.8 shows an example of a lattice HMM for the compound morpheme "haeng-hae-jin," which means "an attempt

made much.” Since the lemma form of “hae” is “ha-eo”, and that of “jin” is “ji-n,” the vertices of these lemmas are extended and merged into a base lattice.

The CRF model for this research used the SEJONG and ETRI corpora which contain 3,466,378 and 2,392,210 words with 42 and 72 POS tags, respectively. Training and Test data were separated into 80% for training and 20% for testing. Morphological analysis result showed that with the F-score for the SEJONG corpus outperforms that for the ETRI corpus. In SEJONG, the F-score for morpheme segmentation was 98.41%, with 98.12% POS tagging performance. In ETRI, these figures were 94.65% and 97.08%, respectively. In addition, the results for morphological analysis using pre-analyzed patterns and morphological analysis using pre-analyzed patterns and lattice HMMs for SEJONG and ETRI corpus were 96.23% and 90.66%, respectively. ETRI performed worse than SEJONG in all results.

CHAPTER 3

WORD SEGMENTATION METHODOLOGY

This chapter describes the proposed method of segmentation in detail in section 3.1. The adaptation of Thai Character Cluster (TCC) and Thai Minimum Clusters (TMCs) will be discussed. The adapted unit, Minimum Text Unit (MTU), will be used as a token to create a segmentation model for syllables and create the pattern rules for word segmentation. Section 3.2 will briefly explain related algorithms with Conditional Random Fields (CRFs). The corpora for word segmentation will be described in section 3.3. Finally, section 3.4 will describe the methods and techniques for each segmentation.

3.1 Text Clustering

Since this research focuses on the Thai writing system, the syllable segmentation method will be applied to the written system. For example, "มหาวิทยาลัย" (University) will be clustered into "มหา|วิท|ยา|ลัย" instead of "ม|หา|วิ|ท|ยา|ลัย," although the latter is the correct segmentation based on Thai pronunciation. However, to reduce the ambiguity of word segmentation, a smaller unit is needed. The concept of TMCs is interesting because the smallest units of TMCs and TCC can be used to define the boundary of a word.

The rules of TMCs were established to process Romanized Thai texts rather than those in the Thai script. However, this research needs a rule for the Thai writing system. To generate a new set of rules, the following pages examine the Thai writing system, with a focus on character structure. Thai characters can be separated into 9 types, adapted from Jucksriporn and Sornil (2011) and Haruechaiyasak et al. (2008) and shown in Table 3.1.

Table 3.1 Type of Thai Characters

Consonants	ก ข ฃ ค ฅ ฆ ง จ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป พ ฟ ภ ม ย ร ล ว ศ ษ ส ห อ
Non-Suffix Consonants	ค ฉ ฌ ผ ฝ ห ฮ
Leading Vowels	เ แ โ ใ ไ ฤ ฦ
Upper Vowels	ิ ี ึ ื
Special Vowels	ั ็
Rear Vowels	ะ า ำ ๅ
Lower Vowels	ุ ู
Tones	่ ้ ๊ ๋
Special Symbols	ๆ ๎ ์ ๏

These are rules for the Thai writing system:

1) Leading Vowels are always positioned before at least one consonant.

2) Upper Vowels and Lower Vowels always follow at least one consonant.

3) Special Vowels are always followed by at least one consonant and preceded by a consonant.

4) Rear Vowels always follow at least one consonant, with the exception of "ะ," which may follow "า."

5) If the Rear Vowel "ะ" follows "า," the Leading Vowel "เ" may be exactly two or three positions before "า".

6) A Tone may follow a Consonant, Upper Vowel, Special Vowel, or Lower Vowel, with one or more Consonant(s) behind it.

7) Leading Vowels and Rear Vowels may be combined, as in "เ-ะ," "แ-ะ," "โ-ะ," "เ-าะ," "แ-ะ," "โ-ะ," or "เ-า."

8) Combined Vowel (see 7) may include a tone.

9) Leading Vowels and Upper Vowels may be combined, and always follow a Consonant after an Upper Vowel.

10) The Leading Vowels "ฤ" and "ฦ" appear only as singletons or preceding "ๅ".

11) The Special Symbol "ๆ" is always a singleton.

12) The Special Symbol "ฯ" appears only in "ฯ," "ฯลฯ," or "ฯพณฯ."

13) The Special Symbol “ ์ ” always follows at least 3 characters.

14) The Special Symbol “ ์,” when preceded by a Leading Vowel, always follows and precedes at least one Consonant, except for “ก็.”

15) Special Symbol “ ์,” when not preceded by a Leading Vowel but followed by the consonant “อ,” is always followed by an additional consonant.

16) “ก็” and “บ” are exempt from the rules.

17) A Non-Suffix Consonant is not always positioned at the end of a word.

The TCC is a unit smaller than a word but larger than a character, helping to reduce the ambiguity of a word. In TCC, Thai characters are divided among three levels and there are seven types of character. The levels of Thai characters are the upper level, middle level, and lower level, depending on the position of each character. The character types are upper / lower / front / rear vowel, consonant, tone, and Karan (“ ์ ”). Figure 3.1 defines the character levels and types for the word “การเชื่อมต่ออุปกรณ์” (The equipment connection).

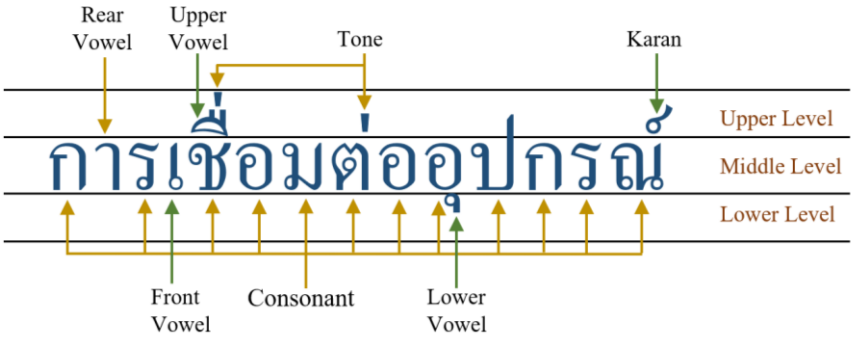

Figure 3.1 Example of Thai Levels and Character Types
Source: Adapted from Theeramunkong et al. (2004).

Below are the EBNF forms for TCC segmentation.

<TCC> → ‘ก็’ | ‘อี’ | ‘หี’
| <Cons> ‘รร’, <Cons> < ์ >
| <Cons> <BCons> <Cons> < ์ >
| <Cons> <TCC1> <Karan>

		| <FSara> <Cons> <TCC2> <Karan>
<TCC2>	→	<Cons> ‘ าะ ’
		| ‘ ็ ’ <BCons>
		| <USara> {<Tone>} <BCons> [‘ า ’ | ‘ ะ ’]
		| {<Tone>} [‘า’ | ‘ าะ ’ | ‘ ะ ’]
<TCC1>	→	<DSara> {<Tone>}
		| {Tone} ‘ ่า ’
		| [‘ ็ ’ | ‘ ็ ’] {<Tone>} <BCons>
		| ‘ ั ’ {<Tone> [‘ ์ ’ | ‘ ์ ’]}
		| ‘ ็ ’ <BCons>
		| <Tone> [<TSara> | <DSara>] {‘า’} <BCons>
		| ‘ ์ ’ {<Tone> {<BCons>}}
		| ‘ ็ ’ {<Tone>}
		| {<Tone>} <BSara>
		| NULL
<Karan>	→	<Cons> {<Cons>} {[<DSara> | ‘ ์ ’]} ‘ ์ ’
		| NULL

The concepts of TMC and TCC are adapted for MTU in section 3.3.1, and for pattern rules for word segmentation in section 3.3.3.

3.2 Algorithm for Sequence Learning

In Natural Language Processing (NLP), Machine Learning techniques are used in various problems: POS tagging, sentiment analysis, summarization, segmentation, etc. For text classification, a Machine Learning model learns assigned data to understand and analyze human language. Machine Learning methods with statistical modeling, such as the Hidden Markov Model (HMM), Maximum Entropy Markov Model (MEMM), and Conditional Random Field (CRF) are commonly applied to NLP tasks. To understand the segmentation and labelling of sequence text, the following subsection describes the frameworks of HMM, MEMM, and CRF techniques.

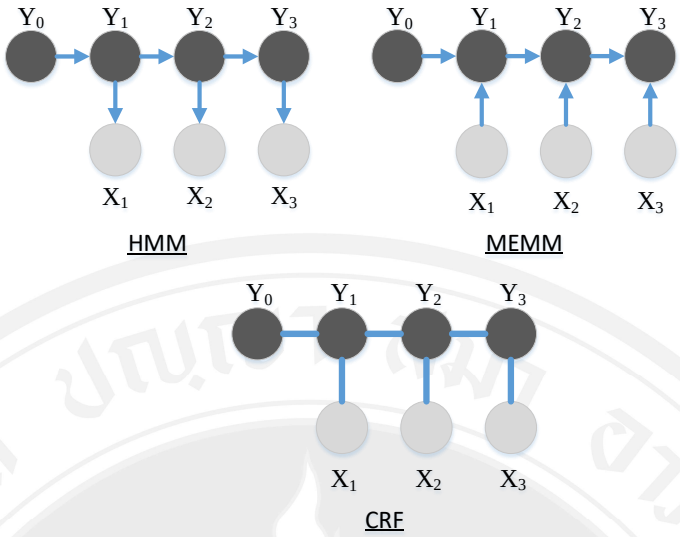

Figure 3.2 A Comparison of General CRF and Linear-Chain CRF Model

3.2.1 Hidden Markov Model (HMM)

$$P(y,x) = \prod_{t=1}^{T} P(y_t|y_{t-1})P(x_t|y_t) \quad (1)$$

The Hidden Markov Model (HMM) is a graphical probability model that can be used to stochastically assign POS tags or carry out word segmentation in a sentence. An HMM contains two probability matrices, state transition probability and emission probability. A Markov Model only considers the most recent state's probability in calculating the future state in its state transition. An HMM is a Markov Model with mainly hidden states. These are states whose probability is not directly calculated but only influence the state transition. In POS tagging, the hidden states are the POS tags and the probabilities that words in the sentence will be a given part of speech. With a high number of words and tags, the Viterbi algorithm is typically used to make the calculation tractable by finding the maximum likelihood over observations in a dataset.

3.2.2 Maximum Entropy Markov Model (MEMM)

$$P(y|x) = \prod_{i=1}^{n} \frac{exp(W, f(y_i, y_{i-1}, x))}{Z(y_{i-1}, x)} \quad (2)$$

The Maximum Entropy Markov Model (MEMM) is an HMM which uses Maximum Entropy (MaxEnt) for its likelihood. MaxEnt takes conditional probabilities of all observations in log-likelihood with one probability matrix rather than a joint probability of an observation of the current state, as in HMM. MEMM is better than HMM at capturing overall observation sequences. However, this overall observation ability causes a label bias problem when it prefers states with fewer transitions in order to maximize its conditional probabilities.

3.2.3 Conditional Random Field (CRF)

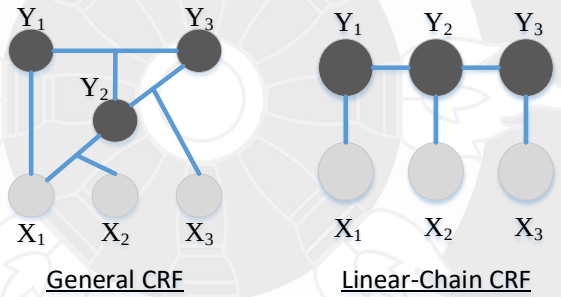

Figure 3.3 General CRF vs Linear-Chain CRF. Linear-Chain CRF is a Stricter Version of General CRF, Where the Nodes are Linearly Sequential

A Conditional Random Field (CRF) is very similar to an MEMM, but it uses global normalization to avoid the label bias problem. For sequential data, Linear-Chain CRF (as shown in Figure 3.3) is typically adopted. CRF employs the log-likelihood of conditional probabilities of all observations, as well as all possible states. Global normalization is performed using a sum of all possible sequences of states to ensure the maximum likelihood of all states.

The model for linear-chain CRF is explained in (3), where X is the input sequence and Y is the output label sequence. The conditional probability distribution is $P(Y|X)$, which can be written as follows:

$$P(Y|X) = \frac{1}{Z(x)} \exp\left(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, x, t)\right) \quad (3)$$

where λ_k is a learned weight of the feature function (f) $\lambda_1, \ldots, \lambda_k$, f_k is a set of feature functions $f_1, \ldots, f_k$, k is the index of the feature functions, and K is the weight index. By summing the feature function f_k with all observations (x), it will become a global feature.

Inference / Decoding finds the best label for each input sequence using maximum likelihood:

$$\arg max_{y \in Y^m} P(Y|X; w) \quad (4)$$

$P(Y|X; w)$ refers to equation (3). To make the calculation tractable, the Viterbi algorithm is normally employed. This sets up a lattice with columns for observations and rows for states. For each column o_t, there is a cell $V_t(j)$ for each state q_j at time t. The Viterbi probability computes the values of cell $V_t(j)$ by taking the most probable path extension (Jurafsky & Martin, 2019):

$$V_t(j) = max_{i=1}^{N} V_{t-1}(i) \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, x, t) \quad (5)$$

Parameter (w) estimation is used to maximize the log-likelihood for the best fit to training data, because the conditioning is inside the logarithm. The conditional log-likelihood function is

$$L(\theta) = \sum_{i=1}^{m} \log P(Y|X; w). \quad (6)$$

When the data are the pair

$$D = \{x(m), y(m)\}_{m=1}^{M}, \quad (7)$$

then the parameter estimation is

$$w^* = arg\, max_{w \epsilon \mathbb{R}^d} \sum_{i=1}^{m} log\, P(Y|X; w) - \frac{\lambda}{2} \|w\|^2, \quad (8)$$

where $\frac{\lambda}{2} \|w\|^2$ is regularization.

In the word segmentation, Y is a boundary marker, Y = {B, M, E, S}. The example sentence X is "ฉันไม่สบาย" (I have a cold). Table 3.2 shows an example of sequence labels (y_t) and input data (x_t) for syllable segmentation. The feature function (f_k) of the syllable segmentation will be described in the next section. In POS tagging, X is a word and Y is a POS tag.

Table 3.2 Example of Input and Boundary Markers in Syllable Segmentation

t	x_t	y_t
0	ฉัน	S
1	ไม่	S
2	ส	B
3	บา	M
4	ย	E

3.3 Word Segmentation Method and Technique

CRFs will be used to create MTU and syllable segmentation model. In the training process, features are an important part of specifying the identities of tokens, and the template for predicting a sequence of tokens is probability.

3.3.1 Minimum Text Unit Extraction

In this step, MTUs are extracted from the input text. The boundaries of MTUs are determined by a CRF model. The features used in this step include two consonant types: consonant (C) and non-suffix consonant (N), a consonant that cannot be placed at the end of a word; six vowels: front vowel (F), special vowel (S), upper vowel (U), rear vowel (B), lower vowel (L) and other vowel (O); tone (T); number (D); space (G); and symbol (Q). A boundary marker is the answer tag that identifies the boundary of an MTU. There are 4 markers to indicate a border: the beginning token, labeled "B"; the middle token, labeled "M"; the ending token, labeled "E"; and the standalone token, labeled "S." The template details for this character sequence are shown in Table 3.4. Examples of character features are shown in Table 3.5.

Table 3.3 Features for Minimum Text Unit Segmentation

Feature	Description	Value
Consonant	To check, if token is single consonant.	Y, N
Non-Suffix Consonant	ค ฉ ฌ ผ ฝ ห ฮ	Y, N
Front Vowel	เ แ โ ใ ไ	Y, N
Special Vowel	ะ ็	Y, N
Upper Vowel	ิ ี ึ ื	Y, N
Rear Vowel	า ๅ ะ ำ	Y, N
Lower Vowel	ุ ู	Y, N
Other Vowel	็ ์ ๆ ๅ	Y, N
Tone	To check if token is tone.	Y, N
Number	To check if token is a number.	Y, N
Space	To check if token is an empty space.	Y, N
Symbol	To check if token is a symbol.	Y, N

Table 3.4 Feature Template for Minimum Text Unit Extraction

Type	Feature	Description
Unigram	C_n, n = -3, -2, -1, 0, 1, 2, 3	The third previous character, the second previous character, the previous character, the current character, the next character, the second next character, the third next character
Bigram	$C_{-1}C_1$	The previous character and the current character

Table 3.5 Characters and Features for Minimum Text Unit Segmentation

Character	ร	้	า	น	อ	า	ห	า	ร
Feature	C	T	B	C	C	B	N	B	C

3.3.2 Syllable Identification

Syllable identification uses MTUs extracted in the previous step as input to a CRF model. The features are proposed from characteristics of Thai characters, which consist of consonant, vowel, tone (T), space (S), number of input characters (D), and the first character (FC) and last character (LC) of input tokens.

Consonants are categorized into 6 types: single consonant (C); non-suffix consonant (N); combined consonant (CC), which is a consonant that can be combined with the previous consonant and when pronounced will still have one syllable; prefix combined consonant (PC), which is a consonant that can be combined with a combined consonant (CC); consonant as vowel (CV), which is a consonant representing a vowel; and the character "Aor" (อ). Vowels are categorized into 12 types: front vowel_1 (FV1), which is a leading vowel that cannot have another character in front of it; front vowel_2 (FV2), which is a leading vowel that can have another character in front of it; special vowel (SV), which is a upper vowel that must be followed by a consonant; vowel "Maiyamok," (ๆ); vowel "Garund," (์); vowel "Maitaikoo," (็); vowel "Paiyarn," (ๆ); upper vowel (UV); lower vowel (LV); rear vowel_1 (RV1), which is a rear vowel that cannot have any consonant after it; rear vowel_2 (RV2), which is a rear vowel that can have a consonant after it, and combined vowel (CV), which is any vowel that can combine with other vowels into one syllable. Table 3.7 shows the unigram and bigram templates used for syllable identification. Table 3.8 and 3.9 show examples of feature assignment for syllable segmentation.

Table 3.6 Features for Minimum Text Unit Segmentation

Feature	Description	Value
Consonant	To check if token is a single consonant.	Y, N
Non-Suffix Consonant	ค ฉ ฌ ผ ฝ ห ฮ	Y, N
Combined consonant	ร | ล | ว	R, L, V, N
Prefix combined consonant	ก ข ค ต ท ป พ ผ | จ ช ศ ท | ห	T, F, S, N

Feature	Description	Value
Character as vowel	ย ล ว อ	Y, N
Aor	อ	Y, N
Front vowel_1	เ ไ โ	Y, N
Front vowel_2	แ ใ	Y, N
Special vowel	ั ็	Y, N
Maiyamok	ๆ	Y, N
Garund	์	Y, N
Maitaikoo	๎	Y, N
Paiyarn	ฯ	Y, N
Upper vowel	ิ ี ึ ื	Y, N
Lower vowel	ุ ู	Y, N
Rear vowel_1	ะ ำ	Y, N
Rear vowel_2	า	Y, N
Combined vowel	เ◌ีย เ-ะ แ-ะ เ-า โ-ะ | เ◌ิ เ◌ี เ◌ื เ◌็	C, B, N
Tone	◌่ ◌้ ◌๊ ◌๋	Y, N
Space	To check if token is space.	Y, N
Number of input characters	To check if token is number.	Y, N
First character of input token	The first character of a token.	First character
Last character of input token	The last character of a token.	Last character

Table 3.7 Feature Template for Syllable Identification

Type	Feature	Description
Unigram	C_n, n = -4, -3, -2, -1, 0, 1, 2, 3, 4	The fourth previous MTU, The third previous MTU, the second previous MTU, the previous MTU, the current MTU, the next MTU, the second next MTU, the third next MTU, the fourth next MTU
Bigram	$C_{-1}C_1$	The previous MTU and current MTU

Table 3.8 Minimum Text Unit with Character Features for Syllable Segmentation

	C	N	CC	PC	VC	A	LV1	LV2	SV	UV	LV
ร้า	N	N	R	N	N	N	N	N	N	N	N
น	Y	N	N	N	N	N	N	N	N	N	N
อา	N	N	N	N	N	Y	N	N	N	N	N
หา	N	N	N	N	N	N	N	N	N	N	N
ร	Y	N	R	N	N	N	N	N	N	N	N

Table 3.9 Minimum Text Unit with Character Features for Syllable Segmentation (continued)

	RV1	RV2	M	G	M	P	CV	T	S	FC	LC	D
ร้า	N	Y	N	N	N	N	N	Y	N	ร	า	3
น	N	N	N	N	N	N	N	N	N	N	N	1
อา	N	Y	N	N	N	N	N	N	N	อ	า	2
หา	N	Y	N	N	N	N	N	N	N	ห	า	2
ร	N	N	N	N	N	N	N	N	N	N	N	1

3.3.3 Word Integration

Once syllables are identified, words are constructed by merging nearby syllables together. A combination of the Longest Matching and pattern rules is employed for this task. Pattern rules are constructed from the structure of the Thai language; these rules are shown in Table 3.10. There are 18 rules with 9 types of characters; the character types are described in Table 3.11. The rules are found to enhance accuracy and avoid ambiguities from unknown words, such as the informal text "เป็นมาส์ก" (A facial mask). Without the pattern rules, this would be segmented as "เป็นมา|ส์ก" (Occur | no meaning for "ส์ก"). The pattern rules group this input as "เป็น" (Is) "มาส์" (no meaning) "ก" (consonant "ก").

Table 3.10 Pattern Rules of Thai Word Structure

Pattern Rules
< Consonant + Tone >
< Consonant + Upper vowel >
< Consonant + Upper vowel + Tone >
< Consonant + Lower vowel >
< Consonant + Lower vowel + Tone >
< Consonant + Rear vowel >
< Consonant + Tone + Rear vowel >
< Consonant + Others vowel >
< Consonant + Special vowel 1 + Consonant >
< Consonant + Special vowel 2 >
< Special vowel 2 + Consonant + Consonant + Special vowel 2 >
< Front vowel + Consonant >
< Front vowel + Consonant + Consonant >
< Front vowel + Consonant + Tone >
< Front vowel + Consonant + Upper vowel + Consonant >
< Front vowel + Consonant + Upper vowel + Tone + Consonant >
< Front vowel + Consonant + Rear vowel >
< Front vowel + Consonant + Tone + Rear vowel >

Table 3.11 Type of Character for Pattern Rules

Feature	Description
Consonant	Single consonant
Front vowel	เ ไ โ แ ใ
Upper vowel	◌ิ ◌ี ◌ึ
Lower vowel	◌ุ ◌ู
Rear vowel	า ๅ ะ ◌ำ
Special vowel 1	◌ั ◌็
Special vowel 2	◌ๆ
Others vowel	◌์ ◌๎ ฯ
Tone	◌่ ◌้ ◌๊ ◌๋

```
charList = null
while s_i is a syllable in a sentence S
  block = s_i-5 s_i-4 ... s_i
  foreach character c in the block
    charList = charList + c
    if charList matches a rule or a word in dictionary and longer
      than current word
      word = charList
      charList = null
    end if
  end for
  i = i + 1
```

Figure 3.4 Pseudocode for Word Construction

Six syllables are processed at one time, from left to right. From a sequence of six syllables, one character at a time is considered by checking against the rules, comparing against a self-gathered dictionary that includes official dictionaries, abbreviations, and slang words, implemented as a trie, and matching it with the longest word in the dictionary. For example, in the character sequence "ธรรมชาติ" (Nature), the first entry is "ธรร", which has no exact match in the dictionary. The next entry is the character " ม " Combined with the previous entry, this becomes "ธรรม" (Dharma), which can be found in dictionary. However, when the word "ธรรม" is combined with the next two entries, "ชา" and "ติ," it becomes a new word, "ธรรมชาติ". The previous word is then discarded and replaced by the longer one. The process is repeated until the last entry. When the longest sequence of characters is selected as a word, the remaining characters will become the first syllable of the next sequence to be processed. The process continues until it reaches the end of the text.

CHAPTER 4

PART-OF-SPEECH TAGGING METHODOLOGY

This research proposes a technique for assigning Part-of-Speech (POS) tags to Thai words. To develop POS tagging, this study uses word and POS tags from the famous Thai POS tag corpus ORCHID to design a new set of POS tags (section 4.1). The modified corpus is a new POS tag corpus, in which word units and POS tags are modified from the original. The modified corpus is used to train the Conditional Random Field (CRF) model, and performance is tested against three test datasets. The training data and test data are briefly explained in section 4.2. POS tagging using CRF and the features of CRF are described in section 4.3.

4.1 Design of Part-of-Speech Tag

ORCHID is a Thai part-of-speech tag corpus developed by the Communications Research Laboratory (CRL) and the National Electronics and Computer Technology Center (NECTEC). POS tags in ORCHID can be divided into 14 categories and 47 subcategories. It proposed a tagset based on the roles of words by observing a large amount of text.

The main reason to revise the ORCHID corpus is to reduce its complexity for usage. As ORCHID tried to clarify the ambiguous part of word structure, it increased the number of tags. As a result, it also required the user to be highly knowledgeable about linguistics to understand the meanings of all the POS tags. To reduce this complexity, this research creates a new list of POS tags. The tagset of ORCHID is revised from the original POS tags shown in Table 4.1. The new POS tags consist of 29 tags, as shown in Table 4.2, with the equivalent reference tags from ORCHID in the last column.

Table 4.1 Part-of-Speech Tags in ORCHID

ORCHID	Description
NPRP	Proper noun
NCNM	Cardinal number
NONM	Ordinal number
NLBL	Label noun
NCMN	Common noun
NTTL	Title noun
PPRS	Personal pronoun
PDMN	Demonstrative pronoun
PNTR	Interrogative pronoun
PREL	Relative pronoun
VACT	Active verb
VSTA	Stative verb
VATT	Attributive verb
XVBM	Pre-verb auxiliary, before negator ไม่
XVAM	Pre-verb auxiliary, after negator ไม่
XVMM	Pre-verb auxiliary, before or after negator ไม่
XVBB	Pre-verb auxiliary, in imperative mood
XVAE	Post-verb auxiliary
DDAN	Definite determiner, after noun without classifier in between
DDAC	Definite determiner, allowing classifier in between
DDBQ	Definite determiner, between noun and classifier or preceding quantitative expression
DDAQ	Definite determiner, following quantitative expression
DIAC	Indefinite determiner, following noun; allowing classifier in between
DIBQ	Indefinite determiner, between noun and classifier or preceding quantitative expression

ORCHID	Description
DIAQ	Indefinite determiner, following quantitative expression
DCNM	Determiner, cardinal number expression
DONM	Determiner, ordinal number in between
ADVN	Adverb with normal form
ADVI	Adverb with iterative form
ADVP	Adverb with prefixed form
ADVS	Sentential adverb
CNIT	Unit classifier
CLTV	Collective classifier
CMTR	Measurement classifier
CFQC	Frequency classifier
CVBL	Verbal classifier
JCRG	Coordinating conjunction
JCMP	Comparative conjunction
JSBR	Subordinating conjunction
RPRE	Preposition
INT	Interjection
FIXN	Nominal prefix
FIXV	Adverbial prefix
EAFF	Ending for affirmative sentence
EITT	Ending for interrogative sentence
NEG	Negator
PUNC	Punctuation

Table 4.2 Selected Part-of-Speech Tag

Selected-POS Tag	Description	Reference
CC	Coordinate conjunction	JCRG, JCMP
CD	Cardinal number	NCNM, DCNM
DT	Determiner	DETERMINER (exclude DCNM)
IN	Preposition / Subordinating conjunction	RPRE, JSBR
LS	List item marker	NONM, NLBL
NN	Common noun	NCMN
NNP	Proper noun	NPRP
NTTL	Title noun	NTTL
PRP	Personal pronoun	PPRS
PDMN	Demonstrative pronoun	PDMN
PNTR	Interrogative pronoun	PNTR
PREL	Relative pronoun	PREL
RB	Adverb	ADVN, ADVS
ADVI	Adverb with iterative form	ADVI
ADVP	Adverb with prefixed form	ADVP
SYM	Symbol	PUNC
INT	Interjection	INT
VB	Verb	VACT, VSTA, VATT
NEG	Negator	NEG
EAFF	Ending for affirmative sentence	EAFF
EITT	Ending for interrogative sentence	EITT
FIXN	Nominal prefix	FIXN

Selected-POS Tag	Description	Reference
FIXV	Adverbial prefix	FIXV
CLS	Classifier	CNIT, CLTV, CMTR, CFQC, CVBL
XVBM	Pre-verb auxiliary, before negator "ไม่"	XVBM
XVAM	Pre-verb auxiliary, after negator "ไม่"	XVAM
XVMM	Pre-verb auxiliary, before or after negator "ไม่"	XVMM
XVBB	Pre-verb auxiliary, in imperative mood	XVBB
XVAE	Post-verb auxiliary	XVAE

4.1.1 Coordinate Conjunction

Two original conjunction tags, "JCRG" and "JCMP," are combined into the new tag "CC." This refers to a coordinating conjunction.

4.1.2 Cardinal Number

"DCNM," a determiner for a cardinal number expression, and "NCNM," a cardinal number itself, are combined into the new tag "CD." This refers to a cardinal number. "DCNM" is used when a cardinal number serves as a determiner before a noun, such as "หนึ่งอัน" (One piece) or "2 กล่อง" (2 boxes), while "NCNM" is a cardinal number used on its own, such as 1, 2, 3." It easily for the user to become confused about whether something should be "DCNM" or "NCNM."

4.1.3 Preposition / Subordinating Conjunction

Another original conjunction tag, "JSBR," is combined with "RPRE" to create the "IN" tag. This refers to a preposition or subordinating conjunction. As with the well-known Penn Treebank corpus, the system combines prepositions and subordinating conjunctions into one tag. The Penn Treebank POS tags are shown in Table 4.3.

4.1.4 Determiner

All original determiners (excepted “DCNM”) are grouped into the “DT” tag to refer to words that function as determiners.

4.1.5 List Item Marker

In the original corpus, “NOMN” is an ordinal number. This defines something’s position in a series, such as “ที่1” (First) or “ที่สอง” (Second). “NLBL” is a label noun, defining the series of thing’s label such as 1, 2, or a, b. The Penn Treebank has POS tags to define the markers of list items. “NOMN” and “NLBL” are thus grouped into a new tag “LS” (the same name as used in the Penn Treebank).

4.1.6 Adverb

“ADVN” and “ADVS” are the original tags for adverbs with normal form and sentential adverbs, respectively. Both are combined into one adverb tag, “RB.” The other adverb tags, for adverbs with iterative form and adverbs with prefixed form, are easily understood and thus kept separate from “RB.”

4.1.7 Verb

The subcategories of verb tags, “VACT,” “VSTA,” and “VATT,” are all grouped into the “VB” tag.

4.1.8 Classifier

The classifier tags are grouped together. The new tag for classifiers is “CLS.”

Table 4.3 Penn Treebank Part-of-Speech Tag

Penn Treebank	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential “there”

Penn Treebank	Description
FW	Foreign word
IN	Preposition / Subordinating conjunction
JJ	Adjective
JJR	Comparative adjective
JJS	Superlative adjective
LS	List item maker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Comparative adverb
RBS	Superlative adverb
RP	Particle
SYM	Symbol
TO	“to”
UII	Interjection
VB	Verb base form
VBD	Verb past tense
VBG	Verb gerund or present participle

Penn Treebank	Description
VBN	Verb past participle
VBP	Verb non-3rd person singular present
VBZ	Verb 3rd person singular present
WDT	WH-determiner
WP	WH-pronoun
WP$	Possessive WH-pronoun
WRB	WH-adverb
$	Dollar sign
#	Pound sign
“	Left quote
”	Right quote
(	Left parenthesis
)	Right parenthesis
,	Comma
.	Sent-end punctuation
:	Sent-mid punctuation

4.2 Corpus

4.2.1 Modified Corpus

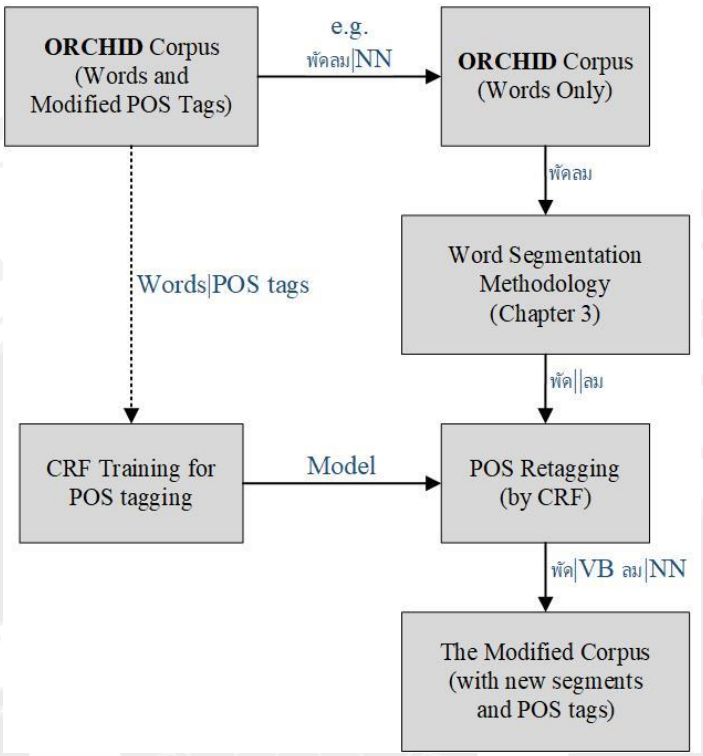

Figure 4.1 Process of the Modified Corpus

This research on POS tagging used words from ORCHID with the modified POS tags, as described in the previous section. In the first stage, the original POS tags are renamed into the proposed tags. The words with modified POS tag in ORCHID will be used as training data to generate POS tags for new word units. The new word units are also created with words from ORCHID segmented using the methodology described in Chapter 3. ORCHID is re-segmented from 296,784 words to 430,860 words. After we have training data and the set of re-segmented words, this training data will be used to train CRF (the features will be described in section 4.3). Then, the CRF model will be tested with the re-segmented words to predict POS tags for each word. The results of this process, the re-segmented word units with modified POS tags, called the modified corpus, will serve as training data for POS tagging.

4.2.2 Word Unit Assignment

The word units from ORCHID and the re-segmented words are quite different, as shown in the following tables. In Table 4.4, the number of word units in the original corpus and the re-segmented one are equal; however, the word units at positions 5 and 6 are different. In the original, the word "ครั้ง" (Time) is tagged as "CFQC" (Frequency classifier) and "ที่ 1" (First) as "DONM" (determiner, ordinal number expression), while in the new segmentation, the word "ครั้งที่" (Time) is tagged as "CFQC" and "1" as "NLBL" (Label noun). In Table 4.5, on the other hand, the word segmentation is very different. The original from ORCHID has only 1 unit of word after re-segmentation but has now been separated into 6 units. The original word, "กระทรวงวิทยาศาสตร์ เทคโนโลยีและการพลังงาน" (Ministry of Science, Technology, and Energy) was tagged as "NRPR" (Proper noun), while the re-segmented word was tagged with different POS.

Table 4.4 Example of Word Segmentation with Equal Units

WORD	1	2	3	4	5	6
Original	การ	ประชุม	ทาง	วิชาการ	ครั้ง	ที่ 1
Re-segmented Word	การ	ประชุม	ทาง	วิชาการ	ครั้งที่	1

Table 4.5 Example of Word Segmentation with Unequal Units

WORD	1	2	3	4	5	6
Original	กระทรวงวิทยาศาสตร์ เทคโนโลยี และการพลังงาน					
Re-segmented Word	กระทรวง	วิทยาศาสตร์	เทคโนโลยี	และ	การ	พลังงาน

4.3 Part-of-Speech Tagging Method and Technique

4.3.1 Features of Part-of-Speech Tagging

The features used in this step include the word in each entry (WD) and the number of characters (NC). In addition, the following features are labelled “Y” for “yes” or “N” for “no”:

1) Word is only a number (N)
2) Word begins with a number (BN)
3) Word ends with a number (EN)
4) Word is a symbol (S)
5) Word begins with a symbol (BS)
6) Word ends with a symbol (ES)
7) Word is in a foreign language (F)
8) Word contains the character “ ์ ” (G)
9) Word contains the character sequence “การ” or “ความ” (CP)
10) Word begins with a prefix character sequence (PF)
11) Word contains the character sequence “ไม่” (NG)
12) Word begins with the negator character (BG) and
13) Word contains one Thai consonant (TH).

Examples of features for the phrase “ความ|หนา|ของ|เบส|ไม่|เกิน|0” (The base thickness not over 0) are shown in Table 4.6. The template details for this character sequence are shown in Table 4.7.

Table 4.6 Examples of Words with Character Features for Part-of-Speech Tagging

	WD	ND	N	BN	EN	S	BS	ES	F	G	CP	PF	NG	BG	TH
ความ	ความ	4	N	N	N	N	N	N	N	N	Y	Y	N	N	Y
หนา	หนา	3	N	N	N	N	N	N	N	N	N	N	N	N	Y
ของ	ของ	3	N	N	N	N	N	N	N	N	N	N	N	N	Y
เบส	เบส	3	N	N	N	N	N	N	N	N	N	N	N	N	Y
ไม่	ไม่	3	N	N	N	N	N	N	N	N	N	N	Y	Y	Y
เกิน	เกิน	4	N	N	N	N	N	N	N	N	N	N	N	N	Y
0	0	1	Y	N	N	N	N	N	N	N	N	N	N	N	N

Table 4.7 Feature Template for Part-of-Speech Tag

Type	Feature	Description
Unigram	C_n, n = -2, -1, 0, 1, 2, -2 -1 0	The second previous character, the previous character, the current character, the next character, the second next character, the second previous character, the previous character and the current character
Bigram	$C_{-1}C_1$	The previous character and the current character

4.3.2 Part-of-Speech Tagging

This process is mainly separated into two experiments, depending on the word unit in the test data. First, there are two sets of test data with the same word units as training data, namely test data A and test data B (see Figure 4.2 and 4.3). Second, there is test data C, with word units that differ from the training data (see Figure 4.4). Although test data B and C contain the same data content, test data B is taken from the modified corpus (words have already been re-segmented into new units) while test data C is from the original ORCHID corpus.

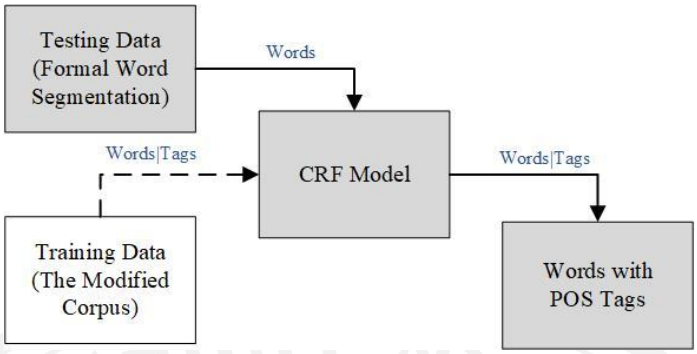

Figure 4.2 Process of Part-of-Speech Tagging for Formal Word Segmentation (Test Data A)

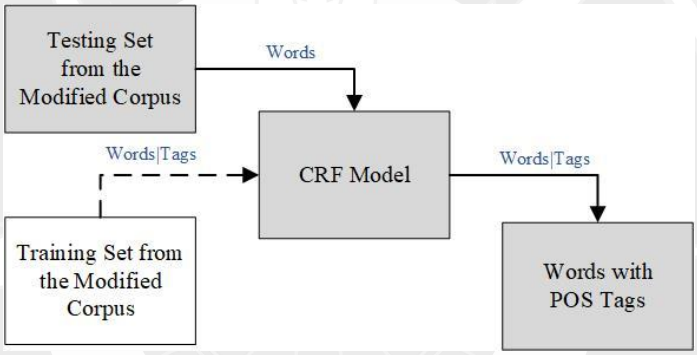

Figure 4.3 Process of Part-of-Speech Tagging for Test Set from the Modified Corpus (Test Data B)

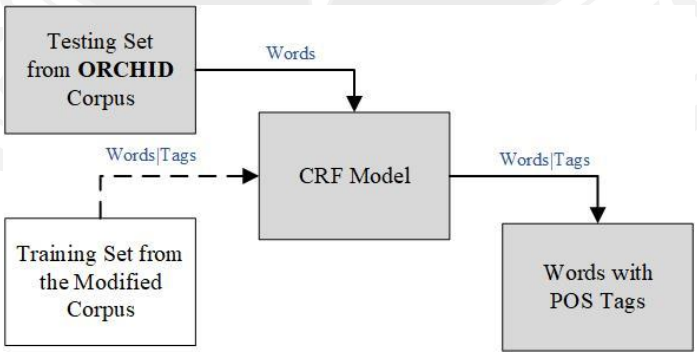

Figure 4.4 Process of Part-of-Speech Tagging for Test Set from the ORCHID Corpus (Test Data C)

CHAPTER 5

EVALUATION

5.1 Performance Measurement

To evaluate performance, precision, recall, and F-measure are used. Precision and recall are the most common measurements for Machine Learning, while F-measure is the harmonic average of precision and recall.

$$Precision = \frac{number\ of\ correct\ tokens}{number\ of\ tokens\ in\ system\ output} \quad (9)$$

Source: Adapted from Kudo et al. (2004).

$$Recall = \frac{number\ of\ correct\ tokens}{number\ of\ tokens\ in\ test\ corpus} \quad (10)$$

Source: Adapted from Kudo et al. (2004).

$$F_{\beta=1} = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (11)$$

Source: Adapted from Kudo et al. (2004).

5.2 Experiment on Word Segmentation

5.2.1 Dataset

The proposed method is evaluated using actual data collected from the web and social networks. The training data is part of the BEST2010 corpus, a Thai corpus

published by the National Electronics and Computer Technology Center (NECTEC). It consists of 132,836 characters of training data for Minimum Text Unit (MTU) segmentation and 35,674 MTUs of training data for syllable segmentation. The test data are collected from several sources to represent both formal and informal texts. Formal texts are collected from the news. Informal texts are collected from social media and include unknown words, jargon, slang and informal ways of expressing opinions. The two test datasets comprise 10,023 and 13,464 characters, respectively, for MTU segmentation and 5,289 and 6,603 MTUs, respectively, for syllable segmentation.

Table 5.1 The Number of Testing Data

Text Number	Minimum Text Unit Segmentation		Syllable Segmentation		Word Segmentation	
	Formal Text	Informal Text	Formal Text	Informal Text	Formal Text	Informal Text
1	2,293	4,108	1,136	1,950	703	1,263
2	2,028	2,495	1,018	1,266	577	750
3	2,651	3,416	1,596	1,720	819	1,055
4	1,505	1,607	757	785	443	509
5	1,631	1,838	782	882	478	574
Total	**10,023**	**13,464**	**5,289**	**6,603**	**3,020**	**4,151**

The baseline model to compare with is a famous system available on the web which uses a convolutional neural network (CNN) to segment words. It was trained on 90% of the BEST corpus from NECTEC.

5.2.2 Result

Given the word segmentation result, we manually check and determine the error, if it matches the dictionary, and if it is reasonable in context. Figure 5.1 shows examples of segmentation for formal and informal texts. There are some incomplete words which should have been grouped together. If these words are not grouped, however, the meaning does not change and the context remains correct. Clearly, there are segmentation errors; “เที่ยวโอซาก้า” (Travel to Osaka) and “ช็อปกระจาย” (Shopping

spree) should be segmented as “เที่ยว|โอซาก้า” (travel to|Osaka) and “ช็อป|กระจาย” (shopping|spree), since the meaning of each word is totally different. In addition, the words “โอซาก้า” and “ช็อป” are informal, being adapted from foreign languages and including the word “ค่าาา” (The suffix word usually use to make sentence more polite), which is a teenage slang word.

Formal Text		
Input Text	เรารู้สึกสบายใจปลอดภัยหรืออยู่ในที่ทางของตน ไม่ใช่เรื่องผิดเมื่อเรารู้สึกพอใจกับสิ่งนั้นโดยไม่คิดถึงการเปลี่ยนแปลงในแง่บวกที่อาจจะเกิดขึ้นได้	If we are feeling comfortable, safe, or like we are living in our own way, that is not wrong, if we are satisfied with things and do not care about the possibility of positive change.
Word Segmentation	เรา|รู้สึก|สบายใจ|ปลอด|ภัย|หรือ|อยู่|ใน|ที่|ทาง|ของ|ตน|ไม่|ใช่|เรื่อง|ผิด|เมื่อ|เรา|รู้สึก|พอใจ|กับ|สิ่ง|นั้น|โดย|ไม่|คิด|ถึง|การ|เปลี่ยนแปลง|ใน|แง่|บวก|ที่|อาจ|จะ|เกิด|ขึ้น|ได้	we | feeling | comfortable | clear | danger | or | live | in | place | way | of | ours | not | yes | subject | wrong | when | we | feeling | satisfy | with | things | that | by | not | think | to | [prefix of next word] | change | in | point of view | positive | that | may | will | happen | come out | can
Incomplete Word Segmentation	ปลอด|ภัย, ไม่|ใช่, คิด|ถึง, การ|เปลี่ยนแปลง, อาจ|จะ, เกิด|ขึ้น	clear | danger, not | yes, think | to, [prefix of next word] | change, may | will, happen | come out
Segmentation Error	none	
Informal Text		
Input Text	วันนี้จะพาไปเที่ยวโอซาก้าลุยๆกินกระจุยช็อปกระจายไปค่าาา	Today, we will take you on an Osaka trip. Let’s go on an eating and shopping spree.
Word Segmentation	วัน|นี้|จะ|พา|ไป|เที่ยว|โอซาก้า|ลุย|ๆ|กิน|กระจุย|ช็อป|กระจาย|ไป|ค่าาา	day | this | will | take | go | travel to Osaka | go | [repeat on the previous word] spread widely | shopping spree | go | yeah
Incomplete Word Segmentation	วัน|นี้, ลุย|ๆ	day | this, go | [repeat on the previous word]
Segmentation Error	เที่ยวโอซาก้า, ช็อปกระจาย	travel to Osaka, shopping spree

Figure 5.1 Examples of Segmentation of Formal Text and Informal Text

The results of word segmentation are shown in Table 5.2 and Table 5.3. For formal texts, the proposed method yields F-scores between 0.9784 and 0.9965, while the baseline method gives F-scores between 0.9428 and 0.9925. For informal texts, the proposed method yields F-scores between 0.9797 and 0.9857 and the baseline model gives F-scores between 0.9196 and 0.9414. The results are summarized in Figure 5.2 and Figure 5.3.

Table 5.2 Word Segmentation Precision, Recall, and F-score for Formal Texts

Formal	Algorithm	P (%)	R (%)	F-score (%)
1	Baseline	0.9933	0.9917	0.9925
	Proposed Method	0.9944	0.9981	0.9963
2	Baseline	0.9877	0.9938	0.9907
	Proposed Method	0.9954	0.9977	0.9965
3	Baseline	0.9343	0.9514	0.9428
	Proposed Method	0.9877	0.9922	0.9899
4	Baseline	0.9639	0.9639	0.9639
	Proposed Method	0.9936	0.9968	0.9952
5	Baseline	0.9717	0.9818	0.9767
	Proposed Method	0.9731	0.9837	0.9784

Table 5.3 Word Segmentation Precision, Recall, and F-score for Informal Texts

Informal	Algorithm	P (%)	R (%)	F-score (%)
1	Baseline	0.9275	0.9558	0.9414
	Proposed Method	0.9834	0.9880	0.9857
2	Baseline	0.9159	0.9459	0.9306
	Proposed Method	0.9857	0.9810	0.9833
3	Baseline	0.9230	0.9411	0.9320
	Proposed Method	0.9815	0.9847	0.9831
4	Baseline	0.9005	0.9396	0.9196
	Proposed Method	0.9819	0.9775	0.9797
5	Baseline	0.9149	0.9526	0.9333
	Proposed Method	0.9803	0.9881	0.9842

We can see that the proposed method outperforms the baseline model in both environments. In precision, the results for formal texts show that the baseline method scored 0.9933, slightly lower than the proposed method's 0.9954. The results for

informal texts show that the highest score for the proposed method is 0.9857, while the highest score for the baseline is only 0.9275. Similarly, the recall for formal texts shows that the baseline method scores lower than the proposed method, while the results for informal texts show that the highest score of the proposed method is 0.9881 while the highest score of the baseline is only 0.9558.

The baseline model often merges two or more words into a single word. For example, in "สั่ง|กาแฟ" (Order coffee), the baseline model combines two words into one word, "สั่งกาแฟ." This segmentation is incorrect because "สั่ง" (Order) and "กาแฟ" (Coffee) cannot be combined into a compound word. Not only are these two words are combined, but there is also the case of "กินคีโตอนุโลม" (eating keto allow), which should be separated into "กิน" (Eat), "คีโต" (Keto), and "อนุโลม" (Allow). Errors with the proposed method mostly occur as with "รถ (Car)|ติดใจ (impress)|กลาง (center)|แมนฮัตตัน (Manhattan)," where every word has its own meaning. This is incorrect because the context of the sentence is incompatible with the segmented words. The given sentence should be segmented as "รถ (Car)|ติด (jam)|ใจกลาง (center)|แมนฮัตตัน (Manhattan)."

Clearly, the proposed technique outperforms the baseline model—especially in informal texts, since the baseline model was constructed from a formal corpus. Therefore, the proposed technique is more applicable to segmenting words in the Thai language in both formal and informal environments.

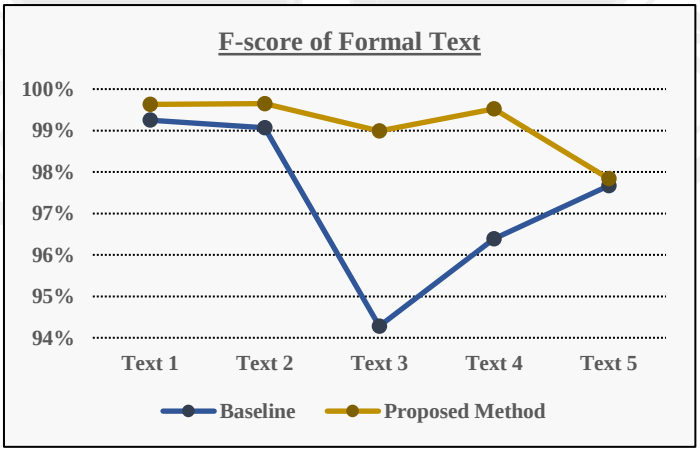

Figure 5.2 The Results for Formal Texts

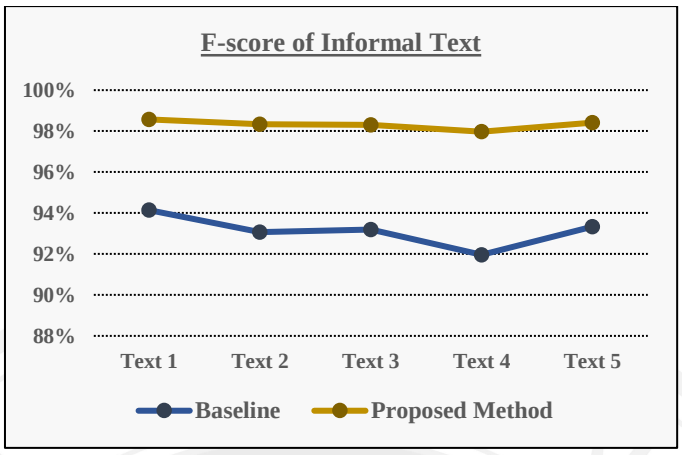

Figure 5.3 The Results for Informal Texts

5.3 Experiment on Part-of-Speech Tagging

5.3.1 Dataset

The research performed tests on three sets of test data to gauge the performance of the proposed tagging technique and the modified corpus. The first set of test data, test data A, consists of five files of formal text, the results of the word segmentation from Chapter 3. Test data A are separated into test data A-1 to A-5, which contain 531, 432, 645, 311, and 392 words, respectively. Test data B is from ORCHID. It is re-segmented into 1003 words using CRF and the longest match technique, as described in Chapter 3. Test data C is taken from ORCHID without word modification. It contains 752 words. This dataset is separated into training and testing sets without overlap.

Table 5.4 Number of Words in Each Dataset Used for Testing

Test Data	A-1	A-2	A-3	A-4	A-5	B	C
Number of Words	531	432	645	311	392	1003	752

5.3.2 Result

The proposed method is evaluated using data collected from the news and the selected corpus. The training data was (1) full data from the modified corpus, used for test data A, (2) part of the modified corpus, which was split into training data and test dataset then used for test data B and test data C.

ORCHID consists of 296,784 words with 47 POS tags. It was modified into a new corpus with new word units and new, modified POS tags. The modified corpus consists of 430,860 words with 29 POS tags. Test data A consists of 5 sets of test data, test data A-1 to test data A-5, which were collected from the news. It was segmented into words using the methodology in Chapter 3. Test data B is the rest of the data that was split from the training data. It has been segmented using the same method as test data A. The last test dataset, test data C, has the same data as B but was not segmented using the proposed word segmentation method. Instead, it consists of word units from the ORCHID corpus.

The results are shown in Table 5.5. For test data A, trained with the full data, the proposed method yields F-scores between 0.8689 and 0.9437. For test data B, trained with part of the modified corpus, the proposed method yields an F-score of 0.9738. Test data C, also trained with part of the modified corpus, obtained the lowest F-score, 0.8572.

Table 5.5 Precision, Recall, and F-score for Part-of-Speech Tagging Using Conditional Random Field

Test data	Precision (%)	Recall (%)	F-score (%)
A-1	0.9822	0.8971	0.9377
A-2	0.9605	0.8244	0.8873
A-3	0.9028	0.8374	0.8689
A-4	0.9897	0.8726	0.9274
A-5	0.9729	0.9162	0.9437
B	0.9780	0.9697	0.9738
C	0.8073	0.9138	0.8572

In the results, the POS tags most commonly confused with one another were NN (Common Noun) and VB (Verb), tag as shown in Table 5.6. There was also incorrect tagging between the PRP (Personal Pronoun) tag and the NN tag. This research found that this was caused by ambiguity in the meaning of a word; for example, consider the word "มัน": if it means "it," then the POS tag should be PRP; on the other hand, if it means "potato," then it should be tagged as NN.

Table 5.6 The Percentage of Incorrect Tags

Part-of-Speech Tag	**A-1**	**A-2**	**A-3**	**A-4**	**A-5**	**B**
NN-VB	28.89	69.77	55.17	68.42	57.69	40.00
NN-PRP	31.11	11.63	12.07	0	0	0
Others	40.00	18.60	32.76	31.58	42.31	60.00

In addition, the proposed method is compared with a Thai POS tagger that uses the perceptron technique, called PythaiNLP. Both techniques use ORCHID as training data. Test data A is used to compare performance. The results of PythaiNLP are shown in Table 5.7; it gives F-scores between 0.7313 and 0.8799. The results of the proposed method and PythaiNLP are compared in Figure 5.3. Overall, the proposed method, trained with a modified ORCHID corpus using CRF, performs best.

Table 5.7 Precision, Recall, and F-score for Part-of-Speech Tagging Using Perceptron

Test data	**Precision (%)**	**Recall (%)**	**F-score (%)**
A-1	0.8286	0.8406	0.8346
A-2	0.8217	0.8090	0.8153
A-3	0.7306	0.7320	0.7313
A-4	0.8760	0.8837	0.8799
A-5	0.9146	0.8444	0.8781

The test dataset contains 2,309 words, with nouns forming the greatest number of tags at 24.73%, followed by verbs and prepositions at 23.08% and 7.32%, respectively. Cases of POS tagging error using the perceptron technique mostly occur in noun tags and verb tags, the same as the error in the proposed method. Note, however, that the word units in the training data and the test data used by the proposed method are similar, as the words have been re-segmented (see 4.2.1 and 4.2.3), and this can influence performance measures.

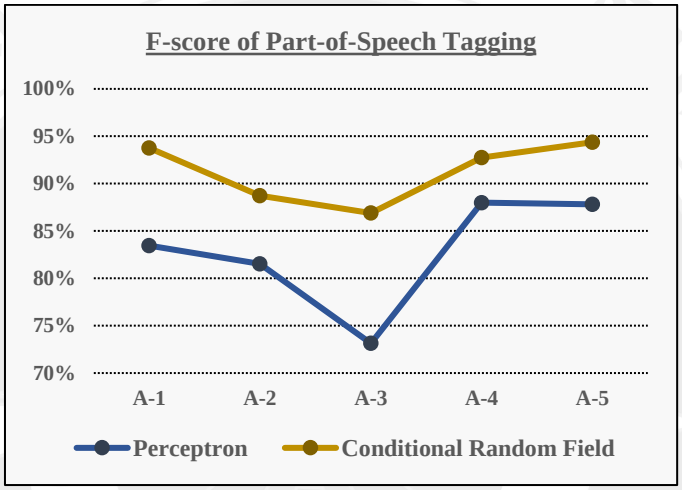

Figure 5.4 Comparison of Part-of-Speech Tagging Results

CHAPTER 6

CONCLUSION

In this thesis, we presented a novel technique for Thai word segmentation that is effective in handling formal words found in dictionaries and formal writing, as well as informal words used on social media. Minimum Text Unit (MTU) extraction and syllable identification are proposed to reduce the main problems of Thai word segmentation, which are word boundary ambiguities and unknown words. MTUs, the smallest units constituting Thai words, are extracted. Characteristics of MTUs are then used to identify syllables. Both these tasks are accomplished using a Conditional Random Field (CRF) with selected character features based on the Thai language system.

Finally, syllables are merged into words using rule-based Longest Matching. In Longest Matching, input syllables for each entry are separated and checked against the rules of Thai unit patterns. The main objective of the pattern rules is to avoid merging errors from the Longest Matching process. The proposed technique is evaluated on both formal and informal datasets against a method based on a Convolutional Neural Network (CNN) that currently gives the best performance for Thai word segmentation. The results show that the proposed method outperforms the compared system, with F-scores of 0.9965 and 0.9857 for formal and informal texts, respectively.

Part-of-Speech (POS) tags indicate the class of each word by grammatic function within a sentence. In Natural Language Processing (NLP), POS tags are useful for subsequent text processing tasks, such as parsing, summarization, speech/text recognition, etc. This research proposes a POS tagging method using CRF for the Thai language and reports its performance in terms of precision, recall, and F-score. This research also reduces the usage confusion of POS tags, since the original version of the ORCHID POS tags needed a high level of knowledge of Thai

linguistics. The features are carefully selected based on the significant properties of each word to improve the results of the model.

Three types of test data are used to evaluate the model. The word units in Test data A and B are similar, with training data re-segmented using the proposed word segmentation method describe above. Test data A is the data from the word segmentation process. Test data B and test data C are split off from the ORCHID corpus. The word units in test data B are also re-segmented using the proposed word segmentation technique. To measure the performance of the proposed method when used with different word units, test data C uses the original word units from ORCHID corpus. The results show that the proposed method achieves the highest F-score on test data B at 0.9738, while the average F-score for test data A was 0.9132 and that for test data C was 0.8572. Furthermore, the baseline using the perceptron technique, which is a module in PythaiNLP, is compared with the proposed method. To avoid duplicated data between the training and test data, test data A is used to measure performance. The results show that the proposed method performs slightly better than the baseline method. The baseline method gives F-scores of 0.8799 at the highest point and 0.7313 at the lowest.

BIBLIOGRAPHY

Aroonmanakun, W. (2002). Collocation and Thai word segmentation. *Proceedings Of SNLP-Oriental COCOSDA*, 68-75.

Boonkwan, P., Supnithi, T., Pailai, J., & Kongkachandra, R. (2013). *Gradient-descent error correction of POS tagging.* Paper presented at the Proceedings of SNLP, Phuket, Thailand.

Darwish, K., Mubarak, H., Abdelali, A., Eldesouki, M., Samih, Y., Alharbi, R., . . . Kallmeyer, L. (2018). *Multi-dialect arabic pos tagging: A crf approach.* Paper presented at the Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC), Edinburgh.

Emerson, T. (2005). *The second international Chinese word segmentation bakeoff.* Paper presented at the Proceedings of the fourth SIGHAN workshop on Chinese language Processing, Jeju Island, Korea.

Haruechaiyasak, C., & Kongyoung, S. (2009, October). *TLex: Thai lexeme analyser based on the conditional random fields.* Paper presented at the Proceedings of 8th International Symposium on Natural Language Processing.

Haruechaiyasak, C., Kongyoung, S., & Dailey, M. (2008). *A comparative study on Thai word segmentation approaches.* Paper presented at the 2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Thammasat University, Pathumthani, Thailand.

Jucksriporn, C., & Sornil, O. (2011, February). *A minimum cluster-based trigram statistical model for Thai syllabification.* Paper presented at the International Conference on Intelligent Text Processing and Computational Linguistics, (pp. 493-505). Springer, Berlin, Heidelberg.

Jurafsky, D., & Martin, J. H. (2019). Speech and Language Processing (3rd draft ed.). In: Stanford: Stanford University Press.

Kittinaradorn, R., Chaovavanich, K., Achakulvisut, T., & Kaewkasi, C. (2019). Deepcut: A Thai word tokenization library using Deep Neural Network. Retrieved from https://github.com/rkcosmos/deepcut

Kudo, T., Yamamoto, K., & Matsumoto, Y. (2004, July). *Applying conditional random fields to Japanese morphological analysis.* Paper presented at the Proceedings of the 2004 conference on empirical methods in natural language processing, Barcelona, Spain.

Limcharoen, P., Nattee, C., & Theeramunkong, T. (2009). *Thai word segmentation based-on glr parsing technique and word n-gram model.* Paper presented at the Eighth International Symposium on Natural Lanugage Processing, Thammasat University, Pathumthani, Thailand.

Liu, H., Nuo, M., Ma, L., Wu, J., & He, Y. (2011, December). *Tibetan word segmentation as syllable tagging using conditional random field.* Paper presented at the Proceedings of the 25th Pacific Asia conference on language, information and computation (pp. 168-177). Institute of Software, Chinese Academy of Sciences, Beijing.

Murata, M., Ma, Q., & Isahara, H. (2002). Comparison of three machine-learning methods for Thai part-of-speech tagging. *ACM Transactions on Asian Language Information Processing (TALIP), 1*(2), 145-158.

Na, S.-H. (2015). Conditional random fields for Korean morpheme segmentation and POS tagging. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 14*(3), 1-16.

Narupiyakul, L., Thomas, C., Cercone, N., & Sirinaovakul, B. (2004, February). *Thai syllable-based information extraction using hidden Markov models.* Paper presented at the International Conference on Intelligent Text Processing and Computational Linguistics (pp. 537-546). Springer, Berlin, Heidelberg.

National Electronics and Computer Technology Center (NECTEC). (2010). BEST: Benchmark for Enhancing the Standard of Thai language processing 2010, BEST 2010. Retrieved from http://www.hlt.nectec.or.th/best/?q=node/10.

Pailai, J., Kongkachandra, R., Supnithi, T., & Boonkwan, P. (2013, May). *A comparative study on different techniques for thai part-of-speech tagging.* Paper presented at the 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, National Electronics and Computer Technology Center (NECTEC), Bangkok,Thailand.

Phatthiyaphaibun, W., Chaovavanich, K., Polpanumas, C., Suriyawongkul, A., Lowphansirikul, L., & Chormai, P. (2016). PyThaiNLP: Thai Natural Language Processing in Python. Retrieved from http://www.lib4dev.in/info/PyThaiNLP/pythainlp/61813823. doi:10.5281/zenodo.3519354

Sornlertlamvanich, V., Charoenporn, T., & Isahara, H. (1997). ORCHID: Thai part-of-speech tagged corpus. *National Electronics and Computer Technology Center Technical Report*(12), 5-19.

Theeramunkong, T., Sornlertlamvanich, V., Tanhermhong, T., & Chinnan, W. (2000). *Character cluster based Thai information retrieval.* Paper presented at the Proceedings of the fifth international workshop on Information retrieval with Asian languages, Bangkok, Thailand.

Theeramunkong, T., & Usanavasin, S. (2001, March). *Non-dictionary-based Thai word segmentation using decision trees.* Paper presented at the Proceedings of the first international conference on Human language technology research.

Xiong, Y., Wang, Z., Jiang, D., Wang, X., Chen, Q., Xu, H., . . . Tang, B. (2019). A fine-grained Chinese word segmentation and part-of-speech tagging corpus for clinical text. *BMC medical informatics and decision making, 19*(2), 179-184.

Zhao, H., Huang, C., Li, M., & Lu, B.-L. (2006, November). *Effective tag set selection in Chinese word segmentation via conditional random field modeling.* Paper presented at the Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation.

BIOGRAPHY

Name-Surname Kannikar Paripremkul

Academic Background
Computer Science, BSc
Kasetsart University
2009, year of graduation
Information Technology Management, MSc
National Institute of Development Administration
2013, year of graduation

Experience
Year 2014 - 2015
IT Officer
Information Standard and Security Office
Metropolitan Electricity Authority
Year 2018 - Present
Information Security Consultant

Publication:
Paripremkul, K., & Sornil, O. (2021). Segmenting Words in Thai Language Using Minimum Text Units and Conditional Random Field. Journal of Advances in Information Technology Vol, 12(2).