PAPER

Multi-Stage Automatic NE and PoS Annotation Using Pattern-Based and Statistical-Based Techniques for Thai Corpus Construction*

Nattapong TONGTEP^{†a)}, *Student Member* and Thanaruk THEERAMUNKONG^{†b)}, *Member*

SUMMARY Automated or semi-automated annotation is a practical solution for large-scale corpus construction. However, the special characteristics of Thai language, such as lack of word-boundary and sentence-boundary markers, trigger several issues in automatic corpus annotation. This paper presents a multi-stage annotation framework, containing two stages of chunking and three stages of tagging. The two chunking stages are pattern matching-based named entity (NE) extraction and dictionary-based word segmentation while the three succeeding tagging stages are dictionary-, pattern- and statistical-based tagging. Applying heuristics of ambiguity priority, NE extraction is performed first on an original text using a set of patterns, in the order of pattern ambiguity. Next, the remaining text is segmented into words with a dictionary. The obtained chunks are then tagged with types of named entities or parts-of-speech (PoS) using dictionaries, patterns and statistics. Focusing on the reduction of human intervention in corpus construction, our experimental results show that the dictionary-based tagging process can assign unique tags to 64.92% of the words, with the remaining of 24.14% unknown words and 10.94% ambiguously tagged words. Later, the pattern-based tagging can reduce unknown words to only 13.34% while the statistical-based tagging can solve the ambiguously tagged words to only 3.01%.

key words: multi-stage annotation, named entity, part of speech, corpus construction, syllabic alphabetic language

1. Introduction

As fundamental tasks, word segmentation, part-of-speech (PoS) tagging, and named entity (NE) recognition are essential steps for various natural language processing applications such as text summarization, machine translation, and question answering. For languages like Burmese, Khmer, Lao, Tamil, Telugu, Balinese, and Thai, which have no explicit boundary marker between words and sentences (similar to space and a full stop in English), word segmentation is required. PoS tagging is another important task which assigns some syntactic categories such as verb, noun, and preposition to a token or a word for resolving innate ambi-

guities, while more specific predefined categories, such as person name, location, and organization are assigned in the steps of NE recognition (NER).

The current trend in PoS tagging and NE recognition is to utilize machine learning techniques, which are trainable and adjustable. Recently among Asian languages, several supervised learning techniques with acceptable performance have been proposed. For PoS tagging, Pandian and Geetha [1] utilized conditional random fields (CRFs), a probabilistic model, to segment and label sequence data, to tag and chunk PoS in Tamil. Huang et al. [2] showed that a bigram PoS tagger using latent annotations could achieve the accuracy of 94.78% when testing on a set of the Penn Chinese Treebank 6.0. For NE recognition, Lee et al. [3] presented a two-level Korean named entity classification (NEC) by cascading highly precise lexical patterns and the decision list. Park and Rim [4] classified bio-entities by using predicate-argument structures as the external context features. Tongtep and Theeramunkong [5] investigated a method to segment Thai word and recognize NE simultaneously by using the concept of character clusters together with discriminative probabilistic models.

Such machine learning tasks, however, require high-quality tagged corpora or annotated corpora for training, which are costly and time consuming to construct. There is still a limited number of studies on a method to annotate a corpus with less human effort. Lee et al. [6] proposed rules to judge the tagging reliability for constructing a Korean PoS tagged corpus. Since the quality of the PoS annotation in a corpus is crucial for the development of PoS taggers, Loftsson [7] examined three-error detection methods for automatically detecting hand-correct PoS errors in the corpus. As a study on corpus size factor, Sasano et al. [8] reported that the PoS tagging performance was not saturated even with a corpus size of 100 billion Japanese words when analyzing case frame acquisition for predicate-argument structure. In the Thai language, Isahara et al. [9] constructed a PoS tagged corpus named ORCHID manually. The ORCHID corpus was annotated on three levels: paragraph, sentence, and word. The ORCHID tag set consists of 14 categories and 47 subcategories, while they were assigned to approximately 2,560,000 words in the corpus. Charoenporn et al. [10] constructed another lexicon by using existing machine-readable dictionaries, and a sort of semantic constraint called selectional preference was added into the

Manuscript received February 1, 2013.

Manuscript revised May 16, 2013.

[†]The authors are with the School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, Thailand.

*This paper is an extension of the work called "Multi-stage Annotation using Pattern-based and Statistical-based Techniques for Automatic Thai Annotated Corpus Construction" that appeared previously in the 9th Workshop on Asian Language Resources collocated with the 5th International Joint Conference on Natural Language Processing (IJCNLP) in 2011.

a) E-mail: m5022350452@studentmail.siit.tu.ac.th

b) E-mail: thanaruk@siit.tu.ac.th

DOI: 10.1587/transinf.E96.D.2245

lexicon by analyzing Thai texts on the web. Chanlekha and Kawtrakul [11] showed a method to build a political news corpus. However the NE tag sets used in these works are still small, including mostly person names, organization names and place names. Moreover, evaluation of NE recognition in Thai was usually done in small-sized corpora. Another large corpus, namely the BEST[†] corpus, has been developed recently as for the Thai word segmentation software contest, starting from the Twelfth National Software Contest 2010 (NSC2010^{††}). It consists of seven million words in eight categories: article, encyclopedia, news, novel, Buddhism, law, talk, and Wikipedia. The BEST corpus was annotated manually based on the BEST guidelines^{†††}. As a more specific-purpose corpus, the LOTUS [12] is a corpus used for large vocabulary Thai continuous speech recognition, collected as Phone-balanced read speech of 70 hours from 50 speakers with 5000 words vocabulary size. Lately, Theeramunkong et al. [13] proposed a framework and annotation tools for tagging NE and constructing corpus in Thai. With their annotation tools, the THAI-NEST corpus was annotated and verified by collaborative experts. However, the process is very costly and time consuming. Minimizing human intervention for automatic construction of a corpus with PoS and NE is a challenging topic, especially in languages without word and sentence boundaries.

Besides the PoS and NE tagged corpus, there have been a number of works attempting to reduce human cost in the annotation. For examples, Zhang and Kordoni [14] proposed a statistical ranking model to speed-up the annotation with improved inter-annotator agreement. This model showed strong correlation to the human annotator behavior. Seraji et al. [15] developed the Persian dependency treebank using a semi-automatically syntactic dependency annotation by alternating between data-driven parsing and manual validation based on already annotated part-of-speech corpus. Kurohashi and Nagao [16] constructed the Japanese parsed corpus which annotators can correct the erroneous analyses produced by the system and also improve the parsing system.

In this paper, we propose a multi-stage automatic annotation framework to construct a PoS- and NE-tagged corpus for the Thai language. The framework is composed of two chunking stages and three tagging stages. With a list words and NEs acquired from electronic resources, a Thai running text is split into a sequence of tokens in the chunking stages. In the tagging stages, a dictionary with part-of-speech information, a set of explicit patterns and probabilistic information are applied to tag parts-of-speech to tokens. From the final tagging results, three types of tokens; token with one single tag (unambiguous token), token with the unknown tag (unknown token) and token with multiple tags (ambiguous token) are analyzed statistically. The remaining part of this

paper is organized as follows. In Sect. 2, the writing system in Thai is discussed. The overall system architecture is proposed in Sect. 3. In Sect. 4, experimental settings and results are reported. The experimental results are discussed in Sect. 5. Finally, a conclusion is illustrated in Sect. 6.

2. Thai Writing System

As an alphabetic language, graphemes (characters or letters) in Thai represent the words (phonemes). Known as an inherent-vowel alphabetic language (also called syllabic alphabet, alphasyllabary or abugida [17]), Thai consists of symbols for consonants and vowels, the same as Burmese, Khmer, Lao, Tamil, Telugu and Balinese. Unlike English (segmental alphabetic language) and Arabic or Hebrew (consonantal alphabetic language or abjad [17]), vowels and consonants in Thai are treated separately, each Thai consonant has an inherent vowel which can be changed to another vowel or muted by means of diacritics and Thai vowel marking is almost mandatory. An example of Thai texts is depicted as shown in Fig. 1.

The Thai language consists of 44 consonants, 21 vowel symbols, 4 tone markers for its 5 tonal levels, and a number of punctuation marks. The Thai writing system is left-to-right direction, without spaces between words and no uppercase and lowercase characters. Vowels can be written before, after, above, or below consonants, while all tone marks, and diacritics are written above and below the main character.

A Thai word is typically formed by the combination of one or more consonants, one vowel, one tone mark, and one or more final consonants to make one syllable. Thai verbs are not inflected for any of tense, gender, and singular or plural form. Instead, we put some additional words to express their inflection. Moreover, Thai has no distinct boundary marker between words and sentences, like a space and a full stop in English. Koanantakool et al. [18] elucidated the history of Thai language and analyzed characteristics of Thai characters involved in Thai text processing in detail.

3. The Framework

In this paper, we propose a multi-stage annotation framework to construct high-quality annotated corpora with an automatic process. The framework comprises two stages for chunking and three stages for tagging (see Fig. 2). Two stages for chunking are (1) named entity extraction and (2) word segmentation while three stages for tagging are (1) dictionary-based tagging stage, (2) pattern-based tagging stage, and (3) statistical-based tagging stage. In this work, we utilize a list of named entities and words as reusable resource for developing a tagged corpus. The output of multi-stage annotation is illustrated in Fig. 3.

In the step of NE extraction, NE tokens are extracted from the running input texts using a set of patterns, which are ordered by pattern ambiguity. The remaining parts (i.e., portions between NEs) are further processed by word seg-

[†]<http://thailang.nectec.or.th/best/>

^{††}<http://fic.nectec.or.th/nsc12>

^{†††}http://thailang.nectec.or.th/downloadcenter/index.php?option=com_docman&task=cat_view&gid=37&Itemid=61

Fig. 1 An example of Thai texts (At 21.30 on May 6, Police Lieutenant Colonel Grittinart Tulyalak, inspector (suppression) from the Bangrak Police Station was informed that there was an affray between customs officers and smugglers at the Patpong Night Market, Suriyawong District.)

Fig. 2 The framework of multi-stage annotation for automatic Thai annotated corpus construction (A: direct named entities, B: indirect named entities) Here, H-Intervention is specified when the process needs human intervention.

mentation. In practice, as dictionary-based word segmentation, a number of words can be detected using a dictionary while some portions are not detected as words since they do not exist in the dictionary. Later we will call the NE tokens and the word tokens detected in this process, as identified tokens while the portions that are not detected as NEs or words, as unidentified tokens.

Later, the identified tokens are tagged with their possible syntactic PoS or NE types while the unidentified tokens are tagged with “Unknown” tags. In the first stage of the tagging process, words are tagged with their parts-of-speech using a dictionary, later called dictionary-based tagging. In

the second stage, NEs are tagged with their potential NE types using a set of predefined patterns. While it is possible that a word or a NE may have several tags, the statistical-based tagging, the third stage, is performed to obtain the most probable tag for such word or NE.

Human interposition is involved two steps, i.e., preparing the multi-stage annotating system and verifying the corpus which is the output from the system. In the process of preparing the multi-stage annotating system, human intervention is involved to collect a list of entities from online resources, construct entity patterns, and verify indirect named entities. Later, annotators can speed-up the annotation by

Fig. 3 The output of multi-stage annotation for automatic Thai annotated corpus construction.

correcting the incorrect analyses produced by the system.

The details of these two chunking stages and three tagging stages are described in the following subsections.

3.1 Chunking

Two stages in the chunking step are (1) named entity extraction and (2) word segmentation. This section introduces the details of these two processes.

3.1.1 Named Entity Extraction

As a preparatory step, in this work, we collected 6,425 named entities from six language resources, i.e., Thai Wikipedia[†], the Royal Institute dictionary^{††}, the Government Information System^{†††}, the Company List in Thailand website^{††††}, LONGDO dictionary^{†††††}, and YAI TRON^{††††††}. In the Thai Wikipedia, some topics are useful for collecting a list of specific names. For example, a topic named ‘A list of universities’, ‘A list of countries’, ‘A list of schools’, ‘A list of important persons’, ‘A list of Thai Prime Ministers’, and etc. In such pages of the topics, we can gather a plentiful of named entities. From the Royal Institute, LONGDO, and YAI TRON dictionaries, we collect the entries which are NEs by utilizing the NE patterns to classify whether a specific entry is NE or not. A list of government organization names is collected from the Government Information System while a list of company names is acquired from the Company List in Thailand website.

[†]<http://th.wikipedia.org>

^{††}<http://www.royin.go.th>

^{†††}<http://www.oic.go.th/ginfo>

^{††††}<http://www.companyinthailand.com>

^{†††††}<http://dict.longdo.com>

^{††††††}[http://th.lug.wikia.com/wiki/YAI TRON_\(English-Thai_dictionary\)](http://th.lug.wikia.com/wiki/YAI TRON_(English-Thai_dictionary))

Besides named entity (NE) expressions, we also create a list of indicative expressions (IND) which are of limited types. This kind of indicative expression can be used as clues or markers for NE extraction. For the news domain that we are going to annotate, major dominant clues are position titles (POS) and family relationship markers (FAM). We collected 244 position titles and 952 family relationship markers. A position title (POS) expresses an organizational position of a person, such as “นาย (Mr.)”, “นางสาว (Miss)”, “พันตำรวจเอก (Police Colonel)”, “พลอากาศโท (Air Marshal)”, “นายกรัฐมนตรี (Prime minister)”, etc. Family relationship marker (FAM) indicates a link between two persons and highly occurred in the running texts. For example, “ปีเตอร์ ผู้เป็นพ่อของแนนซี่” (Peter who is Nancy’s father.) or “เซลีนเป็นพี่สาวของไดอาน่า” (Celine, Diana’s sister.). This FAM can be used as clue or marker for extracting person’s name.

As shown in Algorithm 1, the lists of NE expressions and indicative expressions are used to chunk a running text in order to obtain named entities, i.e., detection of NE boundaries. There have been four steps in detecting two types of NE; direct NEs and indirect NEs. The direct NEs are named entities that can be detected directly by using NE expressions that are collected from online resources. On the other hand, the indirect NEs are the NEs that are detected by using indicative expressions and patterns created during the direct NEs are extracted. In the first step, direct NEs are extracted using a list of NEs. In the second step, indirect NE expression patterns are constructed from the result after extracting the direct NEs with making use of indicative expressions. Each pattern includes inner clue and/or surrounding left and right contexts. Here, the inner clue is words or sequence of words which are a part of the NE and helpful to recognize similar NEs that possess the same clue. Some examples of inner clues are “สมเด็จพระ” (Her Royal Highness Princess), “บริษัท...จำกัด” (Co. Ltd.),

Algorithm 1: NE Extraction Process

-
- Input** : A text
Input : A list of named entity expressions
Input : A list of indicative expressions
Output: Named entity boundaries
- Step 1: Extraction of direct named entities**
- Use a list of named entity expressions to find direct named entities in the text, i.e., the boundaries of direct named entities are detected and then recorded as output.
- Step 2: Construction of indirect named entity extraction patterns**
- From each of the detected direct named entities,
 - identify its inner clue (denoted by *iClue*) if exists.
 - extract the 20 characters before (denoted by *cL20*) and after (denoted by *cR20*) that direct named entity and form its context pattern using a list of indicative expressions
 - construct a pattern $p = \{cL20, iClue, cR20\}$.
 - verify usability of the pattern. If positive, keep it as a valid pattern. (Human intervention)
- Step 3: Extraction of indirect named entities using patterns**
- Use the patterns acquired in Step 2 to detect indirect named entities.
- Step 4: Verification of detected indirect named entities**
- Manually check whether the extracted indirect named entities are correct or not. If they are correct, their boundaries are detected and recorded as output. (Human intervention)
-

and “นาย” (Mister). In the same way, the left and right contexts are investigated whether they are useful for detecting NEs or not. Some examples are space, tab, conjunction expressions (e.g., “ก็ต่อเมื่อ” (however), “และ” (and)), verb expressions (e.g., “กล่าวว่า” (said that), “เดินทางไป” (travel to)). The constructed patterns are verified in terms of the usability before adding into the list of patterns for further use. We observe detected NEs extracted by the constructed patterns. If the extracted NEs are valid, we keep the pattern in the list of valid patterns. In the third step, candidates of indirect NEs are detected using the constructed patterns. After detecting NEs, the remaining portions in the text are further segmented into words by the word segmentation process. However, the fourth step in Algorithm 1 is optional, the candidates are verified manually to obtain in the list of NEs for further use. In our approach, we reduce the labor-intensive tasks to only Step 2 and Step 4, where we manually check the validity of the entity extraction patterns and the validity of indirectly detected named entities.

3.1.2 Word Segmentation

While detecting words in a language with explicit word boundary (e.g., space) is straightforward, this process in

a language without explicit boundary markers (especially an inherent-vowel alphabetic language) is complicated and non-trivial. Most methods in this so-called word segmentation task are usually based on utilization of a dictionary. However, it is hard to hold all possible words in a dictionary since new words are invented or NEs are specially created. The naive method to handle words outside the dictionary is to apply longest matching from left-to-right text direction. Such dictionary-based methods with the longest matching strategy is proved to be powerful in detecting words, even they may not detect out-of-vocabulary words [19]. In this paper, we apply the longest matching technique implemented by Haruechaiyasak [20] with making use of our collected list of 155,088 words. The longest matching process is applied to the text remaining after NEs are detected in the previous step, named entity extraction. The output from NE extraction and word segmentation will be tagged with parts-of-speech and NE types in the next stage, i.e., the tagging stage.

3.2 Tagging

After extracting NEs and detecting words, the tagging process is evoked to assign a tag or a set of tags to each token (i.e., an NE or a word) in three tagging stages, dictionary-based, pattern-based and statistical-based stage, in order. Tokens which are not in the dictionary will be tagged as unknown in the dictionary-based stage. They are checked with a set of patterns in order to obtain tags in the pattern-based tagging stage. In the last stage, the statistical-based process will select the most probable tag for a token in the case that it is assigned with more than one tag from the two previous stages. In this work, twenty five (25) token types; composed of thirteen (13) parts-of-speech, four (4) NE types, two (2) indicative expressions, and six (6) special types are defined for constructing our tagged corpus. The list of these token types is enumerated in Table 1. Here, “UNK” is a special type, assigned to tokens which cannot be classified into any of 24 existing types. In the past, a number of research works applied a similar tagset, such as those in [13], [21], [22].

Figure 4 illustrates an example of the tagging process in three stages. The dictionary-based tagging assigns a single PoS tag or a set of PoS tags to a word found in the dictionary. The pattern-based tagging assigns tags to not only newly detected named entities but also unknown words if they include some clues. Finally the statistical-based tagging will use statistical information to assign the most suitable tag to a token. The details of these three stages are given in the next subsections. For clarity, the notation of $w X; \text{UNK}$ means a token w is assigned a single entity tag as unknown (UNK) and the notation of $y X; \text{CONJ}; \text{NOUN}; \text{PREP}$ means a token y is possible to have an entity tag as conjunction (CONJ), noun (NOUN), or preposition (PREP). “X;” is a separator among a token and a set of entity tags, and “;” is a separator among entity tags.

Thai Tokens	Meaning	Tagging Stage		
		Dictionary-based	Pattern-based	Statistical-based
กระทรวงพาณิชย์	Ministry of Commerce	X;LOC	X;LOC	X;LOC
กับ	and	X;CONJ;NOUN;PREP	X;CONJ;NOUN;PREP	X;CONJ;NOUN;PREP
กลุ่ม	group	X;CLAS;NOUN	X;CLAS;NOUN →	X;NOUN
ผู้ค้า	smugglers	X;UNK →	X;POS	X;POS
ตลาด	market	X;NOUN	X;NOUN	X;NOUN
พื้	Pat	X;UNK	X;UNK	X;UNK
พงษ์	Pong	X;NOUN	X;NOUN	X;NOUN
์	-	X;UNK	X;UNK	X;UNK
<SPACE>	-	X;SPC	X;SPC	X;SPC
แขวงสุริยวงศ์	Suriyawong District	X;LOC	X;LOC	X;LOC
และ	and	X;CONJ	X;CONJ	X;CONJ

Fig. 4 An example of the three tagging stages. (Ministry of Commerce and smugglers at the Patpong Night Market, Suriyawong District)

Table 1 The list of possible entity tags.

Type	Tag	Description
Word (Part-of-Speech)	ADJ	Adjective
	ADV	Adverb
	AUX	Auxiliary verb
	CLAS	Classifier
	CONJ	Conjunction
	DET	Determiner
	END	End
	INT	Interjection
	NOUN	Noun
	PREP	Preposition
	PRON	Pronoun
	QUE	Question phrase
	VERB	Verb
Named Entity	DAT	Date expression
	LOC	Location expression
	PER	Person expression
	TIM	Time expression
Indicative Expression	FAM	Family relation expression
	POS	Position title
Other	COMMENT	Comment
	ENG	English
	NUM	Number
	PUNC	Punctuation
	SPC	Space
	UNK	Unknown

3.2.1 Dictionary-Based Tagging

In the dictionary-based tagging, each token in a text is assigned with a single tag or a set of tags. As shown in Table 1, four groups of tags we use are Part-Of-Speech (PoS), Named Entity (NE), Indicative expression (IND) and other expression (OTH). For the PoS tagset, we adopt 13 PoS types from the well-known machine readable dictionary for the Thai language namely YAiTRON[†] dictionary (Yet Another LeXiTRON). It includes Thai-English and English-

Thai translations, stored in a well-formed XML format. YAiTRON is also known as a homogeneous structure dictionary, adapted from National Electronics and Computer Technology Center (NECTEC^{††})’s LEXiTRON^{†††} dictionary. In this dictionary, 13 PoSs provided to classify 32,350 unique words are adjective (ADJ), adverb (ADV), auxiliary verb (AUX), classifier (CLAS), conjunction (CONJ), determiner (DET), end (END), interjection (INT), noun (NOUN), preposition (PREP), pronoun (PRON), question phrase (QUE), and verb (VERB). Figure 5 displays an example of the word entry of the verb “ขัน” (screw, v) in the dictionary. For the NE tagset, we adopt four common named entity types; date (DAT), location (LOC), person (PER) and time (TIME). Here, location (LOC) also includes organization (ORG). We do not distinguish LOC and ORG since it is quite hard to classify them automatically. These four NE types frequently occur in news documents. For the IND tagset, we apply two types of tags which can be used as a clue to detect PER. They are family relation expression (FAM) and position title (POS). For the OTH tagset, we annotated comments (COMMENT), English expressions (ENG), numbers (NUM), punctuation (PUNC), space (SPC) and unknown (UNK). These tags are defined separately since they are quite limited in their forms and relatively easy to be detected. The UNK type is given to a part in a text which is not previously annotated by words in a dictionary and patterns.

As the result, there are three possible tagging statuses of a token w as follows. Here, let $T(w)$ be the tag set given to the token w .

An unambiguous token: A token with a single tag, excluding the “UNK” tag. That is, the number of tags is one ($|T(w)| = 1$) and the tag is not unknown (“UNK” $\notin T(w)$).

An ambiguous token: A token with multiple tags, excluding the “UNK” tag. That is, the number of tags is larger

[†][http://th.lug.wikia.com/wiki/YAiTRON_\(English-Thai_dictionary\)](http://th.lug.wikia.com/wiki/YAiTRON_(English-Thai_dictionary))

^{††}<http://www.nectec.or.th>

^{†††}<http://lexitron.nectec.or.th>

Table 2 An example of Thai grammatical patterns.

Entity	#Patterns	Pattern Examples	Entity Examples
Adverb	1	อย่าง-	อย่างรวดเร็ว (quickly)
Location	49	กระทรวง-, สถานี-, ชมรม-, ธนาคาร-, อุทยาน-,...	กระทรวงกลาโหม (Ministry of Defence), สถานีรถไฟ (Railway Station), ชมรมกีฬา (Sport Club), ธนาคารออมสิน (The Government Savings Bank), อุทยานประวัติศาสตร์ (Historical Park)
Noun	61	เครื่อง-, หนังสือ-, กระเป๋า-, กล้อง-, โครงการ-,...	เครื่องซักผ้า (Washing Machine), หนังสือแนะนำ (Guide Book), กระเป๋าเดินทาง (Travelling Bag), กล้องดูดาว (Telescope), โครงการวิจัย (Research Project)
Position	11	นัก-, ผู้-, รัฐมนตรี-,...	นักเขียน (Writer), ผู้ประพันธ์ (Author), รัฐมนตรีมหาดไทย (Minister of Interior)
Verb	3	ตัด-, ร้อง-, ไม่-	ตัดขาด (cut), ร้องขอ (request), ไม่ชอบ (dislike)

```

<entry lang="tha">
  <headword>ขัน</headword>
  <sense level="1">2</sense>
  <pos>V</pos>
  <translation lang="eng">tighten</translation>
  <translation-similar lang="eng">screw tight</translation-similar>
  <translation-similar lang="eng">wrench</translation-similar>
  <antonym>คลาย</antonym>
  <antonym>ผ่อน</antonym>
  <definition>ทำให้ตึงหรือให้แน่น</definition>
  <example>อย่าลืมขันตะปูให้แน่นหน่อยนะ</example>
  <lexitron id="10376"/>
</entry>

```

Fig. 5 An example of YaiTRON entries.

than one ($|T(w)| > 1$).

An unknown token: A token which cannot be assigned with any of 24 tags. Finally it will be tagged with “UNK” ($T(w) = \{\text{“UNK”}\}$).

3.2.2 Pattern-Based Tagging

Besides NEs and words in a dictionary, there are some patterns (e.g., prefix, suffix or both), that can make us clarify the type of NEs or words. For example, a token beginning with “Ministry of” is likely to be a location (organization), or a token starting with “Minister of” tends to be a person’s position. By investigating the dictionary, we can form a set of 125 patterns with 100% correctness. They include 1 pattern for adverb, 49 patterns for locations, 61 patterns for nouns, 11 patterns for positions and 3 patterns for verbs. An example of Thai grammatical patterns is shown in Table 2. Furthermore, other tokens such as comment, number, punctuation, space, and English characters, will be automatically assigned with an entity tag as COMMENT, NUM, PUNC, SPC and ENG, respectively. Every unknown token which does not match with these patterns in this tagging stage will be assigned with an entity tag as UNK. These patterns are evaluated through the actual annotation process and discussed in Sect. 5.

Table 3 Features for constructing a naïve Bayes classification model to select the most probable tag for a multi-tag word in the statistical-based tagging stage.

Feature	Definition
tagL2	The second entity tag on the left
tagL1	The first entity tag on the left
tagR1	The first entity tag on the right
tagR2	The second entity tag on the right
tagL2L1	Two entity tags on the left
tagR1R2	Two entity tags on the right
tagL2L1R1	Two entity tags on the left and one entity tag on the right
tagL1R1R2	One entity tag on the left and two entity tags on the right
tagL2L1R1R2	Two entity tags on the left and the right

3.2.3 Statistical-Based Tagging

After the previous stages, some tokens may have several possible tags. To make a complete corpus, we need to solve this ambiguity problem. To this end, we can apply a machine-learning approach to create a PoS classifier which is trained from the output of the previous stages. In this work, we exploit naïve Bayes classifier since it is simple and efficient to learn parameters for classification. A naïve Bayes classifier is a simple probabilistic classifier based on applying Bayes’ theorem with strong (naïve) independence assumptions. In spite of their naïve design and apparently over-simplified assumptions, naïve Bayes classifier has worked quite well in many complex real-world situations. In this paper, nine surface contextual clues are used as features for classification. They are listed in Table 3.

Predicting an entity tag t given a vector of context features $F = (f_1, f_2, \dots, f_{|F|})$. One simple way to accomplish this is to assume that once the entity tag t_j is known, all the features are independent. The result is based on a joint probability model of the form:

$$p(t_j|F) = \frac{p(t_j)p(F|t_j)}{p(F)}, \quad (1)$$

with independence assumption

$$p(t_j|F) = \frac{p(t_j) \prod_{i=1}^{|F|} p(f_i|t_j)}{p(F)}. \quad (2)$$

The best entity tag t_{best} among the output tags T is

$$t_{best} = \arg \max_{t_j \in T} p(t_j|F) \quad (3)$$

$$t_{best} = \arg \max_{t_j \in T} \frac{p(t_j) \prod_{i=1}^{|F|} p(f_i|t_j)}{p(F)} \quad (4)$$

$$t_{best} = \arg \max_{t_j \in T} p(t_j) \prod_{i=1}^{|F|} p(f_i|t_j). \quad (5)$$

We train our statistical-based entity tagger by using the output of the previous stages. It is possible to apply this naïve Bayes (NB) classifier to determine the most probable tag for a multi-tag (ambiguous) token. However, we have set a constraint that an ambiguous token will turn to be an unambiguous token (a token with only one exact tag) if the most probable tag suggested by the NB classifier is in the list of possible tags $T' = (t'_1, t'_2, \dots, t'_n)$ of the token. Otherwise, the tag of the token is not determined and still ambiguous. The formalism is as follows.

$$t_{best} = \begin{cases} t_{best} & \text{if } t_{best} \in T' \\ T' & \text{otherwise} \end{cases} \quad (6)$$

4. Experimental Settings and Results

In the Thai language, finding a corpus for annotation is quite difficult due to a limited number of online language resources. Towards this, we decided to annotate named entity and part-of speech for news documents since normally news documents are available online and their contents are various, dynamic and plentiful of newly introduced words and named entities. Moreover, a motivation of this work is an attempt to automate the construction process of a corpus named THAI-NEST, a collaboration project between

two Thai research groups in Thammasat University and NECTEC [13]. This corpus was constructed with detailed design with collaboration among a number of linguists in Thammasat University and NECTEC. It is planned to use as a standard for several contests and experiments in the future. As for our experimental data set, we collected 764 news documents, comprised of approximately 1,560,000 characters from a number of online news publishers.

As the result of NE extraction, 19,528 identified NE tokens are obtained. Their entity types and occurrences are provided in Table 4. After that, by the longest-matching word segmentation using a list of 155,088 words collected from a number of online resources, we got 316,653 identified word tokens. With these two sources, 336,181 tokens were used as the input data for the tagging process, the succeeding process. Table 5–6 indicate the numbers of tokens output after three sequential stages, classified by unambiguous tokens, unknown tokens and ambiguous tokens.

As the first stage, making use of a dictionary and named entity extraction, 218,237 out of 336,181 tokens, i.e., 64.92% of the tokens can be uniquely tagged. Later this type is called unambiguous tokens. Moreover, 81,170 tokens, i.e., 24.14% of the whole tokens, cannot be assigned with any tag. The tokens of this type are marked as unknown tokens. As the final type, 36,774 tokens, that is 10.94%, have more than one possible tag. Called as ambiguous tokens, such tokens are given with more than one tag. After the stage of using language resources, the second stage uses 125 manually constructed patterns to tag unknown tokens. By the patterns, we can tag additional 36,329 tokens uniquely. Therefore the uniquely identified token (i.e., unambiguous tokens) increases from 64.92% to 75.72% (i.e., 254,566 tokens) after this stage while the portion of unknown tokens reduces from 24.14% to 13.34% (i.e., 44,841 tokens). Table 7 indicates the detail of tag change from unknown to unambiguous. By the patterns, the top-5 most (among 10 possible tags) tagged tokens are in the type of COMMENT (comment), NUM (number), PUNC (punctuation), NOUN (noun) and POS (position). As the last stage, the statistical-based tagging can determine a suitable tag for 26,639 ambiguous tokens. By this the number of identified tokens increases to 281,205 tokens, which is 83.65% of the whole tokens. The number of ambiguous tokens reduces from 36,774 tokens to 10,135 tokens, i.e., 3.01% of the whole tokens.

Table 8 displays the details of the tag change from

Table 4 Types of entities and their occurrences in the experimental corpus.

Entity	#Tokens
Person	5,010
Verb	4,208
Location	3,781
Time	2,571
Date	1,386
Adverb	1,342
Conjunction	658
Position	467
Family relationship	99
Question phrase	6
TOTAL	19,528

Table 5 Numbers of tokens after three sequential tagging stages, classified by unambiguous tokens, unknown tokens and ambiguous tokens.

Token Type	Dictionary-based	Pattern-based	Statistical-based
#Tag = 1, Tag ≠ UNK (Unambiguous tokens)	218,237 (64.92%)	254,566 (75.72%)	281,205 (83.65%)
#Tag = 1, Tag = UNK (Unknown tokens)	81,170 (24.14%)	44,841 (13.34%)	44,841 (13.34%)
#Tag > 1, Tag ≠ UNK (Ambiguous tokens)	36,774 (10.94%)	36,774 (10.94%)	10,135 (3.01%)
Total tokens	336,181 (100.00%)	336,181 (100.00%)	336,181 (100.00%)

Table 6 Numbers of tokens after three sequential tagging stages, classified by unambiguous tokens, unknown tokens and ambiguous tokens in detail.

Token	Tag	Dictionary-based	Pattern-based	Statistical-based
Unambiguous tokens	ADJ	0	0	0
	ADV	1,342	1,789	3,163
	AUX	0	0	0
	CLAS	335	335	1,599
	COMMENT	0	14,331	14,331
	CONJ	11,536	11,536	14,684
	DAT	1,386	1,386	1,386
	DET	1,247	1,247	1,471
	END	34	34	35
	ENG	0	249	249
	FAM	99	99	99
	INT	19	19	19
	LOC	3,781	4,737	4,737
	NOUN	56,048	59,434	69,830
	NUM	0	6,867	6,867
	PER	5,010	5,010	5,010
	POS	467	3,205	3,205
	PREP	4,621	4,621	8,676
	PRON	2,974	2,974	3,526
	PUNC	0	4,350	4,350
	QUE	271	271	294
	SPC	61,736	62,794	62,794
	TIM	2,571	2,571	2,571
	VERB	64,760	66,707	72,309
		218,237	254,566	281,205
Unknown tokens	UNK	81,170	44,841	44,841
Ambiguous tokens		36,774	36,774	10,135
		336,181	336,181	336,181

Table 7 The statistical results of the tag change from unknown tokens in the dictionary-based tagging stage to unambiguous tokens in the pattern-based tagging stage (UnK → UnA).

UnK → UnA	#Tokens
UNK → UNK	44,841
UNK → COMMENT	14,331
UNK → NUM	6,867
UNK → PUNC	4,350
UNK → NOUN	3,386
UNK → POS	2,738
UNK → VERB	1,947
UNK → SPC	1,058
UNK → LOC	956
UNK → ADV	447
UNK → ENG	249
TOTAL	81,170

ambiguous tokens to unambiguous tokens. As shown in the table, the number of tags of the most ambiguous token is 5. There are totally 20 types of two-tag tokens, 12 types of three-tag tokens, one type of four-tag tokens and one type of five-tag tokens. The numbers of ambiguous tokens with 2-5 tags are 24187, 8604, 3962 and 21, respectively. After the statistical-based tagging, a large number of tokens obtain a single tag. The numbers of ambiguous tokens with 2-5 tags changed to unambiguous tags are 18086, 5587, 2947 and 19, respectively. The ten most frequent types are (1) NOUN;VERB, (2) CLAS;NOUN, (3) CLAS;NOUN;PREP;PRON, (4) CONJ;PREP, (5) ADJ;ADV;AUX, (6) NOUN;PREP, (7) NOUN;PREP;VERB, (8) PREP;VERB, (9) CLAS;

NOUN;VERB, and (10) CONJ;NOUN. These ten types cover around 82.89% of the token ambiguous tokens with 72.60% ambiguity resolution. On average, 72.44% of ambiguous tokens can be resolved to unambiguous ones.

5. Discussion

This subsection gives a number of discussions on the experimental results. Since the main objective of this work is to construct a large Thai tagged corpus and there is no gold standard for tagging, we can check only the tagging precision (later use correct or correctness to avoid confusion) but not tagging recall.

In the chunking process, the correctness of extracted close-type NEs and indicative expressions (i.e., date/time expression, family relationship expression, clue-based location name and position title) is 100% since they usually have explicit and unambiguous clues. Moreover, the extraction of words with more than two syllables for adverbs, conjunctions, question phrases and verbs, is quite perfect. However, extraction of person names gains approximately 91.95% correct since Thai language has no word boundary and sometimes strings that represent person titles (e.g., Mr., Ms., Dr., etc.) may not be the real one but a part of some words or some names. By using these incorrect person titles, some strings are misrecognized as person names. Moreover, for recognizing verbs or action parts in the chunking process, a string that represents a short verbal word may be misrecognized if it is a part of a longer word or name. By this situation, we restrict the recognition of verbal words to those

Table 8 The statistical results of tag change from ambiguous tokens in the pattern-based tagging stage (AmB) to unambiguous tokens in the statistical-based tagging stage (UnA).

#Tag	Multi-Entity Tag	#AmB	#UnA	#UnA/ #AmB (%)	Cumulative #AmB	Cumulative #AmB (%)	Cumulative #UnA	Cumulative #UnA (%)	Cumulative #UnA/ #AmB (%)
2	NOUN;VERB	6,402	5,188	81.04%	6,402	17.41%	5,188	19.48%	81.04%
2	CLAS;NOUN	5,856	4,397	75.09%	12,258	33.33%	9,585	35.98%	78.19%
4	CLAS;NOUN;PREP;PRON	3,962	2,947	74.38%	16,220	44.11%	12,532	47.04%	77.26%
2	CONJ;PREP	3,257	2,304	70.74%	19,477	52.96%	14,836	55.69%	76.17%
3	ADJ;ADV;AUX	2,543	665	26.15%	22,020	59.88%	15,501	58.19%	70.40%
2	NOUN;PREP	2,365	1,571	66.43%	24,385	66.31%	17,072	64.09%	70.01%
3	NOUN;PREP;VERB	1,618	1,401	86.59%	26,003	70.71%	18,473	69.35%	71.04%
2	PREP;VERB	1,575	1,300	82.54%	27,578	74.99%	19,773	74.23%	71.70%
3	CLAS;NOUN;VERB	1,498	1,297	86.58%	29,076	79.07%	21,070	79.09%	72.47%
2	CONJ;NOUN	1,407	1,019	72.42%	30,483	82.89%	22,089	82.92%	72.46%
3	CONJ;NOUN;PREP	1,107	844	76.24%	31,590	85.90%	22,933	86.09%	72.60%
2	CONJ;END	683	490	71.74%	32,273	87.76%	23,423	87.93%	72.58%
2	PRON;VERB	614	468	76.22%	32,887	89.43%	23,891	89.68%	72.65%
2	CONJ;VERB	496	405	81.65%	33,383	90.78%	24,296	91.20%	72.78%
3	CLAS;NOUN;PREP	496	378	76.21%	33,879	92.13%	24,674	92.62%	72.83%
3	CLAS;NOUN;PRON	488	400	81.97%	34,367	93.45%	25,074	94.13%	72.96%
2	NOUN;PRON	414	307	74.15%	34,781	94.58%	25,381	95.28%	72.97%
2	DET;PRON	405	140	34.57%	35,186	95.68%	25,521	95.80%	72.53%
3	CONJ;PREP;VERB	380	297	78.16%	35,566	96.72%	25,818	96.92%	72.59%
2	DET;PREP	318	201	63.21%	35,884	97.58%	26,019	97.67%	72.51%
2	CLAS;VERB	237	197	83.12%	36,121	98.22%	26,216	98.41%	72.58%
3	DET;NOUN;VERB	189	152	80.42%	36,310	98.74%	26,368	98.98%	72.62%
3	DET;PRON;QUE	128	49	38.28%	36,438	99.09%	26,417	99.17%	72.50%
2	DET;NOUN	115	80	69.57%	36,553	99.40%	26,497	99.47%	72.49%
3	DET;END;VERB	108	75	69.44%	36,661	99.69%	26,572	99.75%	72.48%
3	DET;NOUN;QUE	48	28	58.33%	36,709	99.82%	26,600	99.85%	72.46%
2	CLAS;PREP	28	14	50.00%	36,737	99.90%	26,614	99.91%	72.44%
5	CLAS;CONJ;NOUN;PREP;VERB	21	19	90.48%	36,758	99.96%	26,633	99.98%	72.45%
2	NOUN;QUE	7	1	14.29%	36,765	99.98%	26,634	99.98%	72.44%
2	INT;VERB	3	2	66.67%	36,768	99.98%	26,636	99.99%	72.44%
2	INT;NOUN	2	2	100.00%	36,770	99.99%	26,638	100.00%	72.44%
2	CLAS;PRON	2	0	0.00%	36,772	99.99%	26,638	100.00%	72.44%
2	END;INT	1	0	0.00%	36,773	100.00%	26,638	100.00%	72.44%
3	CLAS;INT;NOUN	1	1	100.00%	36,774	100.00%	26,639	100.00%	72.44%
TOTAL		36,774	26,639	72.44%					

with more than two syllables.

In the word segmentation process, the correctness decreases when the processed text contains words that do not exist in the dictionary or list of words. Longest matching algorithm can be considered as using some heuristics to solve the ambiguity problem by selecting the longest possible term. However, using longest matching algorithm has generally an overlapping problem. For example, if we have 'ABC' letter sequences which can be segmented into either 'AB-C' or 'A-BC' depending on the context, the longest match using left-to-right direction search always fail to find 'A-BC' segmentation. Even if word segmentation process can not achieve 100% correctness, the output tokens from segmentation 'AB' and 'C' can be assigned or classified with possible parts-of-speech defined by the reliable dictionary in the dictionary-based tagging stage. From the experimental results, we obtained 13.34% unknown tokens.

In the dictionary-based tagging stage, the performance depends on the reliability of the dictionary and the output from the prior chunking steps. In this work, a token was assigned with possible parts-of-speech defined in well-known dictionaries. From this assumption, all unambiguous tokens,

unknown tokens and ambiguous tokens were automatically classified, although the chunking process can not achieve 100% correctness.

In the pattern-based tagging stage, our patterns can transform 36,329 unknown tokens to unambiguous tokens with 100% correctness. That is, the number of unknown tokens reduces from 81,170 to 44,841. These remaining unknown tokens are usually misspelling words or unregistered words. To solve this problem, some manual processes are required.

In the statistical-based tagging stage, 26,639 ambiguous tokens were transformed to be unambiguous tokens in this stage, i.e., the reduction of 36,774 ambiguous tokens to 10,135 tokens. The correctness of the statistic-based process which is evaluated over 200 from 26,639 examples is 148 (74%). The statistical-based process selects the best entity tag with the highest probability for ambiguous tokens, among their possible tags. If the best tag is not in the possible tag set of the word, the tags of ambiguous tokens will be left without determination.

There might be errors even in chunking process (NE extraction and word segmentation). However, such er-

rors can be handled by treating as unknown tokens in the dictionary-based tagging stage. We can conclude that the output from the dictionary-based tagging stage and the pattern-based tagging stage achieved 100% correctness. Based on statistics, the statistical-based tagging stage can help to reduce unambiguous tokens. The proposed framework can detect boundary of words and named entities, nevertheless, detecting the sentence boundary is necessary for further annotation (e.g., parsing) since sentence boundaries play an important role in summarizing linguistic phenomena. Sentence extraction stage may process before the tagging stage in the proposed framework.

Our multi-stage annotation framework helps to reduce the manual effort in constructing a Thai entity annotated corpus. From this framework, the expert can focus and speed up on selecting the correct PoS from all possible parts-of-speech provided by the dictionary for ambiguous tokens and correct unknown tokens, which are only 16.35% to complete the annotated corpus construction (3.01% from ambiguous tokens and 13.34% from unknown tokens). In order to construct an annotated corpus where all annotations are correct, human experts should check not only unknown and ambiguous tokens but also unambiguous tokens. Since the accuracies of the automatically determined word segments and tags in multi-stage annotation are high enough, the proposed system would alleviate human burden even though experts should check all tokens.

We compare the annotated corpus with other existing corpora. In the Thai language, even we have several constructed corpora, they are various in types and information in the corpora. Currently there is one well-known public PoS-annotated corpus namely the ORCHID Corpus [9] while the other corpora are specific to word segmentation or speech or tailored to some specific domains, such as the BEST corpus[†] and the LOTUS speech corpus [12]. Contrasting our approach with the other corpora, the following conclusions can be made.

- (Corpus Size) The pilot corpus output from our proposed method is smaller with 336,181 tokens but includes both NE tags and PoS tags. The ORCHID corpus is larger with approximately 2,560,000 words but includes only PoS tags. The BEST corpus includes 5,000,000 words with only word boundary marks for word segmentation research. The LOTUS corpus is a corpus used for large vocabulary Thai continuous speech recognition with 70-hour speech from 50 speakers with 5000 words vocabulary size.
- (Manual vs. Semi-automatic Construction) Our proposed method aims to reduce the labor by applying both patterns and statistics to automate the tagging process while the ORCHID corpus, the BEST corpus and the LOTUS corpus are constructed manually with labor-intensive process.
- (Annotation Level) The corpus created by our proposed method includes both PoS and NE, the ORCHID cor-

pus has only PoS, the BEST corpus has no PoS and NE but word boundary marks and the LOTUS corpus does not facilitate the text process. We propose to annotate both PoS and NE in order to enable us to create a parse tree in the future.

Currently since our method relies on dictionary, we assume that all possible tags for a known word is given. However, in the real situation, it is possible to face with some missing tags for a known word. In our preliminary observation, we found out that we have two cases of this situation as follows.

1. A missing tag for a token with multiple possible tags (Ambiguous token)
2. A missing tag for a token with a single tag (Unambiguous token)

For the first situation of ambiguous tokens, we have done preliminary observation on the 10,135 ambiguous tokens. We found out that approximately 2.61% (265/10,135) of them have some missing possible tags which are not assigned to the word in the dictionary, while 1.50% (3/200) is obtained by the second situation of unambiguous tokens. In the future, we plan to cope with these two situations by allowing statistics to help us obtain the missing tags. We recognize that it is possible to have the best tags chosen by the tagger is not included in the possible tags. As our future work, we will consider this kind of issues.

6. Conclusion and Future Work

This paper has presented a practical solution for large-scale corpus construction to reduce the amount of manual annotation. Our multi-stage annotation framework, consisting of two stages of chunking and three stages of tagging, can automatically discover unknown tokens and ambiguous tokens. As future work, we plan to exploit semi-supervised learning schemes to recognize named entities and segment words. Detecting more new content entities among various domains and measuring tagger reliability should also be investigated.

Acknowledgments

This work has been supported by the Thailand Research Fund (TRF), under Grant No. BRG50800013, a TRF Royal Golden Jubilee Ph.D. Program, under Grant No. PHD/0034/2550, the National Electronics and Computer Technology Center, Thailand, under Grant No. NT-B-22-KE-38-54-01, Government Research Fund via Thammasat University, Thailand, and the National Research University Project of Thailand Office of Higher Education Commission.

References

- [1] S.L. Pandian and T.V. Geetha, "Crf models for tamil part of speech tagging and chunking," ICCPOL '09: Proc. 22nd International Conference on Computer Processing of Oriental Languages. Language

[†]<http://thailang.nectec.or.th/best/>

- Technology for the Knowledge-based Economy, Berlin, Heidelberg, pp.11–22, Springer-Verlag, 2009.
- [2] Z. Huang, V. Eidelman, and M. Harper, “Improving a simple bigram HMM part-of-speech tagger by latent annotation and self-training,” NAACL ’09: Proc. Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, Morristown, NJ, USA, pp.213–216, Association for Computational Linguistics, 2009.
 - [3] S. Lee, J. An, B.K. Kwak, and G.G. Lee, “Learning korean named entity by bootstrapping with web resources,” IEICE Trans. Inf. & Syst., vol.E87-D, no.12, pp.2872–2882, Dec. 2004.
 - [4] K.M. Park and H.C. Rim, “Semantic classification of bio-entities incorporating predicate-argument features,” IEICE Trans. Inf. & Syst., vol.E91-D, no.4, pp.1211–1214, April 2008.
 - [5] N. Tongtep and T. Theeramunkong, “Simultaneous character-cluster-based word segmentation and named entity recognition in thai language,” Proc. Fifth International Conference on Knowledge, Information and Creativity Support Systems (KICSS 2010), Nov. 2010, pp.167–175, 2010.
 - [6] D.G. Lee, G. Hong, S.K. Lee, and H.C. Rim, “Minimizing human intervention for constructing korean part-of-speech tagged corpus,” IEICE Trans. Inf. & Syst., vol.E93-D, no.8, pp.2336–2338, Aug. 2010.
 - [7] H. Loftsson, “Correcting a pos-tagged corpus using three complementary methods,” EACL ’09: Proc. 12th Conference of the European Chapter of the Association for Computational Linguistics, Morristown, NJ, USA, pp.523–531, Association for Computational Linguistics, 2009.
 - [8] R. Sasano, D. Kawahara, and S. Kurohashi, “The effect of corpus size on case frame acquisition for predicate-argument structure analysis,” IEICE Trans. Inf. & Syst., vol.E93-D, no.6, pp.1361–1368, June 2010.
 - [9] H. Isahara, Q. Ma, V. Sornlertlamvanich, and N. Takahashi, “Orchid: building linguistic resources in thai,” Lit. Linguist Computing, vol.15, no.4, pp.465–478, April 2000.
 - [10] T. Charoenporn, C. Kruengkrai, T. Theeramunkong, and V. Sornlertlamvanich, “Construction of thai lexicon from existing dictionaries and texts on the web,” IEICE Trans. Inf. & Syst., vol.E89-D, no.7, pp.2286–2293, July 2006.
 - [11] H. Chanlekha and A. Kawtrakul, “Thai named entity extraction by incorporating maximum entropy model with simple heuristic information,” Proc. IJCNLP 2004, pp.49–55, 2004.
 - [12] P. Cotsomrong, T. Sunpetchniyom, S. Kasuriya, N. Thatphithakku, and C. Wutiwiwatchai, “Lotus: Large vocabulary thai continuous speech recognition corpus,” NSTDA Annual Conference S&T in Thailand: Towards the Molecular Economy (NAC2005), 2005.
 - [13] T. Theeramunkong, M. Boriboon, C. Haruechaiyasak, N. Kittiphattanabawon, K. Kosawat, C. Onsuwan, I. Siriwat, T. Suwanapong, and N. Tongtep, “THAI-NEST: A Framework for Thai Named Entity Tagging Specification and Tools,” CILC ’10: Proceedings of the 2nd Int’l Conference on Corpus Linguistics (CILC10), May 13–15, 2010, pp.895–908, May 2010.
 - [14] Y. Zhang and V. Kordoni, “Discriminant ranking for efficient treebanking,” COLING (Posters), ed. C.R. Huang and D. Jurafsky, pp.1453–1461, Chinese Information Processing Society of China, 2010.
 - [15] M. Seraji, B. Megyesi, and J. Nivre, “Bootstrapping a persian dependency treebank,” Linguistic Issues in Language Technology, vol.7, no.18, 2012.
 - [16] S. Kurohashi and M. Nagao, “Building a japanese parsed corpus while improving the parsing system,” in Abeillé, A. (ed.) Treebanks: Building and Using Parsed Corpora, ch. 14, pp.249–260, Kluwer Academic Publishers, 2003.
 - [17] P.T. Daniels and W. Bright, The World’s Writing Systems, Oxford University Press, 1996.
 - [18] T.H. Koanantakool, T. Karoonboonyanan, and C. Wutiwiwatchai, “Computers and the thai language,” IEEE Ann. Hist. Comput., vol.31, no.1, pp.46–61, 2009.
 - [19] C. Haruechaiyasak, S. Kongyoung, and M. Dailey, “A comparative study on thai word segmentation approaches,” Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on, pp.125–128, May 2008.
 - [20] C. Haruechaiyasak, “Lexto: Thai lexeme tokenizer.” <http://sansarn.com/lexto/download/LongLexTo.zip>, 2006.
 - [21] N. Tongtep and T. Theeramunkong, “Discovery of predicate-oriented relations among named entities extracted from thai texts,” IEICE Trans. Inf. & Syst., vol.E95-D, no.7, pp.1932–1946, July 2012.
 - [22] N. Ketui, T. Theeramunkong, and C. Onsuwan, “A rule-based method for thai elementary discourse unit segmentation (ted-seg),” Proceedings of the Fifth International Conference on Knowledge, Information and Creativity Support Systems (KICSS 2012), Oct., 2012, pp.195–202, 2012.

Nattapong Tongtep received his B.Sc in Computer Science from Prince of Songkla University, Thailand. He is currently a student in combined M.Sc/PhD. program, School of Information, Computer and Communication Technology (ICT), Sirindhorn International Institute of Technology, Thammasat University, Thailand. His research interests are information extraction, natural language processing and data mining.

Thanaruk Theeramunkong received a bachelor degree in Electric and Electronics, and master and doctoral degrees in Computer Science from Tokyo Institute of Technology in 1990, 1992 and 1995, respectively. His current research interests include data mining, machine learning, natural language processing, and information retrieval.