Character-based Thai Named Entity Recognition

Dittaya Wanvarie[†]　Hiroya Takamura[††]　Manabu Okumura[††]

[†]Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology

[††]Precision and Intelligence Laboratory, Tokyo Institute of Technology

dittaya@lr.pi.titech.ac.jp　{takamura,oku}@pi.titech.ac.jp

1 Introduction

Named Entity Recognition (NER) is the task to identify word or phrase of Named Entity (NE) such as name of person, organization, in a document. NER roughly consists of 2 steps, the NE boundary identification and the NE type classification. We consider only the NE boundary identification in this paper.

NER task in major languages such as English is well explored and rather successful with high accuracy [8]. In contrast, there is lack of interest in the same task of minor languages including Thai. Although we can apply a successful learning technique in one language to another language, the same technique may not be successful due to differences in characteristics of languages. Moreover, some previous works [8] rely on external knowledge such as WordNet, which requires human effort to beuilt. Such knowledge is not available in most minor languages.

Every language shares a characteristic that a word consists of characters, although there are several different types of characters. We can apply a character-based model to any languages without any prior linguistic knowledge. An NE which usually is an unknown word, is problematic in word-level processing because it requires an additional process to be handled. In contrast, there is no unknown character since the character class is finite. We can automatically handle the unknown word problem with the character-level processing.

To our knowledge, there are few previous works in Thai NER task. Kijsirikul et al.[4] proposed a dictionary-based method for the task. A sequence is segmented into words with corresponding part-of-speech (POS) by a word 3-gram tagger. Dictionary and heuristic rules were applied to generate possible NE candidates from the word/tag sequence. The system recognizes NEs from the n-best word/tag candidates using a voting-perceptron-like algorithm. They used manually word-segmented 25K-word corpus, which approximately contains 1800 NEs. Their best recall of the NE candidate generating process was 91.94%. The NE recognizer achieved the F_1 score of 93.58% from the generated output. The accuracy relies on the size of dictionary and the recall rate of the NE candidate generating process.

Chanleka and Kawtrakul[2] proposed to use a Maximum Entropy model with heuristic cues for the Thai NER task. They used a manually word-segmented 10K-word corpus annotated with 3 sub-classes of NEs, person name, location, and organization. NE candidates are generated by a rule-based system, NE dictionary matching, and statistics cues. A Maximum Entropy model was applied to recognize NEs from the candidate list. After a post-processing, their best model, the word 3-gram model, achieved 87.70% of F_1 measures on average of all NE categories.

Works of Kijsirikul et al.[4] and Chanleka and Kawtrakul[2] were done in word-level. However, there is currently no high performance word boundary identification model for Thai. In this paper, we propose to do both the NE candidates generating and NE recognizing task in character level. We do not require prior word boundary information.

Klein et al.[5] analyzed the character-level features for English and German NER task. Their result shows that the all-substring Hidden Markov Model (HMM) outperforms the word-based HMM because the all-substring features have already subsumed the word feature. With all-substring features, POS tag, contextual features, sequential features, and other error driven features, their best model achieved the F_1 score of 86.31% for English and 71.90% for German at CoNLL 2003 shared task.

Asahara and Matsumoto [1] proposed a character-based Japanese NER system using n-best morphological analysis output on Support Vector Machines (SVMs). The redundant n-best word/POS answers were proposed to alleviate the error propagation problem. With n-best morphological information, character types, and relative position, they achieved F_1 measure of 87.21% on CRL NE data.

Chen et al.[3] proposed the state-of-the-art Chinese NER system at SIGHAN 2006 bakeoff 3. Character n-gram features, word boundary and keyword features were automatically extracted from the corpus and used with Conditional Random Fields (CRFs) to generate and classify an NE into its appropriate category. Their system achieved the F-score of 85.14% at MSRA, 89.03% at CityU, and 76.27% at LDC corpus respectively.

Chinese and Thai are very similar in that there is no explicit word boundary and no explicit clues of NE like capitalization in English. However, Chinese is a logographic language whose characters contain meaning, while Thai is a

segmental language whose characters consist of consonants and vowels. A character-based model in Chinese could be classified as a sub-morpheme based system due to the meaning in the character. Although each Thai character contains no meaning, we assume that some consecutive characters could be grouped together to create a *pseudo-morpheme* by only simple heuristic rules.

In this paper, we propose a character-based model for Thai NER task. We also analyze the effect of class label granularity and the size of context window. The result show that only simple raw character features are effective in the task.

2 Learning model

We consider the NER task as a sequential labeling task. In our character-based model, we classify each character into a beginning-NE ("B"), an inside-NE ("I", or an outside-NE ("O") class. An NE is a character sequence starting with a character in the beginning-NE class and the rest of characters are in the inside-NE class. The system is evaluated in word-level.

In this paper, we adopt Conditional Random Fields (CRFs)[6] as our labeling algorithm.

A CRFs model in sequential labeling tasks attempts to assign target labels to the input sequence using the conditional probability of the target sequence and the input sequence. Let $\mathbf{y} = y_1 \dots y_t$ denotes a sequence of target label for an observation sequence $\mathbf{x} = x_1 \dots x_t$ for $t = 1, ..., T$ where T is the sequence input length. $\mathbf{y}$ and $\mathbf{x}$ are sets of random variables. Let c be the clique index of a clique $\{\mathbf{y}_c, \mathbf{x}_c\} \in C$ in the undirected model over $\mathbf{x}$ and $\mathbf{y}$. A linear CRFs model defines the conditional probability as follows:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{\{\mathbf{y}_c, \mathbf{x}_c\} \in C} \Phi(\mathbf{y}_c, \mathbf{x}_c).$$

Φ is the potential function for a clique and the partition function $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{c \in C} \Phi(\mathbf{y}_c, \mathbf{x}_c)$ is a normalization factor over all state sequences for the sequence $\mathbf{x}$.

3 Experiments

Our corpus approximately contains 4.6M words with word boundary and NE annotation. Each NE is considered as a single word in this corpus, i.e. there is no word boundary inside an NE. We randomly selected 5% of data to be our validation set, and performed 5-fold cross-validation with the rest of data.

3.1 Outline

According to Thai lexical grammar, not all combinations of characters are permitted. For example, some characters should be placed only in front of another character, while some others should be placed after another character. To avoid the segmentation result that may be against the grammar, we will conduct a pre-processing by grouping some characters with either their preceding or following characters according to the lexical rules. All rules are summarized as follows:

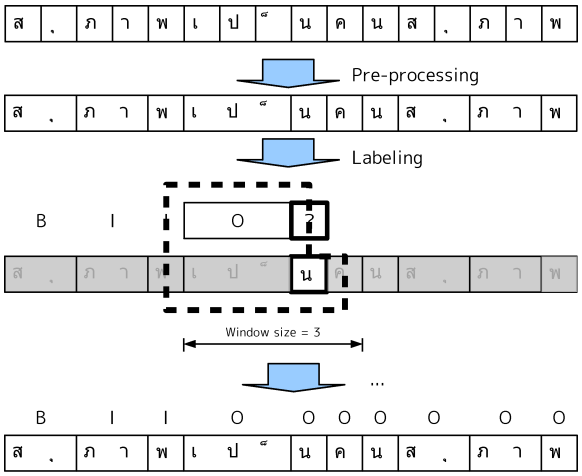

Figure 1: Model outline

- A whitespace character is an explicit word boundary.
- Consecutive digits are grouped together.
- Non-Thai characters are grouped together.
- Non-starting vowels, which are vowels that cannot be placed at the beginning of a word, are grouped with their preceding characters.
- Non-starting symbols, which are symbols that cannot be placed at the beginning of a word, are grouped with their preceding characters.
- Tonal marks are grouped with their preceding characters.
- Non-ending vowels, which are vowels that cannot be placed at the end of a word, are grouped with their following characters.

A word in our corpus contains 3.98 characters on average before the pre-processing. The average size of a word is reduced to 2.39 characters after the pre-processing. From the following, we will use a term "character" for a group of characters after the pre-processing.

Thai is an isolating language in which almost every word consisting of a single morpheme[7]. Word statistics in Table 1 confirms that we can regard a character after pre-processing as a *pseudo-morpheme* since there are 39.24% of

words consisting of a single character. These single character words therefore contain some meaning. We can enlarge the context window to more than 1 character in order to capture words consisting of several characters. Our context window could be considered as a *pseudo-morpheme*. With the context window of size 3, we can capture 86.17% of words in this corpus. As a result, we can perform experiments in character-level with contexual features similar to the processing in morpheme-level.

3.2 Baseline model

We used the IOB label set in the baseline settings. Characters in "B" and "I" classes are the beginning and the intermediate characters of an NE respectively. The other characters are classified into the "O" class, which is the class for characters outside any NE.

The context window size W was set to 3 and consisted of the preceding character, the current character, and the following character. We used the character 1,2,3-gram features in each window.

Figure 1 shows the outline of our baseline model for the following input sequence:

สุภาพ	เป็น	คน	สุภาพ	
Suparb	*pen*	*khon*	*suparb*	
Suparb	is	man	polite	(Suparb is a polite man.)

The character "น" in the third line is the character in focus. The window size of 3, starting from "เป็" to "ค". The dash box consisting of characters "เป็" and "น", is an example of a 2-gram feature in this window. We also consider the label of the preceding character, which is "O" of the character "เป็" in this example, as one of our features.

We achieved 90.49% of precision but rather low recall of 85.41% from our baseline system.

3.3 Effectiveness of word boundary information

The NE boundary identification task in this paper is divided into 2 sub-tasks: the word boundary identification and the NE recognition task. The effectiveness of character-level features is also evaluated in each subtask.

The baseline system without word boundary information (WS model: none) was trained with character n-gram model and achieved 87.64% in precision and 82.72% in receall. When we added correct word boundary information (WS model: oracle) to the baseline model, we obtained a significant increase from the baseline to 95.76% in precision and 93.44% in recall.

A word segmentation model was trained with the character 1,2,3-gram features (WS model: char n-gram)[1]. Although 94.66% of the NEs are correctly segmented with the word segmentation model (NER model: oracle), the NE recognizer trained on the correct word boundary information (NER model: char n-gram) achieves only 75.71% of F_1 measure. The significantly decrease shows that the NE recognizer heavily depends on the word boundary information and is not robust to the erroneous input.

We re-trained the NE recognizer with the extracted word boundary information and the raw character n-gram input in order to alleviate the effect from word boundary errors. We obtained the NER F_1 score of 85.19%, which is comparable to the model trained with character n-gram features alone.

The extracted word boundary features did not improve the overall accuracy of the system while the correct word boundary information significantly improved the accuracy. For this reason, we can conclude that using only raw character n-gram features is effective and efficient when a high accuracy word segmentation tool is not available. The comparison between the models with and without word boundary information is summarized in Table 2.

3.4 Contextual features vs. granularity of label set

All previous settings suffer from low recall. One possible reason is that the simple IOB label set cannot capture the word boundary information. Instead of directly extracting the word boundary information from the input sequence, we incorporate the information in the label set. We have created a fine-grain label set, "C/N-B/I", where "C-B" and "C-I" are the beginning and the intermediate characters of a common word, and "N-B" and "N-I" are the beginning and the intermediate characters of an NE. With this fine-grain settings, we obtained 1.01% increase in recall but a slightly drop in precision from the coarse-grain settings. In total, we obtained an small increase in F_1 measure from 85.11% to 85.33%.

We also tried a finer-grain label set by adding relative position information to the "C/N-B/I" label set. The new label set is "C/N-B/I-#", where # is the relative position of the character in the word. We reduced the position information to 1, 2, 3, 4, and 5 or over, in order to reduce the computational cost. We obtained an increase in both precision and recall and resulting in the increase of F_1 score to 86.63%.

However, the accuracy improvement from finer-grain class was less significant than the improvement from long contextual features. Moreover, the fine-grain class label set did not improve much accuracy when long context is available. With W=9, the model with simple IOB label set obtained a comparable F_1 score of 88.37% to 88.40% of F_1 score from the C/N-B/I label set. The comparison of effect from the window size and the grain of label set are shown in Table 3.

One possible explanation is that most of the word boundary information is embedded in a long context. A single window of size 9 can capture more than 95% of words in the corpus while a window of size 3 can capture only 86.17%

[1] We discard all NEs in the training of the word segmentation model since NEs are not segmented into short words in this corpus

Word size	**1**	**2**	**3**	**4**	≥ 5	AvgChar
Common word	39.20	30.12	16.26	6.48	4.91	2.23
Ambiguous NE	0.02	0.17	0.19	0.31	0.29	4.30
Unambiguous NE	0.02	0.08	0.21	0.28	1.56	8.20
Total	39.24	30.47	16.46	6.97	6.76	2.39

Table 1: Word statistics

WS Model	**NER model**	**Precision**	**Recall**	F_1
none	char n-gram <*baseline*>	87.64	82.72	85.11
oracle	char n-gram	95.76	93.44	94.58
char n-gram	oracle	100.00	94.66	97.26
char n-gram	char n-gram	87.71	82.81	85.19

Table 2: Word boundary effect to the model, window size = 3

Model	**Precision**	**Recall**	F_1
Less context: W=3			
IOB <*baseline*>	87.64	82.72	85.11
C/N-B/I	86.99	83.73	85.33
C/N-B/I-#	88.29	85.03	86.63
More context: W=9			
IOB	90.69	86.17	88.37
C/N-B/I	89.91	86.93	88.40

Table 3: Comparison between an effect of the window size and the grain-level to the model

of words in the corpus.

The higher precision and recall should be traded with the more memory consumption for the rich context window and the fine-grain label set. Suppose that we have a character class of size N, with character n-gram features and the label of preceding character, in context window of size W, and a label class of size L, the memory complexity is $O(N^n W L^2)$. Increasing the window size only increases the complexity in polynomial degree while increasing the grain-level size increases the complexity in exponential degree. We can conclude that the contextual features are more efficient than the grain-level size in terms of memory requirement.

4 Conclusion and future work

Additional features extracted from the input do not improve the overall accuracy of the model. Without any high accuracy information from external resource, using only character n-gram features is effective and efficient in Thai NER task. The effect of other conventional features comparing to the character-level features, is left for future work.

Extending context window is more effective and efficient than refining the grain-level of the label set in this task. However, the contextual feature has a disadvantage in any long range dependencies. In such cases, those features should be extracted and incorporated as additional features.

As every language shares the same characteristics that a word consists of characters. We can consider a character-based model as a language independent model. A further study on character-based model for other natural language processing task is worth to be explored.

References

[1] M. Asahara and Y. Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 8–15, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[2] H. Chanleka and A. Kawtrakul. Thai named entity extraction by incorporating maximum entropy model with simple heuristic information. In *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP)*, Hainan Island, China, 2004.

[3] W. Chen, Y. Zhang, and H. Isahara. Chinese named entity recognition with conditional random fields. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 118–121, Sydney, Australia, July 2006. Association for Computational Linguistics.

[4] B. Kijsirikul, P. Charoenpornsawat, and S. Meknavin. Comparing winnow and ripper in thai namedentity identification. In *Proceedings of the Natural Language Processing Pacific Rim Symposium*, 1999.

[5] D. Klein, J. Smarr, H. Nguyen, and C. D. Manning. Named entity recognition with character-level models. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 180–183, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[6] J. D. Lafferty, A. Mccallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[7] E. E. Loos, S. Anderson, D. J. Dwitght H., P. C. Jordan, and J. D. Wingate, editors. *Glossary of linguistics terms*, chapter Glossary (Linguistics): I. SIL International, 2003.

[8] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007.