Thai Named Entity Recognition using BiLSTM-CNN-CRF enhanced by TCC

VIRACH SORNLERTLAMVANICH¹,² and SUMETH YUENYONG³
¹Asia AI Institute, Faculty of Data Science, Musashino University, Tokyo 135-8181, Japan (e-mail: virach@musashino-u.ac.jp)
²Faculty of Engineering, Thammasat University, Thailand (e-mail: virach@gmail.com)
³Faculty of Engineering, Mahidol University, Thailand (e-mail: sumeth.yue@mahidol.edu)

Corresponding author: Virach Sornlertlamvanich (e-mail: virach@musashino-u.ac.jp).

This work is supported by the Thailand Science Research and Innovation Fundamental Fund under Contract No. TUFF19/2564 and TUFF24/2565.

ABSTRACT The languages spoken in Asia share common morphological analysis errors in word segmentation which normally propagate to higher-level processing, i.e., POS tagging, syntactic parsing, word extraction, and NER, as we discuss in this research. We introduce the Thai character cluster (TCC) to reduce the errors propagated from word segmentation and POS tagging by incorporating into character representation layer of BiLSTM for NER. The initial NER model is created from the original THAI-NEST named-entity tagged corpus by applying the best performing BiLSTM-CNN-CRF model with word, part-of-speech, and character cluster embedding. We determine the errors and improve the consistency of the NE annotation through our holdout method by retraining the model with the corrected training set. After the iteration, the overall result of annotation F1-score has been improved to reach 89.22%, which improves 16.21% from the model trained on the original corpus. The result of our iterative verification is a promising method for low resource language modeling. As a result, The NE silver standard corpus is newly generated for the Thai NER task, called BKD Corpus (Bangkok Data NE tagged Corpus). The consistency of annotation is checked and revised according to the improvement of the scope of NE detection by TCC which can recover the errors in word segmentation.

INDEX TERMS Low-resource languages, Named entity recognition, Thai language, Thai named entity annotation corpus

I. INTRODUCTION

This paper proposes a novel method of iterative NE tagging refinement that can be applied to a noisy NE corpus, to generate a silver standard BKD Corpus (Bangkok Data NE tagged Corpus) to solve the problem of the limited resources of the Thai language. Some difficulties of language processing in the fundamental issues are also a high barrier to overcome, to improve the language processing. Thai is an alphabetic language, with no explicit word or sentence boundary, and it is an isolated language, without grammatical markers. These issues can induce a vast amount of ambiguities in morpho-syntactic analysis. Therefore, the consistency problem in word segmentation and grammatical tag annotation is not trivial. The errors are always propagated to consecutive tasks such as word dependency, parse tree annotation, word sense disambiguation, and certainly in the current task of named-entity annotation. Since word segmentation is not in the scope of this paper, we manually correct the result when necessary and apply the state-of-art word segmentation based on the trigram part-of-speech (POS) tagging model [17]. The POS tagset is introduced from the ORCHID POS tagged corpus [20].

The THAI-NEST corpus [27] is currently the largest Thai NE corpus, collected from 21 Thai online newspaper publishers from January to December 2009. The collection contains a good balance in the variation of the text domain. In the corpus, word segmentation is applied and annotated with the POS tagset. On top of that, the seven types of NE tags, namely, date (DAT), location (LOC), measure (MEA), name (NAM), organization (ORG), person (PER), and time (TIM) are annotated to the corresponding words. The corpus is manually
annotated with one type of NE for a particular file. The size of the corpus is significantly large. It is a corpus of approximately seven million words or about 80 thousand sentences, as shown in Table 1. However, it needs a proper data cleansing process, especially for the word segmentation errors and inconsistency in NE tagging, as we preliminarily conducted a consistency test on the corpus and found that the errors have a significant effect on the accuracy of the NER task.

The accuracy of word segmentation has a strong impact on the quality of the corpus. The error normally propagates to higher-level processing in producing the features of word spelling and its POS labeling. To recover the errors, TCC is used to instead of character in many cases. [19] proposed TCC, the smallest standalone character unit according to the spelling rules, to represent the character in order to reduce the errors in determining the breakable positions in the string. For example, the next breakable position in the string after “กระทรวงการคลัง” is “กระทรวงการคลัง”, not “กระทรวงการกง” because the vowel sign “ิ” has to be combined with a base consonant like a diacritical sign. “ิ” is called a character cluster or Thai Character Cluster (TCC).

TCC can be defined by a set of spelling rules. There is no ambiguity in forming a cluster therefore there is no error in clustering and it can provide a better context to represent character-level feature of a word.

The main contribution of this research is to overcome the problem of the shortage and the quality of annotated corpus for model training and evaluation. The existing corpora, though there are not many, still have a big problem in the consistency of word segmentation and annotation. Using a noisy corpus for training certainly cannot expect a high precision out of the model. With the limitations of availability of the NE annotated corpus, we propose an efficient method to refine the existing corpus though it is full of errors because developing a new large corpus is labor intensive and costly. We refine the noisy corpus of THAI-NEST automatically to construct a so called silver standard corpus (automatically constructed corpus with comparable quality to the gold standard corpus) [7] for NER study.

In terms of improving the performance of NER, especially for the non-segmented language such as Thai, we found that utilizing the advantages of TCC representation instead of character in the character embedding layer of BiLSTM can mitigate the NER errors according the inaccurate word segmentation results. The approach is also viable for other non-segmented language having similar character composition clue such as Lao, Myanmar, Cambodian. Though it is out of the scope of this research, the larger unit of character composition can reduce the perplexity at the character representation level.

In this paper, we propose an efficient method to clean up a noisy corpus with language difficulties by state-of-the-art NER using the combination of bidirectional LSTM, CNN, and CRF (BiLSTM-CNN-CRF) [13]. Our novel approach of applying the Thai character cluster (TCC) proposed by [18] for character-level representation in the character-embedding layer performs better in BiLSTM-CNN-CRF with POS and word embedding [22], [23].

The paper is structured as follows. Section II summarizes the previous works on NE dataset development and the proposed tagset for preparing the NE corpus. Some effective approaches for the Thai language NER are discussed. Section III gives the information of the THAI-NEST corpus which is used in this study. The size and the annotation scheme are elaborated. Section IV discusses how CNN-TCC for character-level representation in the character embedding level can capture the NE spelling pattern to enhance the performance of BiLSTM-CNN-CRF for Thai NER. Section V describes the performance and the comparison results of each model when applying to the same corpora. Section VI proposes a method of iterative NE tagging refinement to improve the existing noisy corpus, and analysis results of the detected annotation errors.

II. RELATED WORKS

Many types of NE tagsets have been proposed. The types and number of tags are defined according to the groups and the tasks they are used in. The following are some examples of the representative tagsets used in the NER task: question-answering, information extraction, text summarization, and machine translation.

- CONLL-2003, reported in NER shared task dataset [29], is a well-known collection of Reuters 1,393 newswire articles that contains a large portion of sports news. It is annotated with four entity types (PER (person), LOC (location), ORG (organization), and MISC (miscellaneous)).

- MUC-6 [5] is a dataset consisting of newswire articles from the Wall Street Journal annotated with PER (person), LOC (location), ORG (organization), as well as a number of temporal and numerical entities.

- OntoNotes 5.0 dataset [8] is annotated with 18 fine-grained NE categories. Those categories are PERSON, NORP (Nationalities or religious or political groups), FACILITY, ORGANIZATION, GPE (Countries, cities, states), LOCATION, PRODUCT, EVENT, WORK OF ART, LAW, LANGUAGE, DATE, TIME, PERCENT, MONEY, QUANTITY, ORDINAL, and CARDINAL.

- IJCNLP-08 NERSSEAL shared task tagset is a dataset consisting of 12 fine-grained NE tags. Those tags are NEP (person), NED (designation), NEO (organization), NEA (abbreviation), NEB

1available at http://ltc.iiti.ac.in/ner-ssea-08
(brand), NETP (title-person), NETO (title-object), NEL (location), NETI (time), NEN (number), NEM (measure), and NETE (terms).

For the Thai language, there is a THAI-NEST corpus which is word-segmented and annotated with POS and seven types of NE tags, namely, date (DAT), location (LOC), measure (MEA), name (NAM), organization (ORG), person (PER), and time (TIM). The tagset is detailed enough for common tasks but due to Thai language difficulties in word segmentation and POS tagging, it is difficult to find common agreement in the annotation. These morphological errors cause difficulties in higher levels of NE annotation.

[12] has exhaustively surveyed NER research and classified the approaches into (i) Rule-based approach, which does not need annotated data as it relies on hand-crafted rules; (ii) Unsupervised learning approach, which relies on unsupervised algorithms without hand-tagged training examples; (iii) Feature-based supervised learning approach, which relies on supervised learning algorithms with careful feature engineering; (iv) Deep-learning-based approach, which automatically discovers representations needed for the classification and/or detection from raw input in an end-to-end manner. The state-of-the-art NER in the deep-learning-based approach has been proposed by [13], using the combination of BiLSTM, CNN and CRF (BiLSTM-CNN-CRF). An experiment has been conducted on the CoNLL-2003 corpus, obtaining 91.21% F1 for the NER task.

Some approaches have been applied to the Thai language NER task. In the rule-based approach, [3] made a survey to show that NE lexicon and clue words for NE can be used to create a rule set for extracting and annotating the class. For example, province name, person name, or company name usually follow a particular word, such as "นาย" (Mr.), "นางสาว" (Miss), and "นาง" (Mrs.), respectively. In case of the name without a clue word, the frequency of word co-occurrence is used to give a threshold for selecting the NE. They combined the heuristic rule set and the frequency of word co-occurrence threshold to annotate PER, ORG and LOC of 200 articles of Kinnaree Magazine and newspapers. The results showed an average precision of 78.8% and average recall of 66%. This study has shown that it is possible to use clue words to extract and classify the NE. [24] proposed a method to extract Thai personal named entity without relying on word segmentation or POS tagging to avoid the errors of the resulting words and POS tags. Instead, the gazette of 1,487 Thai personal name is created from a 900 news article collection. The variable length of character n-gram were trained and matched to the patterns. [30] prepared 15,077 patterns to map the three types of NE tags (DAT, LOC, and PER). Patterns of the NE contexts and clue words are the keys for creating a rule set to extract the NE. The experiment was conducted on a very small set of corpus and the F1-score widely ranged between 68% to 100%.

In the feature-based approach, Winnow [1] was introduced to extract proper noun from Thai text [4]. It used the surrounding words and their POS as the features for Winnow to predict the the POS of the target unknown word with NPRP (proper noun) as its POS. It is assumed that the POS of the word with NE type of PER, LOC and ORG is likely to be NPRP. The authors reported that 92.17% of the test set from 5,000 sentences are correctly annotated. [2] avoided using POS as a feature in extracting NE because of the unreliable result of word segmentation and POS tagging. In stead, a combination of a heuristic rule set with word co-occurrence approach for detecting NE, and maximum entropy model of word features from its orthography and the surrounding context was used to extract PER, LOC, and ORG from a political news corpus of 110,000 words. The F1-score of using plus-minus one word context (87.70%) was higher than plus-minus two words context (79.78%). The comparison results varied according to the type of NE. It was hard to make a conclusion of the suitable features for their approach. [25] investigated SVM in selecting the features among word, POS, word concept, and orthography (types of character) for NER. The experiment was conducted on a collection of 500 articles of Thai business news from Krungthep Turakij news site2. The combination of word, word concept, and orthography features yielded the best F1-score for all PER, ORG, and LOC evaluation with the average of 86.31%. There is no doubt at all for the word concept feature because normally the class of the concept can make a better contribution than others. However, the paper did not discuss how the word concept was assigned. Word sense disambiguation is not trivial in this evaluation. [28] proposed the syllable-segmented input rather than the general word-segmented input to avoid word segmentation errors. The experiment was conducted on BEST2009 corpus3 by using the CRF approach. The size of the corpus was about 80,000 words for training. The study evaluated the effectiveness of CRF model based on syllable-segmented training set against word-segmented training set. As expected, the average F1-score of syllable-segmented model is 80.80%, which is higher than 80.39% of word-segmented model for the test on PER, ORG, and LOC annotation. The features used in word-segmented model are word dic-

2 http://www.bangkokbiznews.com
3 Prepared by National Electronics and Computer Technology Center (NECTEC) for the Thai word segmentation algorithm contest in 2009
tionary, keyword list, and word uni-gram and bi-gram, while the features used in syllable-segmented model are syllable list, and syllable uni-gram and bi-gram. [9] utilized k-character prefix and suffix of a word in addition to its word n-gram and POS n-gram context to train MIRA [6] for PER, ORG, LOC, and DAT annotation. The results showed that k-character prefix and suffix played an important role in the NE annotation task. The overall F1-score was 82.71% when testing on THAI-NEST corpus.

In the deep-learning-based approach, the Variational BiLSTM with CRF (V-BiLSTM-CRF) provided a variational inference-based dropout technique to regularize the model [31]. The experiment was conducted on BEST2010 corpus\(^4\) with 5,238 text files (2,924,433 words) for training and 249 text files (227,302 words) for testing. There were twelve types of NE tags annotated. An 83.7% F1-score can be achieved with POS embedding.

Many particular Thai NE characteristics have been raised and studied. The accuracy of word segmentation and POS tagging are still a big barrier in improving the NER performance. The effective features proposed in the studies can be summed up to a list of clue word, character prefix/suffix, POS, syllable, TCC, character type. These features are introduced in the above three types of approaches. Unfortunately, the reported F1-score are based on various types and sizes of the corpus due the lack of gold standard corpus. It is not fair to compare the approaches by the reported results. However, we conduct the comparison experiment of some recent approaches with the same THAI-NEST and BKD corpus to show the improvement when applied on the revised corpus, and also to show the contribution of TCC embedding to the model.

In this paper, we propose a novel method to improve the quality of the existing THAI-NEST NE corpus by applying the BiLSTM-CNN-CRF method, iteratively. The model is also enhanced by the TCC embedding scheme for generating the character-level representation. In contrast to word and syllable segmentation, TCC is a string unit defined in-between the unit of word and syllable, which can correctly be used to separate a Thai string according to the spelling rules. Handling a Thai string in the TCC manner is reported to perform better than other string units in many experiments, e.g., Thai word boundary segmentation for open compound extraction in [19], and Thai word indexing for information retrieval in [26]. We also prepare to release a large enough standard Thai NE corpus for future study and evaluation.

\(^4\)Prepared by National Electronics and Computer Technology Center (NECTEC) for the Thai word segmentation algorithm contest in 2010

III. THAI NAMED ENTITY CORPUS

The THAI-NEST corpus is a collection of news articles collected from 21 Thai online newspaper publishers from January to December 2009. There are more than 300,000 news articles covering seven major categories of crimes, politics, foreign affairs, sports, education, entertainment, and economy. Table 1 shows the statistics of the THAI-NEST corpus with additional information of the number of characters and TCC. There are more than 7 million words in 83,248 sentences.

| Table 1. Size of the original THAI-NEST corpus |
|-----------------------------------------------|
| File |
| NE type | Sentence | Word | Character | TCC |
| DAT | DAT | 2,979 | 275,668 | 1,063,364 | 574,575 |
| LOC | LOC | 10,094 | 719,594 | 2,783,544 | 1,510,245 |
| MEA | MEA | 2,084 | 201,650 | 756,661 | 419,346 |
| NAM | NAM | 8,541 | 694,922 | 2,681,076 | 1,462,526 |
| ORG | ORG | 23,584 | 1,743,558 | 6,809,757 | 3,726,951 |
| PER | PER | 35,454 | 3,418,018 | 13,063,267 | 7,232,569 |
| TIM | TIM | 512 | 53,795 | 196,924 | 110,058 |
| Total | 83,248 | 7,107,205 | 27,344,592 | 15,036,870 |

The total size of the text collection is large enough for training a model. However, the corpus is manually tagged. Therefore, there is a lot of problems in the tag consistency. The detail is reported in Subsection VI-A. Moreover, a single file is tagged with only one NE type, and the same file is not used to tag with other NE type at all. This means that each type of NE tagging has been exclusively done on each file. This is proper for evaluating the performance of the model for each NE tagging. However, at the end, we need an algorithm to merge the results from each model. Also, the number of NE tags can be increased by conducting cross tagging among the files. The corpus is archived in seven files, and each file is exclusively tagged by each type of NE.

A. NE ANNOTATION SCHEME

The corpus is tagged by BIO (aka IOB2) formatting, namely, the Begin-Inside-Outside tagging format proposed by [16]. It is same as IOB formatting which is proposed by [15] except that the B- tag is used in the beginning of every chunk (i.e. all chunks start with the B- tag). Each type of NE tag is fully expressed in the example, but in general, it is occasionally found that the clue words of each expression are omitted, i.e., “date” in the date expression, “university” in the organization expression, or “Mr.” in the name expression. This can cause some difficulties in capturing the pattern model of the NE tags. Orthography feature does not help in most cases when common nouns are used to name an organization or person.

In Section IV, our approach has shown that CNN-TCC for character-level representation in the character embedding level can capture the NE spelling pattern even though some clue words are omitted, or some parts of the word are wrongly segmented. As a result, the errors of the absence of the clue words and the
wrong-segmented words can be recovered by the TCC embedding.

**IV. TCC BASED THAI NAMED ENTITY RECOGNITION**

We apply the BiLSTM the state-of-the-art NER using the combination of bidirectional LSTM, CNN, and CRF (BiLSTM-CNN-CRF) [13]. According to our baseline THAI-NEST corpus analysis, we found that word segmentation errors seriously affect the next coming tasks such as POS tagging, NER and more. The errors are propagated to cause misjudging in further modeling. Some typical errors are discussed in Subsection VI-A2. However, we are going to improve word segmentation in this paper but we are going to see how we can recover the such errors in the NER task. It is reported in [19] that TCC is a larger chunk of characters which can be unambiguously segmented. It also contains more information about word component comparing to a single character.

Figure 1 shows the architecture of the full combination of BiLSTM with a word vector from W2V for the word-embedding level, CNN encoded TCC for character-level representation, and POS embedding.

The proposed NER model consists of five layers as follows: (i) Word Embedding, (ii) Character-level Representation, (iii) POS Embedding, (iv) BiLSTM layer, and (v) CRF layer. Word, POS, and TCC vectors are concatenated before being fed to the BiLSTM layer. Table 2 shows the hyper-parameters setting for all experiments.

In word embedding layer, we use GloVe 100-dimensional embeddings trained on the corpus 5.7 million words as shown in Table 1. For the comparison between character-based and TCC-based performance, 25-dimensional character and TCC embeddings are prepared to extract the character-level representation of words. In POS embedding layer, we use 47 different POS tags from ORCHID tagset [20] to extract the 100-dimensional POS embeddings. In addition, to reduce model overfitting, we apply the dropout method [21] to regularize our model.

The output vector from BiLSTM is fed into the CRF layer for sequence labeling. The previous word features and labels are included in the CRF model to predict the current word label. CRF is formally defined in Equations (1a) and (1b). \( \mathbf{z} = \{z_1, ..., z_n\} \) is the input sequence, and \( z_i \) is the vector of ith word. \( \mathbf{y} = \{y_1, ..., y_n\} \) is the sequence label for \( \mathbf{z} \). \( \nu(\mathbf{z}) \) is the set of possible sequences for \( \mathbf{z} \). \( \mathbf{W}_{y', y} \) and \( \mathbf{b}_{y', y} \) are the weight and bias vectors corresponding to label pair \( (y', y) \), respectively. CRF is used to determine the weights of different feature functions that maximize the likelihood of the labels in the training data.

\[
p(\mathbf{y}|\mathbf{z}; \mathbf{W}, \mathbf{b}) = \frac{\prod_{i=1}^{n} \psi_i(y_{i-1}, y_i, \mathbf{z})}{\sum_{y' \in \nu(\mathbf{z})} \prod_{i=1}^{n} \psi_i(y'_{i-1}, y_i', \mathbf{z})} \quad (1a)
\]

where

\[
\psi_i(y', y, \mathbf{z}) = exp(\mathbf{W}^{L}_{y', y}z_i + \mathbf{b}_{y', y}) \quad (1b)
\]

In the character-level representation, we apply CNN to encode TCC rather than character because it can present a larger unit of a string. The TCC representation is a non-ambiguously segmentable unit. It is used to capture a larger character pattern to reduce the errors from word segmentation. The result of word segmentation has been improved. Furthermore, to make the character-level representation more meaningful, TCC plays an important role to represent the character unit in a larger pattern. Analogically explaining TCC in English alphabet, in the case of “th” in “think” string, it is represented as a unit of “th” rather “t” and “h”. Therefore, the representation in character-level of the word containing “th” can be distinguished from the one containing “t” or “h”.

| Parameter                     | Setting |
|-------------------------------|---------|
| Character dimension           | 25      |
| Character-level CNN filters   | 30      |
| Character-level CNN window size| 3       |
| Word dimension                | 100     |
| Word LSTM dimension           | 200     |
| Word bidirection              | TRUE    |
| POS dimension                 | 100     |
| Dropout rate                  | 0.5     |
| Batch size                    | 10      |
| Learning rate (initial)       | 0.01    |
| Decay rate                    | 0.5     |
| Gradient clipping             | 5.0     |
| Learning method               | SGD     |
| Training epoch                | 60      |

Table 2. Parameter setting for the model

![FIGURE 1. BiLSTM-CNN-CRF with POS and CNN-TCC for character-level representation](image-url)
Figure 2 shows the convolutional network that encodes TCC in the form of character embedding. In character embedding layer, for the word “ประเทศไทย” (Thailand), TCCs (ת | צ | ו | נ | ר | ל | א | n | ד) are fed into CNN rather than characters (ת | צ | ו | נ | ר | ל | א | n | ד) in general approaches.

![FIGURE 2. CNN encoding TCC for character-level representation of “ประเทศไทย” (Thailand)](image)

The effect of CNN-TCC comparing to CNN-CHAR is shown in Table 3. Word nos. 1, 2, and 3 are a pattern of an organization name and all are correctly segmented and POS tagged. Therefore, both CNN-CHAR and CNN-TCC can annotate the correct ORG tag. The problem occurs when the string is an unregistered word for word segmentation and POS tagging. Word nos. 4 and 5 are a part of the name of the organization but they are wrongly segmented. The character embedding by CNN-CHAR is not enough to capture the pattern of the word orthography comparing to CNN-TCC. The NE for the word with word segmentation error can then be correctly annotated by the CNN-TCC, as shown in the correct ORG annotation in the word nos. 4 and 5. Therefore, CNN-TCC is tolerant of the inputs with word segmentation errors.

| Table 3. Comparison of annotation result from CNN encoding character (CNN-CHAR) and TCC (CNN-TCC) for the sentence (Impact Exhibition Management Co., Ltd. has launched ...) |
|----------------------------------|
| No. | BiLSTM | BiLSTM |
|-----|--------|--------|
| 1   | บริษัท/NCMN/B-ORG | บริษัท/NCMN/B-ORG | company |
| 2   | ไทย/NCMN/L-ORG | ไทย/NCMN/L-ORG | Impact |
| 3   | อิมแพ็ค/NTTL/L-ORG | อิมแพ็ค/NTTL/L-ORG | Exhibition |
| 4   | *อิมแพ็ค/NCMN/O | *อิมแพ็ค/NCMN/O | Management |
| 5   | *อิมแพ็ค/NCMN/O | *อิมแพ็ค/NCMN/O | Limited |
| 6   | ได้/XVAC/O | ได้/XVAC/O | get |
| 7   | เจ้าหน้าที่/VACT/O | เจ้าหน้าที่/VACT/O | open |
| 8   | บริษัท/NCMN/O | บริษัท/NCMN/O | itself |

* Word segmentation error

We investigate the effect of W2V vector embedding comparing to the original word in the word embedding layer. The experiments are conducted under the same environments of the combination of BiLSTM, with POS, with CNN-TCC, and with CRF. The corpus is randomly divided into 80% for training and 20% for testing.

Table 4 shows the best F1-score of 89.22% in the total evaluation when applying the full combination of BiLSTM-POS-CNN-TCC-CRF with the W2V vector for word representation. Adding the features of POS, CNN-TCC, and CRF improves the F1-score in all types of NE tags. Comparing to baseline (BiLSTM), the average F1-score is improved 17.21% from 72.01%, while comparing to the performance of the model generated from the original corpus (before cleaning), the average F1-score is improved 16.21% from 73.01% as shown in Table 5.

| Table 4. Performance of the models on each NE file against the baseline BiLSTM model |
|-----------------------------------------------|-----------------|-----------------|
| NE Type | W2V | POS-CNN-TCC-CRF | POS-CNN-TCC-CRF |
| BiLSTM | BiLSTM | BiLSTM | BiLSTM |
| DET | 89.22 | 89.22 | 89.22 |
| LOC | 72.00 | 72.00 | 72.00 |
| MEA | 69.74 | 69.74 | 69.74 |
| NAM | 65.29 | 65.29 | 65.29 |
| org | 70.41 | 70.41 | 70.41 |
| PRB | 64.68 | 64.68 | 64.68 |
| TM | 79.25 | 79.25 | 79.25 |
| Average | 72.01 | 72.01 | 72.01 |

Table 5 shows an evidence of F1-score improvement (step by step) when adding a new feature. The baseline BiLSTM cannot annotate any NAM at all. When POS is additionally applied, word no. 10, which is an abbreviation, is correctly annotated. This is because of the POS feature showing that NPRP is more likely to be a NAM. CNN-TCC can significantly annotate word nos. 3, 4, and 5 though these words are wrongly segmented. Actually, word nos. 3 and 4 are wrongly segmented. In this context, the word no. 3 must be “นายก” (Chairman) rather than “นาย” (Mr.) This means that TCC provides better spelling information to the word than the character within the word itself, especially to the word no. 5 which is an unregistered word. In the last column, CRF shows the effectiveness of the sequential context in additionally annotating word nos. 6, 7, and 8 because these words frequently occur at the end of NAM phrases. As a result, the consecutive word nos. 3-8 are annotated as a phrase of NAM.

As a result of applying our final model (BiLSTM-POS-CNN-TCC-CRF), the annotation errors between

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
Comparison of model performance evaluation are very small comparing to others. Especially, models 2 and 4 which are reported with are applied on the relatively small sizes of corpora. This is because their rules and corpora cannot be which have been done by using pattern and rule based approaches, because their rules and corpora cannot be applied on the relatively small sizes of corpora.

Table 7 shows the performance of each approach and its evaluation environment. It is hard to compare the performance of our model to the first three models, which have been done by using pattern and rule based approaches, because their rules and corpora cannot be reproduced at all. The corpora used in models 4-6 are personally collected and are not available. Some studies are applied on the relatively small sizes of corpora. Especially, models 2 and 4 which are reported with a very high F1-score but the corpora using in the evaluation are very small comparing to others.

Table 7. Comparison of model performance

| Type | Model Approach | Corpora | Size | NE Type | F-score (%) |
|------|----------------|---------|------|---------|-------------|
| TIM | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| PER | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| ORG | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| MEA | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| LOC | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| NAM | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| ORG | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| MEA | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| LOC | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| NAM | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| ORG | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| MEA | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| LOC | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| NAM | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |

Comparing to the models 7-9, which use the similar size of corpus with a comparable number of NE types, our model significantly outperforms them in terms of F1-score measurement. Model 8 applies MIRA approach to the same THAI-NEST corpus as our model. We can confirm that our model can improve the F1-score by 6.51%. Lastly, our model shows the improvement of the similar approaches using CRF in model 7 by 8.42%, and the combination of V-BiLSTM and CRF in model 9 by 5.52%

The table totally shows that the deep-learning-based approaches still can be improved by using a larger and cleaner corpus as well as the appropriate combination of features.

To compare the performance of the recent models in the feature-based and deep-learning-based approaches, we conduct the evaluation on the same corpus to see how the BKD refined corpus can improve their performance, and to confirm on the contribution of TCC in the character embedding. F1-score has been improved in almost all types of NE when applying on the BKD refined corpus. The detailed sizes of the THAI-NEST original corpus and BKD refined corpus are elaborated in Table 1 and Table 10, respectively. For each type of NE, the corpora are randomly divided into 80% for training and 20% for testing.

Table 8. Model performance on the BKD refined corpus comparing to THAI-NEST original corpus

| Type | Model Approach | Corpora | Size | NE Type | F-score (%) |
|------|----------------|---------|------|---------|-------------|
| TIM | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| PER | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| ORG | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| MEA | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| LOC | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| NAM | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| ORG | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| MEA | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| LOC | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| NAM | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| ORG | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| MEA | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| LOC | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |
| NAM | SVM-BiLSTM-CRF | BKD | 7m | 93.00 | 86.90 |

On average F1-score, our proposed model outperforms the other three models trained in SVM, CRF and V-BiLSTM-CRF based approaches on the same corpus, as shown in Table 8. The results show that the refined corpus has been improved in terms of the annotation consistency. Furthermore, the results of our approach with the advantages of using TCC to mitigate the word segmentation errors can achieve the highest F1-score on the average measure.

VI. ITERATIVE NE TAGGING REFINEMENT

Since the original corpus is disjointly annotated by the NE tags, we train each NE dataset to create a model separately. As a result, we obtain seven models and used them to evaluate the performance, one by one. BiLSTM is our baseline to evaluate any significant improvement of the combination of Word2Vec (W2V) [14] to encode the word-level representation, CNN [11] to encode TCC in character representation, and CRF [10] for including the transition score after the output emission score from BiLSTM. The CRF emission score for each NE tag is used to make decision in the case that there are more than one NE tag given to a particular word in the results merging state.

We apply the full combination of BiLSTM-POS-CNNTCC-CRF with the W2V vector for word representation iteratively to improve the errors in the original corpus. The proposed iterative verification method works as an supervised learning in the manner of adjusting the trained model by the corrected training set resulting from the errors detected when comparing between the
test set and the generated tagged text. It is applied across the seven files of different NE annotations, as shown in Figure 3. We perform repeated holdout method to find the difference between the test set and the generated tagged text. The corpus is divided into two random disjoint subsets i.e. training set and test set. The model is retrained by the corrected training set until there is no difference, or the error is less than a threshold and becomes steady. Most of the errors described in Subsection VI-A are manually removed by comparing the generated result with the original tagged text.

After a certain cycle of retraining the model, the accuracy has been improved as well as the number of proper training set. The total correction of words, POSs, and NE taggings of each file is shown in Table 9, and the total statistics of the refined corpus (BKD) is shown in Table 10. The major errors are from the result of word segmentation (14,527 corrections) which causes the errors in POS (14,693 corrections) and NE tagging (8,121 corrections). Correction of word segmentation errors in PER (9,329 corrections) is highly detected because it contains the highest tags (75,287 as shown in Table 10) comparing to others, and person names are not normally defined in the dictionary.

### TABLE 9. Result of corpus refinement in number of correction

| File | Word | POS | NE Type |
|------|------|-----|---------|
| DAT  | 832  | 3,581 | 2,472   |
| LOC  | 1,242 | 6,027 | 905     |
| MEA  | 962  | 1,784 | 2,416   |
| NAM  | 773  | 759  | 1,461   |
| ORG  | 1,338 | 1,300 | 302     |
| PER  | 9,329 | 1,210 | 557     |
| TIM  | 51   | 32   | 8       |
| Total| 14,693 | 14,693 | 8,121   |

### TABLE 10. Statistics of the refined corpus (BKD)

| File | Sentence | Word | B-x | I-x |
|------|----------|------|-----|-----|
| DAT  | 2,782    | 272,565 | 4,133 | 8,681 |
| LOC  | 8,584    | 709,953 | 21,172 | 8,324 |
| MEA  | 1,967    | 199,305 | 5,205 | 11,142 |
| NAM  | 7,551    | 199,305 | 16,911 | 23,468 |
| ORG  | 20,397   | 1,722,063 | 49,573 | 45,844 |
| PER  | 33,231   | 3,388,885 | 75,287 | 146,789 |
| TIM  | 419      | 53,330  | 495  | 1,048  |
| Total| 74,931   | 7,033,809 | 172,776 | 245,296 |

In terms of word segmentation errors, we generate a list of words for both THAI-NEST and BKD to measure the similarity between them. We calculate the Levenshtein distance between words from the two word lists and convert into a similarity score by normalizing the score by the longer word of the pair. The overall mean of the similarity score is 0.9865. The errors can be found in all types of NE when facing with the ambiguity of expression. For example, “ทะเลสาป” and “ทะเล” are combined to be a correct word of “ทะเลสาป” (lake). An example of proper noun is “ศักข์” (unknown) and “ศักฐาย” (unknown) are combined to be a correct word of “ศักฐาย” (Kowloon District Court).

By observing the difference between the input and the generated NE tagged text in the process of iterative verification shown in Figure 3, we found the following three types of significant errors in the incorrect and inconsistent annotation.

1) Abbreviation tagging errors

Abbreviations are not obvious. In many cases they result in the same form as a common word i.e. “กก” which has the meaning of “a reed” or “to embrace”, while it is also a shortened form of กิโลกรัม (kilogram), กรมบัญชีกลาง (Comptroller General’s Department). Sometimes, they result in a meaningless string if they are not registered. It is difficult for word segmentation to determine the word boundary and POS of these types of strings. The errors can affect the NE tagging of the surrounding words.

2) Word segmentation error

The typical error in word segmentation can be found in the string of “นาย ” or “นาย ” as shown in the result of นาย/NTTL or นาย/NCMN. This kind of error frequently occurs when the string contains a part of abbreviation, proper noun, or out-of-vocabulary word (OOV).

3) NE and POS annotation error

Most of the cases are from POS tagging errors. Normally, the digits in the date expression must be tagged as a DONM (determiner, ordinal number expression). With inconsistency in POS tagging, digits are sometimes
tagged as NCNM (cardinal number) and DCNM (determiner, cardinal number expression). The NE model then cannot capture the pattern of the date expression.

4) Annotation error
It is reported that the corpus is manually revised but some pairs of NE tags can confuse the annotators in making decisions. We found that there are many errors in the confusion cases between PER and common word, ORG and LOC, and NAM and ORG.

The tags of PER, ORG and LOC are actually confusing because they are, for one reason, named by using a proper name which is often an OOV (out-of-vocabulary word), unknown to the word segmentation. For example, the name of "Washington" which is correctly tagged as "Washington/NPRP/B-LOC", but it is segmented and tagged as "Washington/NCMN/B-LOC (palanquin)", "Washington/NCMN/I-LOC (fight for)", and "Washington/VATT/I-LOC (stuck)". For another reason, they can be found in a compound word or phrase expression, which can be composed of common words. For example, the name of a political party "ภูมิใจไทย" (Bhumjaithai) which is correctly tagged as "ภูมิใจไทย/NPRP/B-ORG", but it is segmented and tagged as "ภูมิใจ/VSTA/B-ORG (proud of)", and "ภูมิ/NPRP/I-ORG (Thai)". In case of PER, it is found in many examples, such as "ยุทธมรรคม (Yubamrung)" which is correctly tagged as "ยุทธมรรคม/NPRP/B-PER", but it is segmented and tagged as "ยุทธ/VSTA/O (stay)", and "มรรคม/VACT/O (maintain)".

Once the word segmentation and POS tagging are corrected, we do not pass the input string to the word segmentation again because it will produce the same errors, and we do not want to make any changes in word segmentation. Instead, the word segmented with POS tagged string is passed directly to the NER module, and the new words together with the tags are registered.

The correction in the tags of DAT, MEA, NAM, and TIM are relatively low. The expression of DAT and TIM is quite straightforward with a common format and a closed set of name of the months. The errors in NAM are similar to the case of PER, ORG, and LOC because of the way of naming the entities.

The errors in MEA are an interesting case for the Thai language because of the expression of classifier. There is a particular set of classifiers which is designated by the head noun of the noun phrase. For example, "รถ (car)" always takes a classifier of "คัน (classifier of car)". However, the difficulty occurs in the case of words having itself as the classifier. For example, the classifier for "คน (person)" is the same as itself "คน (classifier of person)". So, the expression of "I person" is "คน 1 คน". Some example of errors found in the case of MEA are, "1 วัน (1 day)" is tagged as "1/DCNM/O วัน/NCMN/O (day)" instead of "1/DCNM/B-MEA วัน/CMTR/I-MEA (classifier of day)", or "90 คน (90 persons)" is tagged instead of "90/DCNM/B-MEA คน/CMTR/I-MEA (classifier of person)".

VII. CONCLUSION
Since there are many difficulties in the Thai language, it is very cost-intensive to prepare a high-quality consistently-annotated corpus. Our proposed methods are effective in detecting the errors which occur in word segmentation, POS tagging and the inconsistently NE tagging processes. CNN encoding of TCC for the character-level representation can also recover the word segmentation and POS tagging errors in many cases. Following by our proposed NE tagging method (BiLSTM-POS-CNN-TCC-CRF) which can achieve 89.22% in F1-score measure when trained on the refined corpus, comparing to 74.96% of the baseline model BiLSTM, 73.01% of the same model but trained on the original noisy corpus, and 87.77% of the same model with CNN-CHAR embedding. Iteratively, the corpus is refined and used to re-trained the model. As a result, the performance of the NE tagging is improved along with having the NE tags in the corpus refined. This means that our iterative NE tagging refinement method is effective in constructing a silver standard NE corpus. The proposed iterative NE tagging refinement method is general. This can benefit corpus development, especially for low-resource languages. The BKD corpus is a result of the refinement of an existing noisy corpus. It is a silver standard NE corpus which is available for NER model training and evaluation.

VIII. ACKNOWLEDGMENT
This work is supported by the Thailand Science Research and Innovation Fundamental Fund. The authors gratefully acknowledge the financial support provided by Thammasat University Research fund under the TSRI, Contract No. TUFF19/2564 and TUFF24/2565.

REFERENCES
[1] A. Blum, “Empirical Support for Winnow and Weighted-Majority Algorithms,” Results on a Calendar Scheduling Domain, Machine Learning, vol. 26, no. 1, pp. 5-23, 1997.
[2] H. Chanlekha, and A. Kawktrakul, “Thai Named Entity Extraction by incorporating Maximum Entropy Model with Simple Heuristic Information,” in Proc. IJCNLP-2004, 2004, pp. 49–55.
[3] H. Chanlekha, A. Kawktrakul, P. Varasrai, and I. Mulassas, “Statistical and Heuristic Rule Based Model for Thai Named Entity Recognition,” in Proc. SNLP-Oriental COCOSDA, 2002.
[4] P. Charenpornwatt, B. Kijsiiruk, and S. Meknavin, “Feature-based Proper Noun Identification in Thai,” in Proc. NCSEC98, 1998.
[5] N. Chinchor, and B. Sundheim, “Message Understanding Conference (MUC) 6,” in Proc. LDC2003T13, Web Download, Philadelphia: Linguistic Data Consortium, 2003.
[6] K. Crammer, R. McDonald, abd F. Pereira, “Scalable Large-Margin Online Learning for Structured Classification,” in Proc. NIPS Workshop on Learning With Structured Outputs, 2005.
[7] Y. Hahm, J. Park, K. Lim, Y. Kim, D. Hwang, and K. S. Choi, “Named Entity Corpus Construction using Wikipedia and DBpedia Ontology,” in Proc. LREC-2014, 2014, pp. 2565–2569.
Virach Sornlertlamvanich: Preparation of Papers for IEEE TRANSACTIONS and JOURNALS

[8] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, “OntoNotes: the 90% Solution,” in Proc. Human Language Technology Conference of the NAACL, Companion Volume: Short Papers, Association for Computational Linguistics, 2006, pp. 57–60.

[9] C. Kruengkrai, V. Sornlertlamvanich, W. Buranasing, and T. Charoenporn, “Semantic Relation Extraction from a Cultural Database,” in Proc. SANLP, COLING2012, 2012, pp. 15–24.

[10] J. Laflerty, A. McCallum, and F. CN. Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data,” in Proc. ICML-2001, vol. 951, 2001, pp. 282–289.

[11] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and D. L. Jackel, “Backpropagation Applied to Handwriten Zip Code Recognition,” Neural Computation, vol. 1, no. 4, pp. 541–551, 1989.

[12] J. Li, A. Sun, J. Han, and C. Li, “A Survey on Deep Learning for Named Entity Recognition,” The Journal of CoRR, abs/1812.09449, 2018.

[13] X. Ma, and E. Hovy, “End-to-end Sequence Labeling via Bidirectional LSTM-CNNs-CRF,” in Proc. The Annual Meeting of the Association for Computational Linguistics, 2016, pp. 1064–1074.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and Their Compositionality,” in Proc. The Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[15] L. A. Ramshaw and M. P. Marcus, “Text Chunking using Transformation-Based Learning,” in Proc. The Third Workshop on Very Large Corpora, Association for Computational Linguistics, 1995, pp. 82–94.

[16] T. K. Sang, “Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition,” in Proc. CoNLL-2002, 2002, pp. 155–158.

[17] V. Sornlertlamvanich, “Word Segmentation for Thai in Machine Translation System, in Machine Translation, National Electronics and Computer Technology Center, Bangkok, (in Thai), 1993, pp. 50–56.

[18] V. Sornlertlamvanich, and H. Tanaka, “The Automatic Extraction of Open Compounds from Text Corpus,” in Proc. COLING-96, 1996, pp. 1143–1146.

[19] V. Sornlertlamvanich, and H. Tanaka, “Extracting Open Compounds from Text Corpus,” in Proc. The Second Annual Meetings of the Association for Natural Language Processing, 1996, pp. 213–216.

[20] V. Sornlertlamvanich, N. Takahashi, and H. Isahara, “Building a Thai Part-Of-Speech Tagged Corpus (ORCHID),” The Journal of the Acoustical Society of Japan (E), vol. 20, no. 3, 1999, pp. 189–198.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, T. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” The Journal of Machine Learning Research, vol. 15, no. 1, 2014, pp. 1929–1958.

[22] K. Suriyachay, T. Charoenporn, and V. Sornlertlamvanich, “Thai Named Entity Tagged Corpus Annotation Scheme and Self Verification,” in Proc. LRTC2019, 2019, pp. 131–137.

[23] K. Suriyachay, T. Charoenporn, V. Sornlertlamvanich, and N. Kaanthanaphong, “Enhancement of Character-Level Representation in Bi-LSTM model for Thai NER,” Journal of Science & Technology Asia (STA), vol. 26, no. 2, Thammasat University, 2021, pp. 61–78.

[24] P. Suwanno, Y. Suzuki, and H. Yamazaki, “Selecting the Optimal Feature Sets for Thai Named Entity Extraction,” in Proc. ICEE-2007 & PEC, vol. 5, 2007.

[25] T. Theeramunkong, V. Sornlertlamvanich, T. Tanhermhong, and W. Chinnan, “Character Cluster Based Thai Information Retrieval,” in Proc. The Fourth Symposium on Natural Language Processing, 2000.

[26] T. Theeramunkong, M. Boriboon, C. Haruechalyasak, N. Kittipattanahawon, K. Kosawat, C. Onsuwan, I. Siriwat, T. Suwanapong, and N. Tongtep, “THAI-NEST: A Framework for Thai Named Entity Tagging Specification and Tools,” in Proc. The International Conference on Corpus Linguistics (CILC2010), 2010, pp. 895–908.

[27] N. Tirarosoj and W. Aroonmanakun, “Thai Named Entity Recognition Based on Conditional Random Fields,” in Proc. The Eighth International Symposium on Natural Language Processing, 2009, pp. 216–220.

[28] K. S. Tjong, E. Erik, and F. De Meulder, “Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition,” in Proc. The Seventh Conference on Natural Language Learning, HLT-NAACL, 2003, pp. 142–147.

[29] N. Tongtep, and T. Theeramunkong, “Pattern-based Extraction of Named Entities in Thai News Documents,” Science & Technology Asia, vol. 15, no. 1, 2010, pp. 70–81.

[30] C. Udomcharoenchaikit, P. Vateekul, and P. Boonkwan, “Thai Named-Entity Recognition Using Variational Long Short-Term Memory with Conditional Random Field,” in Proc. The Joint International Symposium on Artificial Intelligence and Natural Language Processing (ISAI-NLP 2017), 2017, pp. 288–298.

[31] C. Udomcharoenchaikit, P. Vateekul, and P. Boonkwan, “Thai Named-Entity Recognition Using Variational Long Short-Term Memory with Conditional Random Field,” in Proc. The Joint International Symposium on Artificial Intelligence and Natural Language Processing (ISAI-NLP 2017), 2017, pp. 288–298.
SUMETH YUENYONG received his B.Eng. in Electrical Engineering and M.Eng in Information and Communication Technology for Embedded Systems, from Sirindhorn International Institute of Technology (SIIT) in 2004 and 2010, respectively. He received the Japanese Government Scholarship (Monbukagakusho) to study in Japan, and obtained M.Eng and D.Eng in Communication and Integrated Systems from Tokyo Institute of Technology in 2011 and 2014, respectively. Currently he is with the Department of Computer Engineering, Faculty of Engineering, Mahidol University. He received a New Faculty Member grant from the Thailand Research Fund in 2016. His research interests include signal and image processing, deep learning and embedded systems.