Thai Legal Term Correction using Random Forests with Outside-the-sentence Features

Takahiro Yamakoshi[†], **Vee Satayamas**[‡], **Hutchatai Chanlekha**[‡]
Yasuhiro Ogawa[†], **Takahiro Komamizu**[†], **Asanee Kawtrakul**[‡], **Katsuhiko Toyama**[†]
[†] Nagoya University, Japan
{yamakoshi,ogawa,komamizu,toyama}@kl.itc.nagoya-u.ac.jp
[‡] Kasetsart University, Thailand
{vee.sa,hutchatai.c,ak}@ku.th

Abstract

We propose a method for finding and correcting misused Thai legal terms in Thai statutory sentences. Our method predicts legal terms using Random Forest classifiers, each of which is optimized for each set of similar legal terms. Each classifier utilizes outside-the-sentence features, namely, promulgation year, title keywords, and section keywords of statutes, in addition to words adjacent to the targeted legal term. Our experiment shows that our method outperformed not only a Random Forest method without the outside-the-sentence features, but also BERT (Bidirectional Encoder Representations from Transformers), a powerful language representation model, in overall accuracy.

1 Introduction

Legislation drafting requires careful scrutiny. An important consideration is the appropriate use of legal terms. In Thai legislation, allowable usage of similar legal terms is described in the legislation manual from the Office of the Council of State, the bill examining authority. For example, there are two similar Thai legal terms *yang-nueng-yang-dai* (อย่างหนึ่งอย่างใด; lit. thing-one-thing-any) and *yang-dai-yang-nueng* (อย่างใดอย่างหนึ่ง; lit. thing-any-thing-one) separately used in Thai statutory sentences. Both terms are used to choose entities from a given set, like "some of the following items." However, according to the legislation manual, *yang-nueng-yang-dai* is used only when one can choose one or more entities, while *yang-dai-yang-nueng* is used only when only one entity can be chosen (Office of the Council of State, 2008). Drafters must not misuse any legal term in a bill; otherwise the bill can have unintended provisions, and thus unintentionally and incorrectly govern the people. Therefore, drafters need to scan the hundreds of pages of the bill thoroughly to locate misused legal terms and correct them; however, scanning is currently done by humans, which requires an enormous amount of time and is subject to human error.

We therefore propose a legal term correction method for Thai statutory sentences that assists drafters in finding misused legal terms in a draft and offers corrections. Inspired by Yamakoshi et al. (2018)'s idea, we handle legal term correction as a special case of the multiple-choice sentence completion test by regarding a set of similar legal terms as a set of choices. Also, we adopt Random Forest classifiers (Breiman, 2001) to score the likelihood of each candidate for the legal term. Here, we introduce additional features from outside of the statutory sentence, namely, year, title keyword, and section keyword. We expect that the year feature copes with changes in legal term usage over time, the title keyword feature captures the difference in legal term usage by statute type, and the section keyword feature adequately predicts a legal term in an item with few adjacent words.

The contributions of our paper are as follows: (1) we apply a legal term correction method that successfully completes the Japanese legal term correction task and confirm the effectiveness of this method for statutory sentences in other legislation

33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33), pages 279-287, Hakodate, Japan, September 13-15, 2019

systems, (2) we design three additional features, one a temporal feature and the others topical features, and (3) we examine the extended legal term correction method with the original method and a modern method that is based on BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018), and we demonstrate that the extended method outperforms the others.

In Section 2, we introduce several sets of Thai legal terms regulated in the legislation manual. In Section 3, we survey related work. In Section 4, we show our method. In Sections 5 and 6, we present our evaluation experiment and discuss the results, respectively. Finally, we summarize and conclude our paper in Section 7.

2 Thai Legal Terms

In this section, we explain several sets of Thai legal terms whose usage is defined in the Thai legislation manual (Office of the Council of State, 2008).

2.1 *Yang-nueng-yang-dai* and *yang-dai-yang-nueng*

Yang-nueng-yang-dai (อย่างหนึ่งอย่างใด) is literally "thing-one-thing-any," while *yang-dai-yang-nueng* (อย่างใดอย่างหนึ่ง) is "thing-any-thing-one," so they look very similar. In Thai statutory sentences, these terms are used in choosing entities from a particular set. *Yang-nueng-yang-dai* is used when one or more entities can be selected simultaneously. On the other hand, *yang-dai-yang-nueng* is used when only one entity can be selected.

Yang can be substituted for other words such as *khon* (คน; person), so we can use *khon-nueng-khon-dai* (คนหนึ่งคนใด; one or more people) or *khon-dai-khon-nueng* (คนใดคนหนึ่ง; only one person).

2.2 *Amnat-nathi*, *amnat-lae-nathi*, and *nathi-lae-amnat*

Amnat-nathi (อำนาจหน้าที่), *amnat-lae-nathi* (อำนาจและหน้าที่), and *nathi-lae-amnat* (หน้าที่และอำนาจ) consist of *amnat* (อำนาจ; power), *nathi* (หน้าที่; duty), and *lae* (และ; and). *Amnat-nathi* is now considered a compound word, while *amnat-lae-nathi* and *nathi-lae-amnat* are noun phrases. According to a Thai law dictionary, *amnat-nathi* means cognizance or competence (Tipchod and KhotchaSeni, 2013). Although still a matter of discussion, *amnat-nathi*, *amnat-lae-nathi*, and *nathi-lae-amnat* have the following usages: *amnat-nathi* means the power to perform duties; *amnat-lae-nathi* is just a combination of two words, "power" and "duty," and is used when both powers and duties are defined in the statute; and *nathi-lae-amnat* is the concept that one must have duties before having power. It is important to note that the appearance of *amnat-lae-nathi* is recent and that the constitution of Thailand has used only *amnat-nathi*.

2.3 *Panakngan-chaonathi* and *chaonathi*

Both *Panakngan-chaonathi* (พนักงานเจ้าหน้าที่; competent authority (Tipchod and KhotchaSeni, 2013)) and *chaonathi* เจ้าหน้าที่; officer) mean a person who has the power to practice a legal action. However, these terms are used for different kinds of people. The former is used for a person appointed by a minister, while the latter is used more generally.

2.4 *Kharachakan-kanmueang* and *phu-damrong-tamnaeng-thang-kanmueang*

Both *kharachakan-kanmueang* (ข้าราชการการเมือง, lit. official-politics) and *Phu-damrong-tamnaeng-thang-kanmueang* (ผู้ดำรงตำแหน่งทางการเมือง, lit. person-preserve-position-in-politics) mean a certain kind of public servant, but each has a different scope of meaning. The former is predominately used for a minister or their aide. The latter can indicate not only a person of *kharachakan-kanmueang*, but also a national assembly member, the mayor of Bangkok, a city council member, and so on.

3 Related Work

In this section, we survey related work on the legal term correction task. First, we describe the definition of the legal term correction task given by Yamakoshi et al. (2018), and then explain technologies that can be used to solve this task.

3.1 Legal Term Correction

Yamakoshi et al. defined the legal term correction task as follows (Yamakoshi et al., 2018):

Algorithm 1 Algorithm for legal term correction

Input: W, T

Output: Suggests

```
Suggests ← ∅
for all (i, j) such that w_i w_{i+1} ··· w_j = t ∈ T do
    W^ℓ ← w_1 w_2 ··· w_{i−1}
    W^r ← w_{j+1} w_{j+2} ··· w_{|W|}
    t_best ← arg max_{t'∈T} score(W^ℓ, t', W^r)
    if t ≠ t_best then
        Suggests ← Suggests ∪ {suggestion that t in position (i, j) should be replaced into t_best}
    end if
end for
```

- A statutory sentence $W = w_1\ w_2 \cdots w_{|W|}$ and a set of legal terms $T \subseteq V^+$ are given, where V^+ is the Kleene plus of vocabulary V; that is, legal term $t \in T$ can be either a word or multiple words;
- The adequacy of each legal term t found in W is judged;
- If another legal term $t_{\text{best}} \in T$ $(t_{\text{best}} \neq t)$ seems more adequate in the context, t_{best} is suggested as a replacement for t.

They also defined a general algorithm: Algorithm 1, where $\text{score}(W^\ell, t, W^r)$ is a scoring function that calculates the likelihood of term t when two word sequences W^ℓ and W^r are adjacent to the left and right of t, respectively.

This problem can be regarded as a special case of the sentence completion test by introducing the following ideas:

- W^ℓ ___ W^r is a sentence with a blank, where ___ is the blank, and W^ℓ and W^r are as defined in Algorithm 1.
- T is the choices, one of which adequately fills the blank in the sentence.

However, Yamakoshi et al. pointed out that this problem differs from the general multiple-choice sentence completion test in two ways. First, a set of choices (i.e., a legal term set) relates to many sentences with blanks. In contrast, we cannot assume that such a large number of sentences relate to a set of choices in the general multiple-choice sentence completion test, since we usually consider that each sentence with a blank has a different set of choices.

Second, we can consider only meaningful legal term sets mentioned by the legislation manuals. In contrast, we may consider any combination of choices in the general multiple-choice sentence completion test, since they are unrestricted.

3.2 Technologies for Solving the Legal Term Correction Task

In this section, we introduce some technologies for the scoring function of the legal term correction task. We use Random Forest (Breiman, 2001) as the scoring function. We describe Random Forest in Section 3.2.1. In the context of the sentence completion test, BERT (Devlin et al., 2018), a powerful language representation model, can be used as the scoring function. We briefly explain BERT in Section 3.2.2. Finally, in Section 3.2.3, we describe language models whose performance is traditionally evaluated by a sentence completion test.

3.2.1 Random Forest

Random Forest (Breiman, 2001) is a machine-learning algorithm for classification. Figure 1 explains the training and prediction processes of a Random Forest classifier.

It learns the training data by building a set of decision trees. A decision tree is conceptually a suite of if-then rules like the ones in the middle of Figure 1. After learning, the Random Forest classifier predicts the class of the given data by taking a vote on each decision tree. Here, each decision tree is constructed by randomly selected data records and features. Therefore, even if a single decision tree makes an unsophisticated decision, the ensemble of decision trees is better at predicting unseen data.

Yamakoshi et al. (2018) utilized Random Forest classifiers specialized for each legal term set as the scoring function in Algorithm 1. The following equation denotes the scoring function:

$$\begin{aligned} &\text{score}(W^\ell, t, W^r) \qquad (1)\\ &= \sum_{d \in D} P_d(t | w^\ell_{|W^\ell|-N+1}, \ldots, w^\ell_{|W^\ell|}, w^r_1, \ldots, w^r_N), \end{aligned}$$

where D is a set of decision trees, d is a decision tree, and $P_d(t|w_1, w_2, \ldots, w_N)$ is the probability (ac-

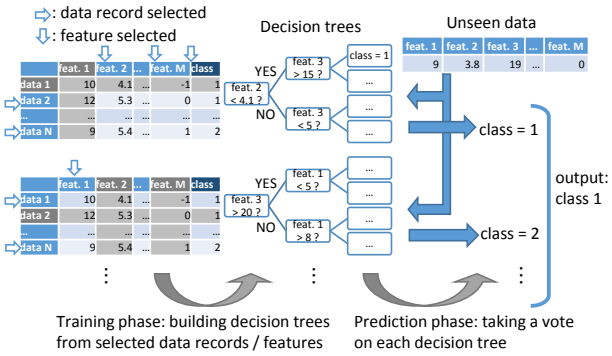

Figure 1: Processes of Random Forest

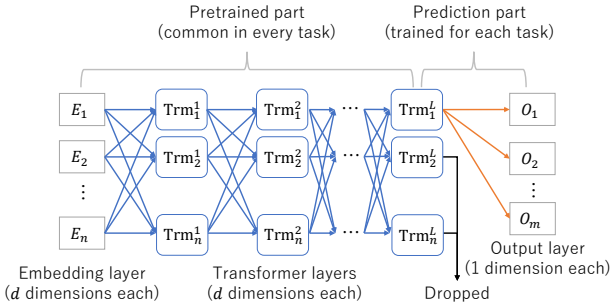

Figure 2: BERT model

tually 0 or 1) that d chooses t based on features $w_1, w_2, \ldots, w_N$. w_i^ℓ and w_i^r are the i-th word of W^ℓ and W^r, respectively. N is the window size (the number of left or right adjacent words focused on). If $|W^\ell| < N$, W^ℓ will be padded with out-of-sentence tokens (same in $|W^r|$).

After training, a Random Forest classifier outputs feature importance that indicates how much each feature contributes to a good prediction. Feature importance is calculated using out-of-bag examples (not sampled examples). The importance of the n-th feature is calculated by the following procedure:

1. Build the m-th decision tree T_m using randomly sampled examples;
2. Acquire the set of out-of-bag examples on decision tree $E_{m,0}$;
3. Make a set of examples $E_{m,n}$, where the n-th feature of each example is randomly shuffled;
4. Predict classes $C_{m,0}$ and $C_{m,n}$ of each example in $E_{m,0}$ and $E_{m,n}$, respectively, using T_m;
5. Calculate the increase of misprediction rate $C_{m,0}$ and $C_{m,n}$;
6. Calculate the total increase of misprediction rate x_n by applying 1. to 5. for every decision tree.
7. If x_n is high, the n-th feature is important because shuffling the feature brought about an inaccurate prediction.

3.2.2 BERT

Devlin et al. (2018) introduced a new language representation model called BERT (Bidirectional Encoder Representations from Transformers). This model is designed for a wide range of NLP tasks such as question answering and language inference. Figure 2 shows the construction of a BERT model. The BERT model in the figure inputs n words and outputs a probability distribution of m classes. In the legal term correction task, each output value denotes the probability of a certain legal term and will be the value of the scoring function in Algorithm 1.

A BERT model is a neural network model that consists of two parts: a pretrained part and a prediction part. The pretrained part consists of an embedding layer and Transformer layers, where each layer's unit is d-dimensional. Transformer (Vaswani et al., 2017) is a neural network model made of multi-head attention units and feedforward connections. The prediction part consists of the final Transformer layer and an output layer connected by feedforward connections. In a sentence-level classification task, only one Transformer unit connects with the output layer and other units are dropped.

When making a classification model for a particular task, we can inherit parameters of the pretrained part from a pretrained common model trained with a large-scale diversified corpus and fine-tune the whole model with a task-specific dataset. Using the pretrained common model, we can get quite high performance for various kinds of tasks with a small amount of training.

3.2.3 Language Model

A language model assigns a likelihood to each word sequence $W = w_1\ w_2 \ldots w_{|W|}$. In the legal term correction task, the model works as a scoring function in Algorithm 1 that outputs the likelihood of a sentence whose blank is filled with a legal term.

Here, each word w_i that constitutes W is chosen from a vocabulary that a language model defines. Therefore, a language model can solve any question of the sentence completion test if each word in a sentence with a blank and a set of choices is in the vocabulary.

To evaluate language models, Zweig and Burges (2011) presented a dataset of the multiple-choice sentence completion test called the MSR Sentence Completion Challenge Data.

A variety of language models are evaluated by this dataset. First, Zweig and Burges (2011) evaluated n-gram models with their dataset. The most powerful language models evaluated by this dataset have neural network architectures. For instance, Mikolov et al. (2013) proposed two neural language models: the Continuous Bag-of-Words Model (CBOW) and Continuous Skip-gram Model (Skipgram). Mnih and Kavukcuoglu (2013) proposed the vector Log-bilinear model (vLBL) and ivLBL. Mori et al. (2015) proposed vLBL(c) and vLBL+vLBL(c), which are improved models of vLBL that are sensitive to the relative positions of words adjacent to the target word.

4 Proposed Method

In this section, we show our proposed method for the legal term correction task. Our method is based on Yamakoshi et al. (2018)'s model that uses Random Forest as a scoring function. Unlike their method, our method introduces three additional features from outside of the sentence to augment prediction performance. We describe these features in Section 4.1, followed by our prediction model in Section 4.2.

4.1 Out-of-sentence Features

We introduce three additional features, namely, year, title keyword, and section keyword to our method. We describe these features and intentions below.

- **Year Feature**
  The year feature denotes the year when the statute was promulgated. We use this feature as a one-dimensional integer variable and introduce it to deal with changes in term usage over time. For example, *amnat-lae-nathi* has appeared recently; therefore, a prediction model with this feature should know that this legal term does not appear in older statutes.

1	มาตรา ๒๗ ผู้มีลักษณะอย่างใดอย่างหนึ่งดังต่อไปนี้ ต้องห้ามมิให้เป็นประธานกรรมการหรือกรรมการ คือ
2	(๑) มีส่วนได้เสียในสัญญากับการรถไฟแห่งประเทศไทย หรือในกิจการที่กระทำให้แก่การรถไฟแห่งประเทศไทย ทั้งนี้ ไม่ว่าโดยตรงหรือโดยทางอ้อม เว้นแต่จะเป็นเพียงผู้ถือหุ้นของบริษัทที่กระทำการอันมีส่วนได้เสียเช่นว่านั้น
3	(๒) เป็นพนักงานของการรถไฟแห่งประเทศไทย
4	(๓) เป็นข้าราชการการเมือง
5	(๔) ขาดคุณสมบัติหรือมีลักษณะต้องห้ามตามกฎหมายว่าด้วยคุณสมบัติมาตรฐาน สำหรับกรรมการ และ พนักงานรัฐวิสาหกิจ

Figure 3: A legal term (underlined) with few adjacent words

- **Title Keyword Feature**
  The title keyword feature denotes the keywords of the statute's title. We use this feature as a n-dimensional boolean variable, where n is the number of keywords defined, as each of its elements represents the existence of a certain keyword. We assume that the use of legal terms slightly differs by statute type. One example is that the constitution of Thailand has used only *amnat-nathi* and has not used *amnat-lae-nathi* or *nathi-lae-amnat*.

- **Section Keyword Feature**
  The section keyword feature denotes keywords of the section to which the statutory sentence belongs. As with the title keyword feature, we use this feature as a n-dimensional boolean variable. We introduce this feature to cope with legal terms having only a few adjacent words. Figure 3 demonstrates an example. In the case of Figure 3, *kharachakan-kanmueang* (ข้าราชการการเมือง) in line 4 is a legal term to be predicted. However, only the word เป็น (*pen*; being) is given as a meaningful feature if we use only adjacent words in the sentence as features. Therefore, we use the section keywords as additional features to solve this problem. In this case, the sentence in line 1 is the section (มาตรา; matra), so that keywords of this sentence are used as the section keyword feature.

Algorithm 2 Our algorithm

Input: W, y, K^t, K^s, T
Output: Suggests

```
Suggests ← ∅
for all (i, j) such that w_i w_{i+1} ··· w_j = t ∈ T do
    W^ℓ ← w_1 w_2 ··· w_{i−1}
    W^r ← w_{j+1} w_{j+2} ··· w_{|W|}
    t_best ← arg max_{t'∈T} score(W^ℓ, t', W^r, y, K^t, K^s)
    if t ≠ t_best then
        Suggests ← Suggests ∪ {suggestion that t in position (i, j) should be replaced into t_best}
    end if
end for
```

4.2 Prediction Model

Because we use the additional features to predict legal terms, we slightly modify the legal term correction task as follows:

- Statutory sentence $W = w_1\ w_2 \cdots w_{|W|}$, year feature y, title keyword feature K^t, section keyword feature K^s, and a set of legal terms $T \subseteq V^+$ are given, where K^t and K^s are a subset of vocabulary V;
- The adequacy of each legal term t found in W is judged;
- If another legal term $t_{\text{best}} \in T$ $(t_{\text{best}} \neq t)$ seems more adequate in the context, t_{best} is suggested to replace t.

Algorithm 2 is a general algorithm for this problem, where the input and scoring function are modified.

We utilize Random Forest as the scoring function $\text{score}(W^\ell, t', W^r, y, K^t, K^s)$, which is calculated by the following equation:

$$\begin{aligned}&\text{score}(W^\ell, t, W^r, y, K^t, K^s)\\ &= \sum_{d \in D} P_d(t | w^\ell_{|W^\ell|-N+1}, \ldots, w^\ell_{|W^\ell|}, w^r_1, \ldots, w^r_N,\\ &\quad y, k^t_1, \ldots, k^t_{|K^t|}, k^s_1, \ldots, k^s_{|K^s|}) = \sum_{d \in D} P_d(t|F), \quad (2)\end{aligned}$$

where D is a set of decision trees, d is a decision tree, and $P_d(t|F)$ is the probability that d chooses t based on the features F. Here, w^ℓ_i and w^r_i are the i-th words of W^ℓ and W^r, respectively. y is the year feature, and

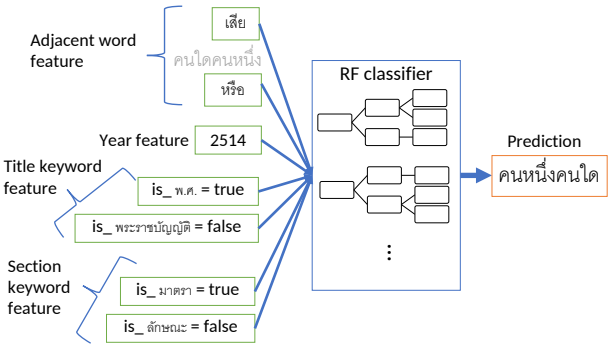

Figure 4: Our model

k^t_i and k^s_i are the existence of the i-th keyword in the title sentence and section sentence, respectively. N is the window size. Figure 4 expresses the input and output of this model.

5 Experiment

To evaluate the effectiveness of our method, we conducted an experiment on predicting legal terms in Thai statutory sentences.

5.1 Outline of Experiment

We compiled a statutory sentence corpus from the website of the Office of the Council of State [1]. We acquired 7,399 Thai statutes that include constitutions, codes, emergency decrees, royal decrees, ordinances, regulations, orders, notices, and more. There are 7,516,792 tokens and 66,671 different words in the corpus after tokenization by PyThaiNLP (v.1.7) [2]. We created the dataset using the following procedure: (1) extract all sentences where more than one legal term appears; (2) unify the sentences so that there are no identical sentences in the dataset; (3) make datasets for each legal term by grouping sentences based on the legal terms contained within; (4) split each dataset into five for five-fold cross validation; then (5) process each sentence to an example for each method.

We defined five legal term sets by referencing the Thai legislation manual (Office of the Council of State, 2008). Table 1 shows each legal term and its number of total occurrences.

[1] http://www.krisdika.go.th/
[2] https://github.com/PyThaiNLP/pythainlp

Table 1: Legal terms

Term set	Legal Term	Counts
Set1-1	*yang-nueng-yang-dai*	1,469
	yang-dai-yang-nueng	1,152
Set1-2	*khon-dai-khon-nueng*	489
	khon-nueng-khon-dai	268
Set2	*amnat-nathi*	5,631
	amnat-lae-nathi	977
	nathi-lae-amnat	519
Set3	*panakngan-chaonathi*	8,006
	chaonathi	4,579
Set4	*kharachakan-kanmueang*	595
	phu-damrong-tamnaeng -thang-kanmueang	411
Total		24,096

We compared our method (Random Forest with additional features; RF+) with Yamakoshi et al. (2018)'s Random Forest (RF) and BERT (Devlin et al., 2018). As a baseline, we also tried maximum likelihood estimation (MLE), which always selects the most frequent legal terms in the training data. For evaluation, we averaged the accuracies of each legal term set in the five datasets.

For the Random Forest methods, we set hyper-parameters as follows: the estimator number is 500; the maximum depth of a decision tree is unlimited; and the window size is 15. We tokenized each sentence by PyThaiNLP (v.1.7). Implementation, training, and testing are done by Scikit-learn (v.0.19.1).

For RF+, we used the most frequent 1,000 words in titles and sections as the keywords of the title and section, respectively. Here, we excluded some functional words using the stopword vocabulary in PyThaiNLP (v.1.7). We also excluded legal terms from the section keywords to prevent them from becoming clues to predict the legal term.

For BERT, we used the *BERT-Base, Multilingual Cased model* [3] that is offered by the authors of the paper (Devlin et al., 2018). The pretrained model has 12 Transformer layers and each layer's unit contains 768 hidden values. We replaced the target legal term in every example into a meta token "^" that is not used in the corpus, so that the model will predict the legal term based on the context around the token. The model accepts a sequence of a maximum 128 subwords and almost all subwords defined in its vocabulary consist of one character. Therefore, we truncated each example so that one example has at most 128 characters. Other hyper-parameters are as follows: the number of epochs is 20; batch size is 32; learning rate is 2e-5; and warmup proportion is 0.1. Implementation, training, and testing were done by Tensorflow on Colaboratory [4].

Table 2: Experimental results

Term set	MLE	BERT	RF	RF+
Set1-1	56.0%	85.4%	83.8%	**86.6%**
Set1-2	64.6%	**93.4%**	90.2%	91.8%
Set2	79.0%	85.5%	84.6%	**89.4%**
Set3	63.6%	**95.2%**	89.3%	94.4%
Set4	59.1%	**95.1%**	89.0%	93.4%
Average	67.2%	91.2%	87.3%	**92.0%**

5.2 Experimental Results

Table 2 shows the experimental results of each model. RF+ acheived the best accuracy in Set2, Set4-1, and overall accuracy. In every legal term set, RF+ achieved better performance than RF.

6 Discussion

In this section, we investigate the experimental results in more detail to reveal the characteristics and effectiveness of our method.

First, we decompose the experimental results per legal term in order to determine whether our method is good at predicting legal terms. Table 3 shows the accuracies of each legal term (averaged in results of five-fold cross validation). According to Table 3, RF+ achieved the best accuracy on average. It is also noteworthy that RF+ performed better than RF for almost every legal term except *kharachakan-kanmueang*, especially for *nathi-lae-amnat*. However, although RF has the same characteristic, RF+ tends to choose more frequent legal terms so that the

[3] https://github.com/google-research/bert

[4] https://colab.research.google.com/

Table 3: Accuracy per legal term

Legal term	Count	BERT	RF	RF+
yang-nueng-yang-dai	1,469	88.1%	91.8%	**95.4%**
yang-dai-yang-nueng	1,152	**81.9%**	73.4%	75.4%
khon-dai-khon-nueng	489	97.1%	97.5%	**98.4%**
khon-nueng-khon-dai	268	**86.6%**	76.9%	80.0%
amnat-nathi	5,631	93.2%	97.8%	**98.5%**
amnat-lae-nathi	977	**64.0%**	43.5%	53.1%
nathi-lae-amnat	519	41.3%	19.0%	**59.9%**
panakngan-chaonathi	8,006	96.1%	97.4%	**98.1%**
chaonathi	4,579	**93.6%**	75.1%	88.0%
kharachakan -kanmueang	595	**97.6%**	96.5%	96.2%
phu-damrong-tamnaeng -thang-kanmueang	411	**91.4%**	78.3%	89.7%
Average		84.6%	77.0%	**84.8%**

Table 4: Most important features

#	Set1-1	Set1-2	Set2	Set3	Set4
1	w-1	w+2	y	w+1	w-3
2	w-4	w-3	t-k_1	w+2	w-2
3	w-2	w-1	w-2	t-k_2	y
4	w+1	w+5	w-3	t-k_3	w-1
5	w+3	w+1	w+2	y	t-k_1
6	w-5	w-9	w-5	s-k_4	w-4
7	y	w-2	t-k_2	t-k_5	s-k_6
8	w-9	w+3	w-4	w-1	w+2
9	w-7	w-4	w-7	w+3	w+5
10	w-6	w-7	w-6	w+4	w+3

accuracies of less frequent legal terms are generally lower than those of the BERT method.

Next, we look at the feature importance of Random Forest classifiers. Table 4 shows the 10 most important features for each legal term set. In Table 4, "w+i" means the i-th right word, "w-i" means the i-th left word, "y" means the year feature, t-k indicates the existence of keyword k in the title, and s-k indicates the existence of keyword k in the section. Here, k_1, k_2, k_3, k_4, k_5, and k_6 mean รัฐธรรมนูญ (*ratthamnun*; constitution), ว่าด้วย (*waduai*; regarding), มาตรา (*matra*; section), รัฐ (*rat*; state), เจ้าหน้าที่ (*chaonathi*; officer), and ศาลฎีกา (*sandika*; supreme court), respectively.

Although most of the important features were adjacent words, the year feature and some keywords became important features. For example, the year feature was the most important one in Set2 (*amnat-nathi*, *amnat-lae-nathi*, and *nathi-lae-amnat*). This is because *amnat-lae-nathi* is a newer legal term (refer to Section 2.2). Also, รัฐธรรมนูญ (*ratthamnun*; constitution) is an important keyword in the legal term set because constitutions only use *amnat-nathi* out of the three legal terms.

The advantage of our RF+ model is not only prediction performance, but also feasibility. In terms of training cost, we need just an ordinary personal computer to train our RF+ model, while we need a TPU (Tensor Processing Unit) and at least a GPU environment to train a BERT model. In addition to that, our RF+ model is quite small compared to a BERT model. In the settings of our experiment, the total amount of RF+ models was less than 40 MB (varying from 2 MB to 20 MB per legal term set), while the total amount of BERT models was about 8 GB (1.6 GB per legal term set), which was 200 times larger than the RF+ models.

7 Summary

In this paper, we proposed a legal term correction method for Thai statutory sentences. Our method uses Random Forest classifiers to determine each legal term, to which we introduced three types of additional features from outside of the sentence: year feature, title keyword feature, and section keyword feature. Our experiment has shown that our method outperformed not only the existing Random Forest-based method, but also a method with BERT, the state-of-the-art language representation model, in overall accuracy.

Acknowledgments

This work was partly supported by JSPS KAKENHI Grant Numbers 18H03492 and the Graduate Program for Real-World Data Circulation Leaders, Nagoya University.

References

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45:5–32.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv preprint, arXiv:1810.04805, 13 pages.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. *Efficient Estimation of Word Representations in Vector Space*. In *Proceedings of the International Conference on Learning Representations*, 12 pages.

Andriy Mnih and Koray Kavukcuoglu. 2013. *Learning Word Embeddings Efficiently with Noise-contrastive Estimation*. In *Proceedings of the Advances in Neural Information Processing Systems 26*, pages 2265–2273.

Koki Mori, Makoto Miwa, and Yutaka Sasaki. 2015. *Sentence Completion by Neural Language Models Using Word Order and Co-occurrences*. In *Proceedings of the 21st Annual Meeting of the Association for Natural Language Processing*, pages 760–763 (In Japanese).

Office of the Council of State. 2008. *Legislative Drafting Manual* (In Thai).

Rachata Tipchod and Viriya KhotchaSeni. 2013. *Thai Law Dictionary Thai – English*. Soutpaisal Press, Thailand.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. *Attention is All You Need*. In *Proceedings of Advances in Neural Information Processing Systems 30*, pages 6000–6010.

Takahiro Yamakoshi, Takahiro Komamizu, Yasuhiro Ogawa, and Katsuhiko Toyama. 2018. *Japanese Legal Term Correction using Random Forest*. In *Legal Knowledge and Information Systems, JURIX 2018: The Thirty-first Annual Conference*, 313:161–170, IOS Press, the Netherlands.

Geoffrey Zweig and Chris J.C. Burges. 2011. *The Microsoft Research Sentence Completion Challenge*. Technical Report of *Microsoft Research*.