2019 IEEE International Conference on Big Data (Big Data)

Japanese Mistakable Legal Term Correction using Infrequency-aware BERT Classifier

Takahiro Yamakoshi*, Takahiro Komamizu*, Yasuhiro Ogawa*, and Katsuhiko Toyama*
*Nagoya University, Japan
Email: yamakoshi@kl.itc.nagoya-u.ac.jp, taka-coma@acm.org, {yasuhiro, toyama}@is.nagoya-u.ac.jp

***Abstract*—We propose a method that assists legislative drafters in locating inappropriate legal terms in Japanese statutory sentences and suggests corrections. We focus on sets of mistakable legal terms whose usages are defined in legislation drafting rules. Our method predicts suitable legal terms using a classifier based on a BERT (Bidirectional Encoder Representations from Transformers) model. We apply three techniques in training the BERT classifier, specifically, preliminary domain adaptation, repetitive soft undersampling, and classifier unification. These techniques cope with two levels of infrequency: legal term-level infrequency that causes class imbalance and legal term set-level infrequency that causes underfitting. Concretely, preliminary domain adaptation improves overall performance by providing prior knowledge of statutory sentences, repetitive soft undersampling improves performance on infrequent legal terms without sacrificing performance on frequent legal terms, and classifier unification improves performance on infrequent legal term sets by sharing common knowledge among legal term sets. Our experiments show that our classifier outperforms conventional classifiers using Random Forest or a language model, and that all three training techniques contribute to performance improvement.**

***Index Terms*—legal term, term correction, Japanese, BERT**

I. Introduction

Legislation drafting requires careful attention. The Japanese government deals with this task by thorough legislation drafting rules and final inspection by the Cabinet Legislation Bureau. The drafting rules regulate the document structures, orthography, and phraseology of the statutes. These rules have been in practice for more than 100 years and are published as legislation manuals (e.g., [1]).

The drafting rules have a noteworthy feature; they explicitly define distinct usage and meaning to many mistakable legal terms. For example, the rules prescribe the usage of three Japanese words "者 $_{(a)}$," "物 $_{(b)}$," and "もの $_{(c)}$" that are all pronounced *mono*. The term (a) only means a natural or juristic person, the term (b) only means a tangible object that is not a natural or juristic person, and the term (c) only means an abstract object or a complex of these objects. Figure 1 displays phrases including these legal terms. Unlike in statutory sentences, (c) can refer to (a) and (b) in ordinary Japanese sentences. For example, we can use (c) as "著作物 を 創作する もの $_{(c)}$" to express "a person who creates a work" in ordinary Japanese sentences, but, to avoid ambiguity, cannot do so in Japanese statutory sentences.

chosakubutsu	*wo*	*sosakusuru*	*mono*
著作物	を	創作する	者$_{(a)}$
work	ACC	create	person

【a person who creates a work】

chikuonkiyoonban	*,*	*rokuontepu*	*sonotano*	*mono*
蓄音機用音盤	、	録音テープ	その他の	物$_{(b)}$
phonograph disc	,	recording tape	such as	tangible object

【a material object such as a phonograph disc or recoding tape】

shiso	*matawa*	*kanjo*	*wo*	*sosakutekini*	*hyogenshita*	*mono*
思想	又は	感情	を	創作的に	表現した	もの$_{(c)}$
thought	or	sentiment	ACC	creatively	expressed	abstract object

【an object which creatively expresses thoughts or sentiments】

Phrases are from the Copyright Act (Act No. 48 of 1970)

Fig. 1. Phrases in Japanese statutory sentences with a legal term (underlined)

Using the drafting rules, legislative officers in the Cabinet Legislation Bureau strictly inspect legislative bills that are prudently written in the Cabinet Office or in each ministry, including the legal term usage. Therefore, any legal term defined in the rules must not be vague or ambiguous in inspected bills. Legislative drafters in local governments must also inspect their ordinances and rules following these drafting rules. Furthermore, many Japanese legal documents, such as contracts, terms of use, and articles of incorporation, are also written in accordance with the rules. However, inspections of those documents are still conducted mainly by human experts in legislation, which requires deep knowledge and an enormous amount of labor. According to Enami [2], this legislative work has become even tougher because of the recent incremental increase in enacted statutes.

Checking and correcting mistakable legal terms are specialized tasks of word usage checking. Word usage checking is a subtask of proofreading in concurrence with spelling, grammar, paraphrasing, sentence organization optimization, content optimization, and so on [3]–[5]. Although there are a number of studies on proofreading methods (e.g., [6]–[10]), to our knowledge, few methods focus on legal terms except Yamakoshi et al. [11], [12].

Prior to developing a legal term correction method, Yamakoshi et al. [11] positioned the task as a special case of the multi-choice sentence completion test by regarding a

This work was partly supported by JSPS KAKENHI Grant Number 18H03492.

978-1-7281-0858-2/19/$31.00 © 2019 IEEE

set of mistakable legal terms as a set of choices. In this setting, they proposed a method that uses Random Forest classifiers [13], each of which is optimized for each set of mistakable legal terms. Classifiers of [11] input words adjacent to the targeted legal term and output the most adequate legal term in the targeted legal term set. Through experiments, [11] showed that the Random Forest classifiers outperformed neural language models such as Continuous Bag-of-Words (CBOW) [14] and vector Log-bilinear model (vLBL) [15]. Classifiers of [12] incorporate outside-the-sentence features such as title keywords and section keywords to adapt them to Thai legal terms.

However, they suffer from two levels of infrequency: *legal term-level infrequency* and *legal term set-level infrequency*. Legal term-level infrequency means relative infrequency of a legal term in its legal term set, which may cause a class imbalance problem; that is, the classifier tends to choose frequent terms. Legal term set-level infrequency means absolute infrequency of a legal term set, which may cause underfitting since the method builds an individual classifier for each legal term set, but the number of training examples is not enough.

To cope with the aforementioned problems, we propose a method for legal term correction with a classifier based on BERT (Bidirectional Encoder Representations from Transformers) [16]. Our BERT classifier has an abundant amount of linguistic knowledge by fine-tuning a “ready-made” model that is pretrained by a large quantity of texts. Furthermore, it utilizes more context than the conventional classifiers, since a BERT classifier can handle one or more whole sentences (128 tokens maximum in our experiment), while the conventional ones do several adjacent words (4 to 30 words in their experiment).

We solve the aforementioned problems by using three techniques in training. The first technique is to preliminarily adapt the pretrained BERT model to Japanese statutory sentences, which contributes to overall performance improvement by providing prior knowledge of statutory sentences. The second technique is to undersample training examples softly and repetitively, which solves the class imbalance problem without sacrificing performance on frequent legal terms. The third technique is to unify classifiers for each legal term set into one model, which solves the underfitting problem of infrequent legal term sets by sharing common knowledge and reduces the total model size.

This paper contributes to the legal term correction task by:

- **a new method** of a BERT classifier with three training techniques that overcome the infrequency problems;
- **performance improvement** in both frequent and infrequent legal terms over the conventional classifiers; and
- **effectivity** of each training technique: preliminary domain adaptation improves prediction performance especially in earlier iterations, an adequate setting of repetitive soft undersampling improves accuracy in not only infrequent legal terms but also frequent legal terms, and classifier unification improves accuracy in infrequent legal term sets.

This paper is structured as follows. In Section II, we introduce the Japanese legal terms regulated in the drafting rules. In Section III, we explain the legal term correction task and its conventional classifiers [11]. In Section IV, we introduce BERT and describe our method in Section V. In Section VI, we present the evaluation experiment and its discussion. In Section VII, we investigate related work and, finally, we summarize this paper in Section VIII.

II. Japanese Legal Terms

The Japanese legislation drafting rules define a number of sets of mistakable legal terms and their usage. The list below illustrates three sets of mistakable legal terms:

- “規定$_{(d)}$” and “規程$_{(e)}$” (both are pronounced as *kitei*)
  Both (d) and (e) are nouns and share the concept of “rules.” However, (d) means a particular rule defined in a paragraph in a statute, while (e) means a suite of rules defined in a statute. According to the Standard Legal Terms Dictionary [17], (d) should be translated as “provision” and (e) should be translated as “rules,” “procedure,” or “regulation.”
- “直ちに$_{(f)}$” (*tadachini*), “速やかに$_{(g)}$” (*sumiyakani*), and “遅滞なく$_{(h)}$” (*chitainaku*)
  These are adverbs and share the concept of “speedily.” In Japanese statutory sentences, these words express different degrees of speed: (f), (g), and (h) express most, moderately, and least speedy, respectively. This strict difference does not exist in general Japanese sentences. According to the dictionary, (f), (g), and (h) should be translated as “immediately,” “promptly,” and “without delay,” respectively.
- “前項 の 場合 に おいて$_{(i)}$” (*zenko no baai ni oite*) and “前項 に 規定する 場合 に おいて$_{(j)}$” (*zenko ni kiteisuru baai ni oite*)
  Both of these phrases are conjunctive and share the concept of “mentioning the preceding paragraph.” In Japanese statutory sentences, (i) is used to mention the whole paragraph, while (j) is used to mention only the condition prescribed in the paragraph. According to the dictionary, (i) and (j) should be translated as “in the case referred to in the preceding paragraph,” and “in the case prescribed in the preceding paragraph,” respectively.

We note that legal terms have wide grammatical diversity; each legal term can be a noun, a verb, and so forth. Furthermore, some legal terms consist of multiple words like (i) and (j). Frequency of legal terms may vary largely both inside a legal term set and among legal term sets. For example, in our experimental dataset, (d) occurs 401,381 times, while (e) occurs 4,139 times, where (e) causes legal term-level infrequency. Also, legal terms in the legal term set $\{$(d), (e)$\}$ occur 405,520 times while legal terms in the set $\{$(i), (j)$\}$ occur only 3,159 times, where the latter legal term set causes legal term set-level infrequency.

III. Legal Term Correction

In this section, we explain the legal term correction task proposed by Yamakoshi et al. [11]. First, we show the definition and a general algorithm for the task in Section III-A.

Algorithm 1 Algorithm for legal term correction

```
Input: $W$, $T$
Output: Suggests
  Suggests $\leftarrow \emptyset$
  for all $(i, j)$ such that $w_i\ w_{i+1} \cdots w_j = t \in T$ do
    $W_l \leftarrow w_1\ w_2 \cdots w_{i-1}$
    $W_r \leftarrow w_{j+1}\ w_{j+2} \cdots w_{|W|}$
    $\hat{t} \leftarrow \arg\max_{t' \in T} score(W_l, t', W_r)$
    if $t \neq \hat{t}$ then
      Suggests $\leftarrow$ Suggests $\cup$ { a suggestion that $t$ in position
      $(i, j)$ should be replaced into $\hat{t}$}
    end if
  end for
```

Next, we introduce technologies for the scoring functions [11] to predict legal terms in Section III-B.

A. Definition

The definition of the legal term correction task is as follows:

- A sentence $W = w_1 w_2 \cdots w_{|W|}$ and a set of mistakable legal terms $T = \{t_1, t_2, \cdots, t_{|T|}\} \subseteq V^+$ are given, where V^+ is the Kleene plus of the vocabulary V, that is, a legal term $t \in T$ can be either a word or multiple words;
- Each legal term t in W is then judged as adequate or not;
- If another legal term $\hat{t} \in T$ $(\hat{t} \neq t)$ seems more adequate in the context, a term $\hat{t}$ is suggested as better than t.

A general algorithm for this is in Algorithm 1, where $score(W_l, t, W_r)$ is any scoring function that calculates the likelihood of t when two word sequences W_l and W_r are adjacent to the left and right of t, respectively.

For example, let the statutory sentence W and the legal term set T be as follows:

$$W = \begin{array}{cccccc} \textit{chosakubutsu} & \textit{wo} & \textit{sosakusuru} & \textit{mono} & \textit{no} & \textit{hogo} \\ \text{著作物} & \text{を} & \text{創作する} & \text{もの}_{\text{(c)}} & \text{の} & \text{保護}, \\ \text{work} & \text{ACC} & \text{create} & \text{a.object} & \text{of} & \text{protection} \end{array} \tag{1}$$

$$T = \{\ \text{者}_{\text{(a)}}, \text{物}_{\text{(b)}}, \text{もの}_{\text{(c)}}\}. \tag{2}$$

Here, T is the legal term set mentioned in Section I. In this case, the algorithm finds (c) $\in T$ from W. Then, it processes two word sequences W_l = 著作物を創作する (*chosakubutu wo sosakusuru*; creating a work) and W_r = の保護 (*no hogo*; protection of). Using W_l and W_r, it calculates scores of all legal terms as follows:

$$score\left(\begin{array}{ccccc} \textit{chosakubutsu} & \textit{wo} & \textit{sosakusuru}, & \textit{mono}, & \textit{no hogo} \\ \text{著作物} & \text{を} & \text{創作する}, & \text{者}_{\text{(a)}}, & \text{の 保護} \\ \text{work} & \text{ACC} & \text{create} & \text{, person} & \text{, of protection} \end{array}\right), \tag{3}$$

$$score\left(\begin{array}{ccccc} \textit{chosakubutsu} & \textit{wo} & \textit{sosakusuru}, & \textit{mono}, & \textit{no hogo} \\ \text{著作物} & \text{を} & \text{創作する}, & \text{物}_{\text{(b)}}, & \text{の 保護} \\ \text{work} & \text{ACC} & \text{create} & \text{, t.object} & \text{, of protection} \end{array}\right), \tag{4}$$

$$score\left(\begin{array}{ccccc} \textit{chosakubutsu} & \textit{wo} & \textit{sosakusuru}, & \textit{mono}, & \textit{no hogo} \\ \text{著作物} & \text{を} & \text{創作する}, & \text{もの}_{\text{(c)}}, & \text{の 保護} \\ \text{work} & \text{ACC} & \text{create} & \text{, a.object} & \text{, of protection} \end{array}\right). \tag{5}$$

Each calculates the likelihood of placing legal terms (a) (person), (b) (tangible object), and (c) (abstract object) instead of (c) in W, respectively. We highly expect the algorithm to choose the first option and to output a suggestion that (c) in W should be replaced into (a).

This problem is positioned as a kind of multi-choice sentence completion test by introducing the following ideas [11]:

- W_l ____ W_r is the sentence with a blank, where ____ is a 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.

B. Scoring Functions

In the general sentence completion test (e.g., [18]), a language model is usually used as the scoring function. However, Yamakoshi et al. [11], [12] suggested using Random Forest [13] classifiers to achieve better performance. In this section, we explain technologies for the scoring function that is examined in [11], [12]. First, we look at language models in Section III-B1 and then discuss Random Forest in Section III-B2.

1) Language Models: A language model outputs a distribution of a word w_i in the sequence $W = w_1\ w_2 \ldots w_{|W|}$ by inputting neighboring words (e.g., w_{i-1} and w_{i-2}). We can utilize the language model as a scoring function by probabilities of the targeted legal terms from the word distribution.

The simplest language model is n-gram, which predicts a word by previously occurring $n-1$ words. In addition to n-gram, Yamakoshi et al. [11] examined several neural language models, namely, the Continuous Bag-of Word (CBOW) model [14], Continuous Skip-gram (Skipgram) model [14], vector Log-bilinear (vLBL) model [15] and its extensions (vLBL(c) [19], and vLBL+vLBL(c) [19]). These neural language models overcome the curse of dimension by treating each word and each sequence of words as vectors [20].

2) Random Forest: A Random Forest classifier learns the training data by building a set of decision trees. A decision tree is conceptually a suite of if-then rules. A random forest 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 a less reliable decision, the ensemble of decision trees predicts unseen data better.

Yamakoshi et al. [11] suggested using Random Forest classifiers as the following scoring function $score_{RF_T}$:

$$score_{RF_T}(W_l, t, W_r) = \sum_{d \in D_T} P_d(t | w_l^{|W_l|-N+1}, \ldots, w_l^{|W_l|-1}, w_l^{|W_l|}, w_r^1, w_r^2, \ldots, w_r^N), \tag{6}$$

where D_T is a set of decision trees for the legal term set T and $P_d(t|w_1, w_2, \ldots, w_N)$ is the probability (actually 0 or 1) that $d \in D_T$ chooses a legal term $t \in T$ based on features $w_1, w_2, \ldots, w_N$. w_l^i and w_r^i are the i-th word of W_l and W_r, respectively, and N is the window size (the number of left or right adjacent words). They prepared a Random Forest classifier for each legal term set T so that we use the scoring function $score_{RF_T}$ for the legal term $t \in T$. Yamakoshi et al. [12] incorporated additional features from outside of the sentence such as title keywords and section keywords to the above function to cope with Thai legal terms.

However, we recognize that this methodology has two problems ascribed to two different levels of infrequency. The first is *legal term-level infrequency* that causes a class imbalance problem. As also pointed out in [11], the classifiers prefer majority legal terms to minority ones in imbalanced datasets. For example, they reported that this method predicted two mistakable legal terms "又は" (*matawa*; or) and "若しくは" (*moshikuwa*; or) at 99.0% and 33.9% accuracy, respectively. In their test dataset, occurrences of these terms are 9,116 and 2,425, respectively.

The second is *legal term set-level infrequency* that causes an underfitting problem. This happens because the method builds classifiers separately for each legal term set so that we cannot prepare enough training examples if the legal term set is infrequent.

IV. BERT Classifier

Instead of Random Forest classifiers, we use a BERT (Bidirectional Encoder Representations from Transformers) [16] classifier as the scoring function. BERT is a general language representation model along with ELMo (Embeddings from Language Models) [21] and others. In this section, we introduce an outline of such models first and then dig into BERT.

General language representation models such as ELMo and BERT have recently recorded remarkable performance in many natural language processing (NLP) tasks (e.g., question answering [22] and language inference [23]). Their good performance is due to two common key points. First, their word representations are pretrained by a huge amount of texts. Second, they are designed to be diverted to various NLP tasks by inheriting the pretrained representations and attaching input and output modules for each target task.

The main component of BERT is Transformer [24], a highly sophisticated neural network architecture. 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 so the model is a m-class sentence-level classifier. We divide a BERT model into two parts for explanation: (1) the pretrained part whose parameters are inherited from the pretrained model and (2) the prediction part that is customized for each task.

The pretrained part consists of L Transformer layers that encode the input words into vectorized representations. Concretely, each Transformer unit Trm_j^i outputs a vector for the j-th token by exploiting its self-attention mechanism that calculates the relationship between the token and every token in the previous layer. The prediction part consists of the final Transformer layer and an output layer, connected by feedforward connections. In a sentence-level classification that we adopt for our task, only Trm_1^L connects with the output layer and other units are dropped.

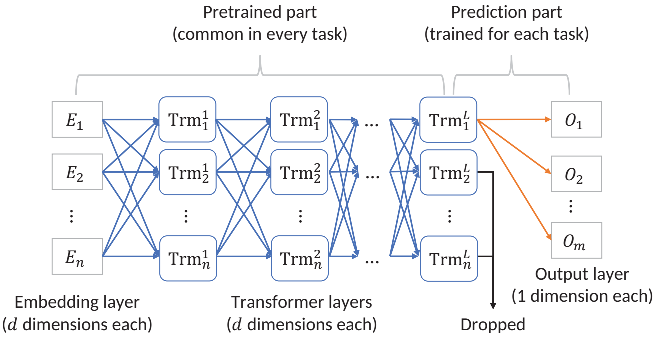

Fig. 2. BERT model

BERT pretraining is based on a multitask of the following two tasks: (1) masked language model and (2) next sentence prediction. In the masked language model task, the BERT model is required to answer words at randomly selected positions, where most of the selected words are masked. In the next sentence prediction task, the BERT model is required to answer whether a given sentence pair is continuous. Practically, examples for this pretraining have information for both tasks. The following denotes an example:

```
Input: [CLS] who are [MASK] ? [SEP] hi [MASK] am
       bert [SEP] [PAD]
Labels for Masked language model: [you, i]
Label for Next sentence prediction: True,
```

where "[CLS]" is the meta token for sentence classification tasks, "[SEP]" is the separator of two sentences, and "[PAD]" is used for padding. Since BERT is pretrained by sentences, we can input a whole sentence for classification.

V. Proposed Method

In this section, we describe our proposed method for the legal term correction task. Section V-A shows an overview of our method. Section V-B describes our training scheme with three techniques to solve the problems ascribed to data infrequency.

A. Overview

As defined in Section III-A, our classifier receives a sentence $W = w_1 w_2 \cdots w_{|W|}$ and a legal term set $T = \{t_1, t_2, \cdots, t_{|T|}\}$, and then finds targeted legal term t from W and evaluates it with the scoring function $score(W_l, t, W_r)$. Here, we utilize a BERT classifier. It inputs a "masked" sentence where the targeted legal term t is hid and outputs a probability distribution of the legal terms in t's legal term set. Therefore, our BERT classifier is a sentence-level classifier.

The following equation shows our scoring function:

$$score_{BERT}(W_l, t, W_r) = BERT(t|S), \tag{7}$$

where $BERT(t|S)$ is a probability of t that the BERT classifier assigns from the given processed sentence S made as follows:

$$S = pp(w_l^1 w_l^2 \cdots w_l^{|W_l|} \text{ [MASK] } w_r^1 w_r^2 \cdots w_r^{|W_r|}), \tag{8}$$

where $pp(W)$ is a function to truncate the input sentence W on the masked legal term "[MASK]" that was originally t. Even

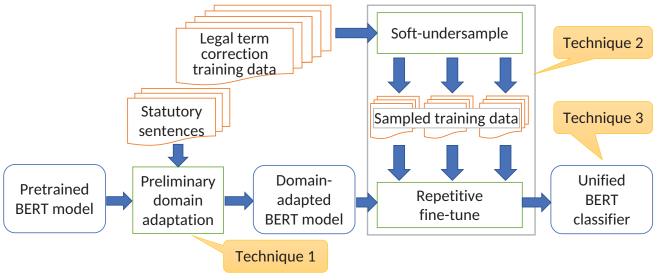

Fig. 3. Training scheme

Algorithm 2 Repetitive soft undersampling

```
Input: $E_{\text{all}}$, $\beta$, $I$
Output: Model
Initialize parameters of Model
for $i \leftarrow 1$ to $I$ do
    $E_{\text{sus}} \leftarrow \{\}$
    for $E$ in $E_{\text{all}}$ do
        $E_{\text{sus}} \leftarrow E_{\text{sus}} \cup choice(E, s(E, E_{\text{all}}; \beta))$
    end for
    Fine-tune Model using $E_{\text{sus}}$
end for
```

when the BERT classifier inputs sentences, it usually accepts the definite number of words (e.g., 128 tokens). However, this is larger than what the Random Forest classifiers accept (4–30 tokens).

We apply three techniques in training a BERT classifier. First, we feed statutory sentences to a BERT model prior to training legal term correction examples, which contributes to overall performance improvement by providing prior knowledge of statutory sentences. Second, we apply "soft and repetitive" undersampling for training data, which solves the class imbalance problem. Without this technique, the BERT classifier tends to choose frequent legal terms since biased prediction to majority classes in imbalanced data is universal in any classifier. Third, we train one unified BERT classifier used for all legal terms regardless of legal term sets, which solves the underfitting problem by sharing common knowledge.

B. Training the BERT Classifier

Figure 3 shows the scheme to train our BERT classifier. Here, we apply three techniques, namely, (1) Preliminary Domain Adaptation, (2) Repetitive Soft Undersampling, and (3) Classifier Unification to cope with the problems of the conventional classifiers. We describe each technique in the following sections.

1) Preliminary Domain Adaptation: We adapt a general-purpose BERT pretrained model to statutory sentences prior to training with legal term correction examples. We expect that this will work favorably for two reasons. First, we can feed prior knowledge on statutory sentences that will not be learned inside the framework of legal term correction. Second, we can accelerate convergence by filling the gap from the domain difference beforehand. Generally, publicly offered BERT pretrained models such as [24] are trained with general text such as a Wikipedia corpus and their scope is different from the statutory sentences that we focus on.

Specifically, we train a pretrained BERT model by statutory sentences in the same manner as the BERT pretraining procedure. That is, we feed the training examples of statutory sentences for the masked language model task and the next sentence prediction task. The following is an example:

```
Input: [CLS] 著作 物 を 創作する [MASK] の 保護 [SEP]
       この 法律 [MASK] 、 ... 。 [SEP]
(Meaning: [CLS] protection of [MASK] who creates a
          work [SEP] [MASK] this act , ... . [SEP])
Labels for Masked language model: [者, において]
(Meaning: [person, in])
Label for Next sentence prediction: False
```

As a result, we get a BERT pretrained model domain-adapted by statutory sentences.

2) Repetitive Soft Undersampling: When frequencies of legal terms in a legal term set are drastically different, a classifier tends to choose the more frequent legal terms. As Yamakoshi et al. [11] reported, this problem happened in their experiment. To solve this problem, we apply an undersampling strategy.

First, we apply "soft" undersampling that "weakens" the magnitude correlation among frequent classes and infrequent classes. Concretely, we undersample examples E_t of a legal term $t \in T$ as much as the value of the following function:

$$s(E_t, E_{\text{all}}; \beta) = |E_t| \cdot \left(\frac{\min\{|E| \mid E \in E_{\text{all}}\}}{|E_t|} \right)^{\frac{1}{\beta}}, \tag{9}$$

where $|E|$ is the number of examples E and $E_{\text{all}} = \{E_t | t \in C\}$ is the set of examples for all legal terms C. Here, $C = \bigcup_i T_i$, that is, C, contains all legal terms regardless of legal term sets. β is a hyperparameter that controls the strength of reduction. $\beta = 1$ creates naive undersampling, while large enough β (i.e., $1/\beta \approx 0$) creates no undersampling. The larger the β, the weaker the undersampling.

Next, to cover the majority of examples, we resample training examples from the whole dataset after certain iterations. This procedure resembles an ensemble training framework such as Bagging [25] and Boosting [26]. However, unlike them, we do not make an ensemble of BERT models since the size of a BERT model is quite huge[1].

Algorithm 2 shows the training algorithm of our model. $choice(E, n)$ is a function that randomly chooses n items from the example set E. I is the number of iterations.

As far as we are aware, there are few cases of applying undersampling to examples for a BERT classifier. Anand et al. [27] randomly undersampled the majority examples (and copied the minority examples). However, they did not repeat the undersampling process during training, while we do and we believe that it enhances performance.

[1] The BERT model that we used in the experiment is more than 1 GB.

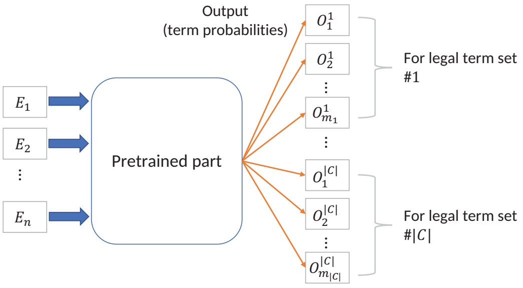

Fig. 4. Unified BERT classifier (global classification)

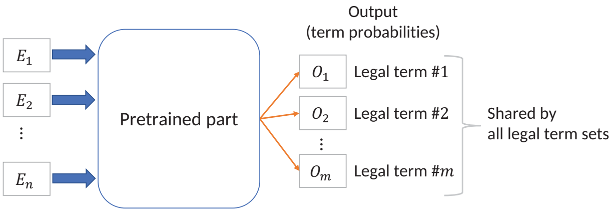

Fig. 5. Unified BERT classifier (merged classification)

3) Classifier Unification: Yamakoshi et al. [11], [12] built classifiers for each legal term set; however, this may cause two problems. One is the underfitting problem caused by the legal term set-level infrequency. The other is a storage problem; that is, we need a huge amount of storage to keep all classifiers, especially when we use BERT (more than 1 GB × the number of legal term sets).

To solve these problems, we propose building a unified classifier that handles all of the legal term sets. We feed examples for legal terms to one unified classifier, regardless of its belonging to a legal term set. That is, the parameters of the classifier are shared among all legal terms so that the classifier can use broader knowledge in predicting legal terms of an infrequent legal term set.

For the output layer, we consider two approaches: global classification and merged classification. Figures 4 and 5 show our unified classifier models with global classification and merged classification, respectively. For global classification, the model outputs likelihoods of all legal terms. It then selects the legal term with the highest likelihood from the outputs of the targeted legal term set. On the other hand, the merged classification model outputs likelihoods of only the targeted legal term set. Here, each unit position in the output layer is shared by legal terms having the same position in their legal term sets. Therefore, m in the figure should be the maximum number of elements of targeted legal term sets. For example, we assume that we have two (ordered) legal term sets $\{$foo, bar$\}$ and $\{$baz, qux$\}$. Then, the output layer has two units O_1 and O_2, where O_1 and O_2 should output likelihoods of "foo" and "baz," and "bar" and "qux," respectively.

VI. Experimental Study

We conducted experiments on predicting legal terms in Japanese statutory sentences to answer the following two questions: (1) whether our method performs better than the conventional classifiers and (2) whether the three training techniques are effective.

TABLE I
List of nominal legal term sets

Legal term (pronunciation; meaning)	Count
規定 (*kitei*; provision)	401,381
規程 (*kitei*; rules, procedure, regulation)	4,139
とき (*toki*; if, when, whenever (condition))	127,861
場合 (*baai*; if, when, whenever (condition))	188,970
時 (*toki*; when (meaning a certain time))	17,808
者 (*mono*; natural or juristic person*)	353,279
物 (*mono*; tangible object*)	29,689
もの (*mono*; abstract object*)	231,715
許可 (*kyoka*; permission, license)	24,145
認可 (*ninka*; authorization, approval, permission, confirmation)	15,677
届出 (*todokede*; notification, report, registration)	22,021
認証 (*ninsho*; certification, accreditation)	1,949
通知 (*tsuchi*; notice)	17,894
通報 (*tsuho*; notification, report, information)	768
報告 (*hokoku*; report)	16,675
連絡 (*renraku*; contact, liaison*)	1,475
命令 (*meirei*; order, direction)	15,176
指揮 (*shiki*; direction, command)	570
指示 (*shiji*; instruction*)	3,034
監督 (*kantoku*; supervision)	3,732
要求 (*yokyu*; requirement*)	1,606
要請 (*yosei*; request*)	1,671
施行 (*seko*; enforcement*)	233,058
適用 (*tekiyo*; application)	83,062
準用 (*jun'yo*; application mutatis mutandis*)	2,041

A. Experiment Settings

We compiled a statutory sentence corpus from e-Gov Statute Search[2] provided by the Ministry of Internal Affairs and Communications, Japan. We acquired 3,983 existing Japanese acts and cabinet orders on May 18, 2018. Next, we tokenized each statutory sentence in the corpus by MeCab (v.0.996), a Japanese morphological analyzer. Statistics of the corpus are as follows: the total number of sentences is 1,223,084, the total number of tokens is 46,919,612, and the total number of different words is 41,470. We divided the 3,983 acts and cabinet orders in the corpus into training data and test data. The training data has 3,784 documents, where there are 1,185,424 sentences and 43,655,941 tokens in total. The test data has 199 documents with 37,660 sentences and 1,557,587 tokens in total. There are 251,085 legal terms in the test data.

We defined 27 legal term sets by referencing the Japanese legislation manual [1]. Tables I to IV show all legal term sets. English translations in this table are taken from the Standard Legal Terms Dictionary (March 2018 edition) [17] provided by the Ministry of Justice, Japan, except for asterisked items.

We compared our BERT classifier with the following classifiers and language models: Random Forest [13], CBOW [14], Skipgram [14], vLBL [15], vLBL(c) [19], vLBL+vLBL(c) [19], and n-gram language.

[2] http://elaws.e-gov.go.jp/

TABLE II
LIST OF VERBAL LEGAL TERM SETS

Legal term (pronunciation; meaning)	Count
処する (*syosuru*; punish*)	4,841
科する (*kasuru*; impose (fine or punishment)*)	1,212
課する (*kasuru*; impose (tax)*)	4,320
適用する (*tekiyosuru*; apply)	21,119
準用する (*jun'yosuru*; apply mutatis mutandis)	66,303
例による (*reiniyoru*; is governed by)	4,368
推定する (*suiteisuru*; presume)	228
みなす (*minasu*; deem)	20,039
とする (*to suru*; shall be)	131,314
である (*de aru*; be*)	58,020
する こと が できない (*suru koto ga dekinai*; may not, be unable to)	6,783
して は ならない (*shite ha naranai*; must not, is prohibited)	4,457
する こと が できる (*suru koto ga dekiru*; may)	29,348
しなければ ならない (*shinakereba naranai*; must, shall)	42,679
する もの と する (*suru mono to suru*; is to)	11,501
この 限り で ない (*kono kagiri de nai*; does not apply to*)	7,380
妨げない (*samatagenai*; does not preclude)	1,419
なお 従前 の 例による (*nao juzen no reiniyoru*; prior laws continue to govern)	36,402
なお 効力 を 有する (*nao koryoku wo yusuru*; remain in force*)	2,734
改める (*aratameru*; revise (an expression))	6,941
改正する (*kaisei suru*; revise (a statute))	24,349

TABLE III
LIST OF ADJECTIVE AND ADVERBIAL LEGAL TERM SETS

Legal term (pronunciation; meaning)	Count
当該 (*togai*; that, the, referenced, relevant)	297,904
その (*sono* ; that*)	213,114
に 係る (*ni kakaru*; pertaining to*)	161,564
に 関する (*ni kansuru*; regarding*)	120,076
に 関係する (*ni kankeisuru*; regarding*)	80
の (*no*; of*)	2,813,563
に 規定する (*ni kiteisuru*; provided for in, prescribed in*)	169,742
の 規定 に よる (*no kitei ni yoru*; pursuant to, under the provisions of*)	113,123
直ちに (*tadachini*; immediately)	2,414
速やかに (*sumiyakani*; promptly)	2,229
遅滞なく (*chitainaku*; without delay)	6,549
に基づき (*ni motozuki*; based on*)	8,883
により (*niyori*; by*)	205,212

TABLE IV
LIST OF CONJUNCTIONAL LEGAL TERM SETS

Legal term (pronunciation; meaning)	Count
又は (*matawa*; or*)	337,058
若しくは (*moshikuwa*; or*)	88,241
及び (*oyobi*; and*)	301,460
並びに (*narabini*; and*)	49,584
その他の (*sonotano*; other*)	55,391
その他 (*sonota*; other*)	29,163
前項 の 場合 に おいて (*zenko no baai ni oite*; in the case referred to in the preceding paragraph)	2,834
前項 に 規定する 場合 に おいて (*zenko ni kiteisuru baai ni oite*; in the case prescribed in the preceding paragraph)	325
ただし (*tadashi*; provided, however, that ⋯)	39,234
この 場合 に おいて (*kono baai ni oite*; in this case)	20,788

Our BERT classifier is based on a publicly available BERT model pretrained by Japanese Wikipedia text [3]. The model's specs are as follows: 12 Transformer layers, 768 hidden vectors, and 12 heads. It has a vocabulary of 32,000 subwords. It receives a maximum of 128 tokens; therefore, we truncate each example to 128 words. For preliminary domain adaptation, we used a script provided by the authors of BERT. The script generated 467,382 examples from 1,500 documents randomly sampled from the training data. We set the iteration number to 150,000 and the batch size to 32; i.e., the number of epochs is 10.27 ($150000 \times 32/467382$). For soft-undersampling, we set β in Eq. 9 to 3 and undersample iterations I to 100. In this setting, 42,721,500 examples (including duplication) are trained. At each iteration of soft-undersampling, we fine-tuned the model by the following settings: the number of epochs is 5, minibatch size is 512, warmup proportion is 0.1, and learning rate is 2e-5. For classifier unification, we adopted the global classification. Implementation, training, and testing were done by TensorFlow on Colaboratory [4].

For Random Forest, we used the Gini coefficient to build decision trees and we optimized the window size $\{2, 5, 10, 15\}$, the number of decision trees $\{10, 50, 100, 500\}$, and the maximum depth of each tree $\{10, 100, 1000, \text{unlimited}\}$ by five-fold cross validation. Implementation, training, and testing were done by Scikit-learn (v.0.19.1). For neural language models (CBOW, Skipgram, vLBL, vLBL(c), and vLBL+vLBL(c)), we set the window size to 5 in accordance with their papers. Other parameters are as follows: the number of vector dimensions is 200, number of epochs is 5, minibatch size is 512, number of negatively sampled words is 10 (only in Skipgram and the vLBL family), and optimization function is Adam [28]. We implemented, trained, and tested the models by Chainer (v.1.7.0). For the n-gram model, we used Katz's backoff trigram and 4-gram [29], referencing Zweig and Burges [18].

Since neural language models and n-gram models are designed to predict a single word, we combined legal terms with multiple words into single words by the longest match principle. The total number of tokens in the corpus thus became 45,213,528. Also, we changed words that occur less than five times in the corpus into unknown words to reduce computational cost. In training and predicting words, we utilized an end-of-sentence token to pad short sequences.

B. Comparison with Conventional Classifiers

Table V shows the overall performance of each model. Here, we measured the accuracy of predicting legal terms in three averages: micro average acc_{micro}, macro average by legal term set $acc_{\text{macro-S}}$, and macro average by legal term $acc_{\text{macro-T}}$. As a baseline, we calculated the accuracy in maximum likelihood estimation (MLE), in which the most frequent legal terms in the train data are always selected.

Our BERT classifier achieved the best performance in all of acc_{micro}, $acc_{\text{macro-S}}$, and $acc_{\text{macro-T}}$. Notably, its $acc_{\text{macro-T}}$

[3] http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT日本語Pretrainedモデル

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

TABLE V
OVERALL PERFORMANCE

Classifier	acc_{micro}	$acc_{\mathrm{macro\text{-}S}}$	$acc_{\mathrm{macro\text{-}T}}$
BERT [Proposed]	**97.57%**	**96.15%**	**92.56%**
Random Forest	95.37%	93.22%	84.68%
CBOW	88.82%	84.65%	74.94%
Skipgram	75.42%	63.07%	65.68%
vLBL	80.23%	75.46%	74.17%
vLBL(c)	91.38%	86.32%	80.67%
vLBL+vLBL(c)	90.95%	85.62%	81.12%
Trigram	87.12%	85.81%	69.36%
4-gram	88.81%	87.83%	72.58%
MLE	78.61%	62.49%	38.81%

TABLE VI
ACCURACY BY FREQUENCY

Classifier	acc_{majority}	acc_{minority}	Difference
BERT [Proposed]	**96.15%**	**89.06%**	7.09
Random Forest	91.27%	78.27%	13.00
CBOW	81.23%	68.82%	12.41
Skipgram	65.33%	66.03%	-0.70
vLBL	77.47%	70.96%	6.51
vLBL(c)	85.17%	76.30%	8.87
vLBL+vLBL(c)	84.47%	77.86%	6.61
Trigram	78.15%	60.82%	17.33
4-gram	80.82%	64.57%	16.25

is 92.56%, which is 7.88 points better than Random Forest, whose score is the second best.

Next, we calculated $acc_{\mathrm{macro\text{-}T}}$ of majority legal terms and minority legal terms to find the sensitivity of frequency in each classifier. We consider legal terms whose frequency is over the median (649 times) as majority ones and the others as minority ones. Table VI shows the accuracies of the majority legal terms acc_{majority} and minority legal terms acc_{minority}, and their difference. BERT achieved the best accuracy in both majority and minority legal terms. Its accuracy for minority legal terms is notably more than 10 points better than any methods. By comparing the difference between the two accuracies, BERT, Skipgram, and the vLBL family tend to have less difference than Random Forest and n-gram. This means that neural-based methods tend to be more robust in imbalanced data.

Finally, we look at the accuracies for particular legal term sets. Table VII shows the accuracy of a legal term set { 者$_{(a)}$, 物$_{(b)}$, もの$_{(c)}$}. The BERT classifier achieved the best accuracy in every legal term in this legal term set. It well predicted the legal term 物$_{(b)}$, the least frequent legal term in the set, while Random Forest classifier dropped the accuracy for it. Table VIII shows the accuracy of a legal term set {直ちに$_{(f)}$, 速やかに$_{(g)}$, 遅滞なく$_{(h)}$}. The BERT classifier achieved the best accuracy except for (h). The trigram model predicted the legal term best; however, it predicted the other infrequent legal terms quite poorly. This is probably because of ambiguity of usage rules for those legal terms; although there are usage rules for them, there is no concrete border to determine them. Therefore, legislation officers use their judgment to select them to some extent. Because of this, a few adjacent words were not a good clue and the model resorted to choosing legal terms on a frequency basis.

TABLE VII
ACCURACY OF { 者$_{(a)}$, 物$_{(b)}$, もの$_{(c)}$}

Classifier	者$_{(a)}$	物$_{(b)}$	もの$_{(c)}$	acc_{micro}
BERT [Proposed]	**98.50%**	**98.25%**	**97.35%**	**98.03%**
Random Forest	92.84%	88.80%	95.46%	94.05%
CBOW	85.71%	86.88%	86.59%	86.25%
Skipgram	74.41%	81.28%	76.26%	75.79%
vLBL	88.90%	94.84%	78.98%	83.81%
vLBL(c)	94.02%	97.29%	88.81%	91.36%
vLBL+vLBL(c)	92.25%	97.11%	92.22%	92.50%
Trigram	85.37%	92.39%	84.97%	85.54%
4-gram	87.07%	93.35%	92.53%	90.39%
Frequency	8,389	1,143	11,440	

TABLE VIII
ACCURACY OF { 直ちに$_{(f)}$, 速やかに$_{(g)}$, 遅滞なく$_{(h)}$}

Classifier	直ちに$_{(f)}$	速やかに$_{(g)}$	遅滞なく$_{(h)}$	acc_{micro}
BERT [Proposed]	**68.29%**	**62.73%**	90.05%	**78.45%**
Random Forest	**68.29%**	37.27%	93.67%	73.61%
CBOW	62.20%	33.64%	85.97%	67.31%
Skipgram	50.00%	43.64%	68.78%	58.35%
vLBL	71.95%	42.73%	59.28%	57.38%
vLBL(c)	47.56%	29.09%	89.59%	65.13%
vLBL+vLBL(c)	57.32%	42.73%	65.61%	57.87%
Trigram	10.98%	17.27%	**98.19%**	59.32%
4-gram	13.41%	22.73%	95.48%	59.81%
Frequency	82	110	221	

C. Ablation Study

In this section, we evaluate the three techniques we introduced: (1) preliminary domain adaptation, (2) repetitive soft undersampling, and (3) classifier unification.

1) Preliminary Domain Adaptation: To evaluate the effect of preliminary domain adaptation, we first compare the domain-adapted BERT model and the pretrained BERT model in overall accuracy. Table IX shows acc_{micro}, acc_{majority}, and acc_{minority} between the domain-adapted model and the pretrained model. From this result, we find that preliminary domain adaptation worked favorably on minority legal terms. We suppose that introducing statutory sentence knowledge augmented performance for minority legal terms.

Next, we compare the transition of accuracy by the number of trained examples to evaluate the effect of preliminary domain adaptation for the convergence speed. Figure 6 shows the transition of accuracy in the domain-adapted BERT model and the pretrained BERT model. The domain-adapted model achieved better performance, especially in early iterations. Concretely, the domain-adapted model achieved 96.17% of acc_{micro} at the first iteration (427,215 examples trained), while the pretrained model achieved 95.88%, which is 0.29 points better. Furthermore, the domain-adapted model maintained

TABLE IX
EFFECTIVENESS OF PRELIMINARY DOMAIN ADAPTATION

Domain adaptation?	acc_{micro}	acc_{majority}	acc_{minority}
Yes	**97.57%**	**96.15%**	**89.06%**
No	97.49%	96.00%	88.38%

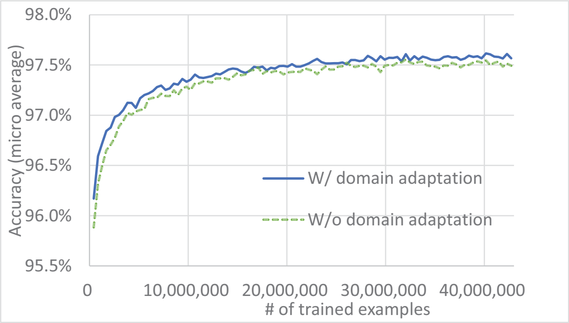

Fig. 6. Transition of Accuracy in the domain-adapted model and the pretrained model

TABLE X
EFFECTIVENESS OF UNDERSAMPLING

β	acc_{micro}	acc_{majority}	acc_{minority}
2	97.10%	95.94%	88.27%
3	**97.34%**	**95.99%**	**89.16%**
100	97.27%	95.59%	85.28%

superiority until the final iterations.

2) Repetitive Soft Undersampling: To evaluate the effect of repetitive soft undersampling, we made BERT classifiers with different β values the undersampling strength. Table X shows acc_{micro}, acc_{majority}, and acc_{minority} between different $\beta \in \{2, 3, 100\}$. Here, each BERT model trained approximately 10,000,000 examples. Weak undersampling (i.e., $\beta = 100$) resulted in worse accuracy for minority legal terms. On the other hand, setting a good β brought better accuracy for both majority and minority legal terms.

Figure 6 also demonstrates the effect of repetitive undersampling. Our (unified) BERT model predicted with 96.17% accuracy at the first iteration and 97.57% accuracy, 1.40 points better, at the last iteration.

3) Classifier Unification: To evaluate the effectiveness of classifier unification, we made a BERT classifier for each legal term set. Table XI shows the accuracy of the unified BERT classifier (proposed) and the separated BERT classifiers. The unified BERT classifier achieved better performance in every criterion. In particular, its accuracy for minority legal terms is 5.80 points better. This comes from the accuracy improvement in infrequent legal term sets. For example, Table XII shows the accuracies of a legal term set {要求$_{\text{(k)}}$ (*yokyu*; requirement), 要請$_{\text{(l)}}$ (*yosei*; request)}. Here, this legal term set has only 3,277 examples in the whole dataset, which is 1/100 of average occurrence. The separated classifier achieved only 55.32% accuracy in the term (k), the infrequent legal term, while the unified classifier achieved 79.17% accuracy.

TABLE XI
EFFECTIVENESS OF CLASSIFIER UNIFICATION

Unified?	acc_{micro}	acc_{majority}	acc_{minority}
Yes	**97.57%**	**96.15%**	**89.06%**
No	97.39%	95.97%	83.26%

TABLE XII
ACCURACY IN AN INFREQUENT LEGAL TERM SET

Unified?	要求$_{\text{(k)}}$	要請$_{\text{(l)}}$	acc_{micro}
Yes	**79.17%**	**92.75%**	**87.18%**
No	55.32%	**92.75%**	77.59%
Frequency	48	69	

TABLE XIII
COMPARISON OF GLOBAL CLASSIFICATION AND MERGED CLASSIFICATION

Type	acc_{micro}	acc_{majority}	acc_{minority}
Global	**97.57%**	**96.15%**	**89.06%**
Merged	95.43%	94.03%	85.57%

Next, we compare global classification and merged classification. Table XIII shows the accuracy of global classification and merged classification. The global classification overall performed better, specifically, 2.14, 2.12, and 3.49 points better accuracy in acc_{micro}, acc_{majority}, and acc_{minority}, respectively. It appears that sharing output positions caused a disturbance in the language modeling because a unit position itself does not have any meaning. In contrast, assigning an individual output to every class could avoid the problem by backpropagating the error of each class separately.

D. Lessons Learned

From the comparison with conventional classifiers, we found that our BERT-based classifier predicted legal terms better than any of the conventional classifiers. More concretely, our classifier outperformed the others in both overall and term set level accuracies. From the ablation study, we found that all three training techniques worked favorably. The preliminary domain adaptation improved prediction performance, especially in the earlier training phase. An adequate setting of the repetitive soft undersampling improved accuracy in both frequent and infrequent legal terms. The classifier unification improved accuracy in infrequent legal term sets.

VII. RELATED WORK

In this section, we look at existing studies of automatic proofreading. Many researchers have studied automatic proofreading methods. One of the oldest systems is CRI-TAC [6], which corrects misspellings using a rule-based and statistic search. Chae [7] also focuses on misspellings, using a confusion matrix built by a multidomain corpus. Cheng and Nagase [8] proposed an example-based system, focusing on Japanese written by non-natives, that corrects not only misspellings, but also inappropriate structures. Zhang and Litman [9] have a different point of view that focuses on predicting the purpose of revisions in proofread documents. A new and notable one is Hitomi et al. [10], which generates proofread sentences using a long short-term memory (LSTM) [30] neural network that copes with various types of correction flexibly.

Some of the above studies focus on word-level correction (e.g., [6], [7]), which is the same as our proofreading scope.

However, as far as we know, there is no study that especially focuses on mistakable expressions such as legal terms, except for Yamakoshi et al. [11], [12]. We consider that correcting mistakable legal terms is more challenging than general word correction because our task handles legal terms that are adequate in certain contexts, while the other task usually handles expressions that are inadequate everywhere.

Although there are a number of proofreading methods for general sentences, there are few proofreading methods for legal documents that are published in the academic field. However, as for commercial web services, there are many available automatic proofreading systems for legal documents. Examples include: LawGeex [5], Luminance [6], and Kira [7].

VIII. Conclusion

In this paper, we proposed a legal term correction method in Japanese statutory sentences that focuses on mistakable legal term sets whose usages are defined in the legislation drafting rules. In this study, we regarded this legal term correction as a special case of the multi-choice sentence completion test. We applied a BERT classifier with three training techniques to cope with the problems caused by two-level infrequency, specifically, preliminary domain adaptation, repetitive soft undersampling, and classifier unification. Our experiment demonstrated that our classifier outperforms the conventional classifiers and the three training techniques solved the problems ascribed to data infrequency.

References

[1] Hoseishitumu-Kenkyukai, *Workbook Hoseishitumu (newly revised second edition)*. Gyosei, 2018, In Japanese.

[2] T. Enami, "Rippobakuhatu to open government ni kansuru kenkyu — horeibunsyo niokeru "open coding" no teian —," Fujitsu Research Institute, Tech. Rep., 2015, In Japanese.

[3] L. Faigley and S. Witte, "Analyzing revision," *College composition and communication*, vol. 32, no. 4, pp. 400–414, 1981.

[4] N. L. Nguyen and Y. Miyao, "Alignment-based annotation of proofreading texts toward professional writing assistance," in *Proc. of the 6th International Joint Conference on Natural Language Processing*, 2013, pp. 753–759.

[5] F. Zhang and D. Litman, "Annotation and classification of argumentative writing revisions," in *Proc. of the 10th Workshop on Innovative Use of NLP for Building Educational Applications*, 2016, pp. 1424–1430.

[6] K. Takeda, T. Fujisaki, and E. Suzuki, "CRITAC — a Japanese text proofreading system," in *Proc. of the 11th International Conference on Computational Linguistics*, 1986, pp. 412–417.

[7] C. Young-Soog, "Improvement of korean proofreading system using corpus and collocation rules," in *Proc. of the 12th Pacific Asia conference on Language, Information and Computation*, 1998, pp. 328–333.

[8] Y. Cheng and T. Nagase, "An example-based Japanese proofreading system for offshore development," in *Proc. of the 24th International Conference on Computational Linguistics*, 1998, pp. 328–333.

[9] F. Zhang and D. Litman, "Using context to predict the purpose of argumentative writing revisions," in *Proc. of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1424–1430.

[10] Y. Hitomi, H. Tamori, N. Okazaki, and K. Inui, "Proofread sentence generation as multi-task learning with editing operation prediction," in *Proc. of the 8th International Joint Conference on Natural Language Processing*, 2017, pp. 436–441.

[11] T. Yamakoshi, T. Komamizu, Y. Ogawa, and K. Toyama, "Japanese legal term correction using random forests," *Legal Knowledge and Information Systems, JURIX 2018: The Thirty-first Annual Conference*, vol. 313, pp. 161–170, 2018.

[12] T. Yamakoshi, V. Satayamas, H. Chanlekha, Y. Ogawa, T. Komamizu, A. Kawtrakul, and K. Toyama, "Thai legal term correction using random forest with outside-the-sentence features," in *Proc. of the 33rd Pacific Asia Conference on Language, Information and Computation*, 2019, pp. 306–314.

[13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. of International Conference on Learning Representations*, 12 pages, 2013.

[15] A. Mnih and K. Kavukcuoglu, "Learning word embeddings efficiently with noise-contrastive estimation," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 2265–2273.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

[17] The Japanese Law Translation Council, "Standard Legal Terms Dictionary (March 2018 Edition)," 2018. [Online]. Available: http://www.japaneselawtranslation.go.jp/dict/download

[18] G. Zweig and C. J. Burges, "The Microsoft Research sentence completion challenge," Microsoft Research, Tech. Rep., 2011.

[19] K. Mori, M. Miwa, and Y. Sasaki, "Sentence completion by neural language models using word order and co-occurrences," in *Proc. of the 21st Annual Meeting of the Association for Natural Language Processing*, 2015, pp. 760–763, (In Japanese).

[20] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[21] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 2227–2237.

[22] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2383–2392.

[23] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proc. of the 2018 Conference on the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 1112–1122.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. of Advances in Neural Information Processing Systems 30*, 2017, pp. 6000–6010.

[25] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[26] Y. Freund and R. E. Schapire, "A short introduction to boosting," in *Proc. of the 16th International Joint Conference on Artificial Intelligence*, 1999, pp. 1401–1406.

[27] S. Anand, D. Mahata, H. Zhang, S. Shahid, L. Mehnaz, Y. Kumar, and R. R. Shah, "Midassmm4h-2019: Identifying adverse drug reactions and personal health experience mentions from twitter," in *Proc. of the 4th Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*, 2019, pp. 127–132.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of International Conference on Learning Representations*, 15 pages, 2015.

[29] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 400–401, 1987.

[30] M. Sundermeyer, R. Schulüter, and H. Ney, "Lstm neural networks for language modeling," in *Proc. of the 11th Annual Conference of the International Speech Communication Association*, 2010, pp. 194–197.

[5] https://www.lawgeex.com/

[6] https://www.luminance.com/

[7] https://kirasystems.com/