Legal Knowledge and Information Systems
M. Palmirani (Ed.)

doi:10.3233/978-1-61499-935-5-161

Japanese Legal Term Correction Using Random Forests

Takahiro YAMAKOSHI [a], Takahiro KOMAMIZU [a,b], Yasuhiro OGAWA [a,b] and Katsuhiko TOYAMA [a,b]

[a] *Graduate School of Informatics, Nagoya University*
[b] *Information Technology Center, Nagoya University*
Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan

Abstract. We propose a method that assists legislation officers in finding inappropriate Japanese legal terms in Japanese statutory sentences and suggests corrections. In particular, we focus on sets of similar legal terms whose usages are defined in legislation drafting rules. Our method predicts suitable legal terms in statutory sentences using Random Forest classifiers, each of which is optimized for each set of similar legal terms. Our experiment shows that our method outperformed existing modern word prediction methods using neural language models.

Keywords. Japanese Legal Terms, Legal Term Correction, Random Forest

1. Introduction

Legislation drafting requires a lot of careful attention. The Japanese government deals with this task by means of thorough legislation drafting rules and final inspection by the Cabinet Legislation Bureau.

The drafting rules regulate document structures, orthography, and phraseology of statutes. These rules have been utilized for more than 100 years and are published as legislation manuals (e.g. [6]). Among the drafting rules, it is a noteworthy feature that they explicitly define distinct usage and meaning to many legal terms that look mutually similar. For example, the three Japanese words “者 $_{(a)}$,” “物 $_{(b)}$,” and “もの $_{(c)}$” are all pronounced *mono*. The Japanese legislation drafting rules prescribe that 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. Phrases in Fig. 1 contain each legal term.

Using the drafting rules, legislative officers in the Cabinet Legislation Bureau strictly inspect legislative bills which 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 appear vaguely or mistakenly in inspected bills. However, these inspections are still conducted mainly by human experts in legislation and that requires deep knowledge and an enormous amount of labor. Furthermore, according to Enami [4], this legislative work has become even tougher because of recent increased enactment of statutes.

Considering the above, we propose a method that assists legislation officers in finding inappropriate legal terms in a draft and offers correction ideas. By regarding a set of similar legal terms as a set of choices, we handle the legal term correction as a special

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

【a production in which thoughts or sentiments are creatively expressed】

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

Figure 1. Phrases with a legal term (underlined)

case of the multiple-choice sentence completion test. Although language models are typically used for the general multiple-choice sentence completion test (e.g. [5,10,12,13,15]), we apply Random Forest classifiers [2] to our method. Each classifier in our method is trained and optimized for a single set of similar legal terms. We assume that this term-specializing approach brings better performance.

This paper contributes to the legal term correction task by formally defining its problem, proposing a Random Forest-based method for the problem, and showing the performance of our method compared with existing language models.

2. Japanese Legal Terms

As described in Section 1, the Japanese legislation drafting rules define a number of sets of similar legal terms and each usage. The list below displays some examples:

- "直ちに $_{(d)}$" (*tadachini*), "速やかに $_{(e)}$" (*sumiyakani*),
  and "遅滞なく $_{(f)}$" (*chitainaku*)
  These are adverbs and share the concept of "speedily." In Japanese statutory sentences, these words express different degrees of speed: (d), (e), and (f) express most, moderately, and least speedy, respectively. No such strict difference among them exists in general Japanese sentences. According to the Standard Legal Terms Dictionary [14], (d), (e), and (f) should be translated to "immediately," "promptly," and "without delay," respectively.
- "前項 の 場合 に おいて $_{(g)}$" (*zenko no baai ni oite*)
  and "前項 に 規定する 場合 に おいて $_{(h)}$" (*zenko ni kiteisuru baai ni oite*)
  Both of these phrases behave as conjunctive and share the concept of "mentioning the preceding paragraph." In Japanese statutory sentences, (g) is used to mention the whole paragraph, while (h) is used to mention only the condition prescribed in the paragraph. According to the dictionary, (g) and (h) 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.

3. Related Work

Since we regard legal term correction as a special case of the multiple-choice sentence completion test, we explain the test in Section 3.1. Then, we mention several studies on language models for solving the test in Section 3.2. In Section 3.3, we introduce Random Forest [2], which we utilize instead of language models in our problem.

3.1. Multiple-choice Sentence Completion Test

In the general multiple-choice sentence completion test, a sentence with a blank and choices to fill in the blank are given. The statement below represents a typical example of the test:

He is ______ at the scoreboard.
(A) look (B) looks (C) looking (D) looked

One must choose the best option for filling in the blank "______" (in this case, (C)). The combination of choices can vary diversely depending on the situation. For example, the sentence with a blank below can be associated with the following choices to examine verb usage.

He is ______ at the scoreboard.
(A) looking (B) watching (C) seeing

Therefore, a method for this problem has to cope with any combination of choices.

3.2. Language Models

In the previous situation, language models are useful because they predict a word from the whole vocabulary. To evaluate language models, Zweig and Burges [16] 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 [16] evaluated n-gram models by their dataset. Most powerful language models evaluated by this dataset have a neural network architecture, which overcomes the curse of dimension by treating each word and each sequence of words as vectors [1]. For instance, Mikolov et al. [10] proposed two neural language models: Continuous Bag-of-Words Model (CBOW) and Continuous Skip-gram Model (Skipgram). Mnih and Kavukcuoglu [12] proposed the vector Log-bilinear model (vLBL) and ivLBL. Mori et al. [13] proposed vLBL(c) and vLBL+vLBL(c), which are improved models of vLBL so that they are sensible of relative positions of words adjacent to the target word. Mirowski and Vlachos [11] proposed a recurrent neural network (RNN) [3,7] language model by incorporating the syntactic dependencies of a sentence.

While most studies propose neural language models, some propose non-neural language models. Gubbins and Vlachos [5] proposed an n-gram-like language model that handles dependency trees. Woods [15] proposed a novel method based on pointwise mutual information.

Algorithm 1 Algorithm that solves our problem

```
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
    $W_l \leftarrow w_1\ w_2 \cdots w_{i-1}$
    $W_r \leftarrow w_{j+1}\ w_{j+2} \cdots w_{|W|}$
    $t_{\text{best}} \leftarrow \underset{t' \in T}{\arg\max}\, \text{score}(W_l, t', W_r)$
    if $t \neq t_{\text{best}}$ then
      Suggests $\leftarrow$ Suggests $\cup$ { a suggestion that $t$ in position $(i, j)$ should be replaced into $t_{\text{best}}$}
    end if
  end for
```

3.3. Random Forest

Random Forest [2] is a kind of machine-learning algorithm for classification.

It learns the training data by building a set of decision trees, which is also called a random forest.[1] A decision tree is conceptually a suite of if-then rules. Then, a random forest predicts the class of the given data by taking a vote of 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 predicts unseen data better.

4. Proposed Method

In this section, we describe our proposed method for legal term correction. In Section 4.1, we formally define the legal term correction problem and a general algorithm for the problem. In Section 4.2, we regard our problem as a special case of the sentence completion test and compare it with the general one. In Section 4.3, we state the way to use Random Forest for our problem and the advantages of using it.

4.1. Definition

Our method inspects legal terms found in given statutory sentences, and outputs correction ideas for some legal terms that seem to be mistakenly used. We define this task as a problem below:

- 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 the vocabulary V, that is, a legal term $t \in T$ can be either a word or multiple words;
- One judges whether each legal term t found in W is adequate;
- If another legal term $t_{\text{best}} \in T$ $(t_{\text{best}} \neq t)$ seems more adequate in the context, one suggests an idea that t_{best} should be placed instead of t.

We define a general algorithm for this problem in Algorithm 1, where $\text{score}(W_l, t, W_r)$ is any scoring function that calculates the likelihood of the term 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:

[1] From here on, we call the algorithm “Random Forest” and a classifier “a random forest.”

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

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

Here, T is the legal term set mentioned in Section 1. In this case, the algorithm finds (c) from W, which is a legal term in T. Then, the algorithm processes two word sequences $W_l = \text{著作物を創作する}$ (*chosakubutu wo sosakusuru*; creating a work) and $W_r = \text{の保護}$ (*no hogo*; protection of). Using W_l and W_r, the algorithm calculates scores of each legal term by the following equations:

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

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

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

Each calculates the likelihood of "著作物 を 創作する 者 $_{\text{(a)}}$ の 保護" (Protection of a person that creates a work), "著作物 を 創作する 物 $_{\text{(b)}}$ の 保護" (Protection of a tangible object that creates a work), and "著作物 を 創作する もの $_{\text{(c)}}$ の 保護" (Protection of an abstract object that creates a work), respectively. The algorithm is highly expected to choose the first option and to output a suggestion that (c) in W should be replaced into (a).

4.2. Characteristics of Problem

We regard the problem defined in Section 4.1 as a kind of sentence completion test by introducing the following ideas:

- 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.

Our problem is different 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 a blank. For example, each term of the legal term set $T = \{\, \text{者}_{\text{(a)}}, \text{物}_{\text{(b)}}, \text{もの}_{\text{(c)}} \}$ appears tens of thousands times in a statutory sentence corpus of nearly 29 million words that is compiled from almost four thousand acts and cabinet orders in effect in Japan. This means that we can make several hundred thousand questions for the legal term set. On the other hand, 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 that Japanese legislation manuals mention. On the other hand, we may consider any combination of choices in the general multiple-choice sentence completion test, since there is no restriction of them.

4.3. Using Random Forests

Because of the characteristics described in the previous section, we apply Random Forest [2] to our problem. We utilize it as the scoring function $\mathrm{score}(W_l,t,W_r)$, which is calculated by the following equation:

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

where D 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 (actually 0 or 1) that d chooses t based on features $w_1,w_2,\ldots,w_N$. w_l^i and w_r^i are 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).

For example, we calculate the scoring function in Equation (3) by the following equation when $N=2$:

$$\begin{aligned} &\mathrm{score}\left(\begin{array}{cccccc} \textit{chosakubutsu} & \textit{wo} & \textit{sosakusuru} & \textit{mono} & \textit{no} & \textit{hogo} \\ \text{著作物} & \text{を} & \text{創作する} & \text{者}_{\text{(a)}} & \text{の} & \text{保護} \\ \text{work} & \text{ACC} & \text{create} & \text{, person} & \text{, of} & \text{protection} \end{array}\right) \\ &= \sum_{d\in D} P_d \left(\begin{array}{c|ccccc} \textit{mono} & \textit{wo} & \textit{sosakusuru} & \textit{no} & \textit{hogo} \\ \text{者}_{\text{(a)}} & \text{を} & \text{創作する} & \text{の} & \text{保護} \\ \text{person} & \text{ACC ,} & \text{create} & \text{, of} & \text{, protection} \end{array}\right). \end{aligned} \tag{7}$$

Our method treats each legal term as a class. Thus, it builds a random forest for each set of legal terms.

We use random forests for three reasons. First, Random Forest classifiers for each legal term set learn from different datasets, and thus they can optimize their parameters for each set. In particular, it is useful to adjust the window size per legal term set because the distance between a legal term and its clue word can vary per legal term. On the other hand, language models learn from a single integrated dataset, and thus they use the same parameters throughout the legal terms. Second, Random forests can predict multiple-word legal terms because they equally handle any legal term as a single class. On the other hand, language models predict only a single word by given words. Therefore, we need some technique like word concatenation to predict multiple-word legal terms using a language model. Third, decision trees with naive if-then rules seem to be sufficient to predict legal terms because statutory sentences are rather more formal than general sentences since their orthography and phraseology are thoroughly regulated by the legislation drafting rules.

5. Experiment

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

5.1. Outline of Experiment

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

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

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 622,527, the total number of tokens is 28,816,368, and the total number of different words is 23,236.

We defined 26 legal term sets by referencing the Japanese legislation manual [6]. Table 1 shows some examples of legal term sets. English translations in this table are taken from the Standard Legal Terms Dictionary (March 2018 edition) [14] provided by the Ministry of Justice, Japan, except for items with an asterisk.

We compared the following models with Random Forest [2]: CBOW [10], Skipgram [10], vLBL [12], vLBL(c) [13], vLBL+vLBL(c) [13], and n-gram. As 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: dimension of vectors 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), optimization function is Adam [9]. We implemented, trained and tested the models by Chainer (v.1.7.0). As for the n-gram model, we used Katz's backoff trigram and 4-gram [8] in reference to Zweig and Burges [16].

We prepared two experiment designs for Random Forest: (1) setting the window size to 5—the same as the neural methods and (2) using a variable number $\{2, 5, 10, 15\}$ of window size that is suitable for each legal term set. The window size in the latter is determined by five-fold cross validation. From here on, we call the former method "Random Forest (fixed)" and the latter method "Random Forest (variable)." In both methods, we used the Gini coefficient to build decision trees, and optimized the number of decision trees $\{10, 50, 100, 500\}$, the maximum depth of each tree $\{10, 100, 1000, \text{unlimited}\}$, and the window size (in case of (2)) by five-fold cross validation. Implementation, training, and testing are done by Scikit-learn (v.0.19.1).

Since neural language models and n-gram models are designed to predict a single word, we combined all legal terms with multiple words into single words by the longest match principle. After this operation, the total number of tokens in the corpus became 27,718,637. Also, we changed words that appear 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 word sequences.

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 598,522 sentences and 26,707,937 tokens in total. The test data has 199 documents with 24,005 sentences and 1,010,700 tokens in total. There are 166,959 legal terms appearing in the test data.

In the evaluation, we measured accuracy of predicting legal terms in two averages: micro average acc_{micro} and macro average by legal term set acc_{macro}.

5.2. Experimental Results

Table 2 shows the experimental results of each model. As a baseline, we calculated the micro and macro averages of accuracy in maximum likelihood estimation (MLE), in which the most frequent legal terms in the train data are always selected.

Random Forest-based methods achieved the best accuracy in both the micro and macro averages. On the other hand, Skipgram was inferior to MLE in the two averages. Both Random Forest-based methods have less than 1% of gap between the two averages, while any other method has more than 2.6% of gap between the two. This means that Random Forest predicts any legal term set with high accuracy.

Table 1. Examples of legal term sets

Legal Term	Pronunciation	Meaning	Count
者 $_{(a)}$	*mono*	natural or juristic person*	286,762
物 $_{(b)}$	*mono*	tangible object*	24,622
もの $_{(c)}$	*mono*	abstract object*	159,496
に 係る	*ni kakaru*	pertaining to*	102,924
に 関する	*ni kansuru*	regarding*	76,405
に 関係する	*ni kankeisuru*	regarding*	80
直ちに $_{(d)}$	*tadachini*	immediately	2,293
速やかに $_{(e)}$	*sumiyakani*	promptly	1,947
遅滞なく $_{(f)}$	*chitainaku*	without delay	6,054
する こと が できる	*suru koto ga dekiru*	may	26,337
しなければ ならない	*shinakereba naranai*	must, shall	38,864
する もの と する	*suru mono to suru*	is to	8,976

6. Discussion

In this section, we investigate the experimental results in more detail to reveal the characteristics and effectiveness of our Random Forest-based methods.

First, we decompose the experimental results per part-of-speech (POS) in order to determine whether our method is good at predicting any POS of legal terms. Table 3 shows micro averages of accuracy per POS: noun, modifier, verb, and conjunction. In Table 3, Random Forest (variable) achieved the highest accuracy in nominal, verbal and conjunctional legal terms, and Random Forest (fixed) achieved the best in nominal. On the other hand, vLBL+vLBL(c) achieved the best in modifier legal terms.

It is an interesting tendency that while vLBL(c) and vLBL+vLBL(c) compare favorably with Random Forest in accuracy of nominal and modifier legal terms, they achieved worse accuracy for verbs and conjunctions. Specifically, vLBL(c) achieved 5.0% and 8.0% worse accuracy for verbs and conjunctions than Random Forest (variable), respectively, and vLBL+vLBL(c) did 6.1% and 10.8% for the same groups.

To reveal a cause for these results, we look into the accuracy per legal term. Table 4 shows the accuracy of conjunctional legal terms in vLBL+vLBL(c), Random Forest (fixed), and Random Forest (variable). Each method is denoted by “vLBL+,” “RF

Table 2. Experimental results

Method	acc_{micro}	acc_{macro}
Random Forest (fixed) [proposed]	91.8%	**91.0%**
Random Forest (variable) [proposed]	**91.9%**	**91.0%**
CBOW	86.9%	82.6%
Skipgram	72.5%	65.0%
vLBL	78.8%	76.1%
vLBL(c)	90.2%	83.0%
vLBL+vLBL(c)	90.1%	85.1%
Backoff trigram	84.9%	83.8%
Backoff 4-gram	86.6%	85.5%
MLE (baseline)	76.5%	66.4%

Table 3. Accuracy per POS

Method	Nominal	Modifier	Verbal	Conjunctional
Random Forest (fixed)	**92.7%**	92.0%	94.3%	87.9%
Random Forest (variable)	**92.7%**	92.1%	**94.7%**	**88.2%**
CBOW	87.4%	87.9%	82.9%	83.7%
Skipgram	77.0%	75.3%	51.5%	64.3%
vLBL	82.1%	78.4%	80.9%	73.6%
vLBL(c)	91.4%	92.2%	89.7%	80.2%
vLBL+vLBL(c)	92.1%	**92.4%**	88.6%	77.4%
Backoff trigram	83.3%	86.1%	89.1%	81.1%
Backoff 4-gram	86.6%	87.2%	91.3%	81.9%
MLE (baseline)	63.3%	82.5%	70.1%	78.5%

Table 4. Accuracy of conjunctional legal terms

No.	Legal term (Pronunciation; meaning)	Count	vLBL+	RF (F)	RF (V)	WS
1	又は (*matawa*; or)	9,116	74.2%	97.4%	**98.9%**	10
	若しくは (*moshikuwa*; or)	2,425	**72.9%**	38.7%	33.9%	
	Total (micro average)	11,541	73.9%	85.0%	**85.2%**	
2	及び (*oyobi*; and)	6,523	79.2%	98.9%	**99.0%**	5
	並びに (*narabini*; and)	921	**87.4%**	46.3%	45.1%	
	Total (micro average)	7,444	80.2%	**92.3%**	**92.3%**	
3	その他の (*sonotano*; other)	1,299	85.1%	91.4%	**91.6%**	5
	その他 (*sonota*; other)	1,036	80.9%	**81.9%**	81.3%	
	Total (micro average)	2,335	83.0%	**87.2%**	87.0%	
4	前項 の 場合 に おいて (*zenko no baai ni oite*; in the case referred to in the preceding paragraph)	87	63.2%	**100.0%**	98.9%	15
	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)	7	**57.1%**	28.6%	42.9%	
	Total (micro average)	94	62.8%	**94.7%**	**94.7%**	
5	ただし (*tadashi*; provided, however, that ···)	725	82.1%	91.4%	**96.4%**	15
	この 場合 に おいて (*kono baai ni oite*; in this case)	466	**85.8%**	82.8%	84.8%	
	Total (micro average)	1,191	83.5%	89.2%	**91.9%**	

(F),” and “RF (V)” in Table 4. In this table, “Count” means the number of the legal terms appearing in the test data, and “WS” means the optimized window size.

According to the table, Random Forest (variable) outperformed Random Forest (fixed) and vLBL+vLBL(c) in legal term set 5. Here, Random Forest (variable) set the window size to 15 for the legal term set. From this fact, we assume that Random Forest (fixed) and vLBL+vLBL(c) utilized insufficient context for the legal term set, while Random Forest (variable) could choose optimal context length. However, Random Forest has a tendency to choose frequent legal terms. For example, according to the table, it prefers to choose major legal terms “又は” (*matawa*; or) and “及び” (*oyobi*; and) from legal term sets 1 and 2, respectively.

7. Summary

In this paper, we proposed a legal term correction method in Japanese statutory sentences, focusing on sets of similar legal terms whose usages are defined in the legislation drafting rules. We regarded this legal term correction as a special case of the sentence completion test with multiple choices, considering a set of similar legal terms as the choices. Our method uses Random Forest classifiers [2], each of which is optimized for a set of similar legal terms. Our experiment has shown that our method outperformed existing modern methods for word prediction using neural language models.

In future work, we aim to improve performance by resolving the problem of frequent term preference by introducing techniques for biased datasets.

Acknowledgments

This work was partly supported by JSPS KAKENHI Grant Numbers 18H03492 and 17K00460.

References

[1] Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of Machine Learning Research 3, 1137–1155 (2003)

[2] Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)

[3] Elman, J.L.: Finding structure in time. Cognitive Science 14(2), 179–211 (2003)

[4] Enami, T.: Rippobakuhatsu to opun gabamento ni kansuru kenkyu — horeibunsyo niokeru “opun kodingu” no teian —. Tech. rep., Fujitsu Research Institute (2015)

[5] Gubbins, J., Vlachos, A.: Dependency language models for sentence completion. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1405–1410 (2013)

[6] Hoseishitsumu-kenkyukai: Shintei wakubukku hoseishitsumu (2nd edition). Gyosei (2018) (In Japanese)

[7] Jordan, M.I.: Serial order: a parallel distributed processing approach. Tech. Rep. ICS Report 8604, Institute for Cognitive Science, University of California. 39 pages (1986)

[8] Katz, S.M.: Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Transactions on Acoustics, Speech, and Signal Processing 35(3), pp. 400–401 (1987)

[9] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations. 15 pages (2015)

[10] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: International Conference on Learning Representations. 12 pages (2013)

[11] Mirowski, P., Vlachos, A.: Dependency recurrent neural language models for sentence completion. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics. pp. 511–517 (2015)

[12] Mnih, A., Kavukcuoglu, K.: Learning word embeddings efficiently with noise-contrastive estimation. In: Proceedings of the Advances in Neural Information Processing Systems 26. pp. 2265–2273 (2013)

[13] Mori, K., Miwa, M., Sasaki, Y.: 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. pp. 760–763 (2015)

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

[15] Woods, A.M.: Exploiting linguistic features for sentence completion. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. pp. 438–442 (2016)

[16] Zweig, G., Burges, C.J.: The Microsoft Research sentence completion challenge. Tech. rep., Microsoft Research (2011)