Coordination Analysis and Term Correction for Statutory Sentences using Machine Learning

YAMAKOSHI Takahiro

Abstract

Laws constitute an essential infrastructure that sustains and improves human society. Statutes stipulate laws by natural language, and they are continuously updated as society changes. Since statutes prescribe the rights and duties of people, statutory sentences are not allowed to contain errors or inconsistencies. To keep high consistency, they are written in accordance with their specific wordings, technical terms, and sentence structures. Therefore, we need to have sufficient knowledge and experience to write and understand statutory sentences appropriately.

In the case of Japan, a number of writing rules and customs have been established since the Meiji era through legislation (works of interpreting, producing, and updating statutes). Furthermore, the Japanese government has legislation bureaus that strictly examine draft bills whether they are written in concordance with the legislation rules. For these two reasons, it is quite important to follow the legislation rules when we handle Japanese statutory sentences, which may burden legislation officers with comprehensive and strict writing. Another point to be noted in Japanese statutory sentences is that they tend to be quite long and complex. One big factor for this characteristic is coordinate structures in Japanese statutory sentences. A coordinate structure is a sentence structure that enumerates multiple things in parallel. In Japanese statutory sentences, such coordination often appears with hierarchy, that is, a coordinate structure contains other coordinate structures inside. The legislation rules stipulate the hierarchy of coordinate structures; in other words, we need to mind the rules when we understand the coordination of statutory sentences. Overall, we identify two subjects that will be obstacles in handling Japanese statutory sentences: strict compliance with legislation rules and complex hierarchical coordinate structures.

In this thesis, we study two themes to provide solutions for the two subjects: coordination analysis and legal term correction. Coordination analysis identifies scopes of conjuncts (phrases in parallel) in a given sentence. With this information, we can simplify long and complex statutory sentences, which supports any person and system that has trouble understanding such statutory sentences. Thus, we place this study on a quite fundamental one for further sentence processes. The second theme, legal term correction, is located to a practical study that aims at drafting statutory sentences. The legislation rules define distinct usage of certain similar legal terms, which should be fulfilled in drafting statutory sentences. Our legal term correction finds misused legal terms and offers correction ideas for them, that is, this is proofreading specialized in legal terms with distinct usage.

The approaches in this thesis are a combination of deterministic legislation rules and machine learning technologies. It is reasonable to import the Japanese legislation rules to the approaches as deterministic rules because these rules are well-established and strictly operated by the government. We then delegate decisions based on context to machine learning methods. Both the formation of coordinate structures and the use of legal terms depend on the context around them. Since the number of context patterns is enormous to cope with deterministic rules, we rely on machine learning methods that automatically learn contexts from training data.

This thesis consists of seven chapters. Chapter 1 is the introduction of this thesis, which begins with an explanation of the legislation and Japanese statutory sentences. After identifying our studies that solve issues in handling Japanese statutory sentences, we position them among their related studies.

In Chapter 2, we describe the knowledge and techniques that are the basis of our proposed methods in this thesis. First, we review the Japanese legislation rules, and then we dig into coordination and legal terms that are the subjects in this thesis.

Next, we look at language models and classifiers that are the core machine learning technologies in the approaches.

In Chapter 3, we describe the study for coordination analysis for Japanese statutory sentences. We first review the background of coordination analysis including issues in current situations. We then propose a coordination analysis method for Japanese statutory sentences by comparing an existing method for them. Our method deterministically identifies the hierarchy of coordination based on the Japanese legislation rules for hierarchical coordinate structures. On the other hand, it identifies the scopes of conjuncts that compose a coordinate structure by utilizing neural language models. Here, we introduce two assumptions on coordination that ensure the validity of conjunct scope candidates. The first assumption is the conjunct similarity, that is, two paired conjuncts have similar context. The second assumption is the conjunct interchangeability, that is, a sentence is still fluent even if we swap two paired conjuncts in its coordinate structure. We calculate scores of these two assumptions by neural language models that are aware of the context of the whole sentence. In addition, the models are trained with sequences of tokenized statutory sentences; In other words, we do not use coordination information for training. This enables us to realize a neural-based coordination analysis method for Japanese statutory sentences with limited training resources.

In Chapter 4, we describe the study for legal term correction for Japanese statutory sentences. As same as Chapter 3, we first review the background and needs of legal term correction. Since the legal term correction task has not been studied yet to the best of our knowledge, we first define this task, and then we consider its characteristics. Next, we propose two approaches for the legal term correction task. The first approach uses Random Forest classifiers, which assigns a trained Random Forest classifier to each legal term set. Here, each classifier is optimized by its corresponding legal term set, and thus high prediction performance is expected. Furthermore, we earn knowledge on legal term correction from optimized parameters and feature importances calculated in the training. The second approach uses a BERT classifier, where we aim to achieve further good prediction performance by utilizing the wider context capability from the self-attention mechanism and the enormous knowledge earned by pretraining. Here, we introduce a problem of two-level infrequency in the legal term correction task and a solution for it.

In Chapter 5, we attempt to apply the legal term correction methodology established in Chapter 4 to foreign statutes, namely Thai statutes. It is a global issue that statutory sentences should be written appropriately. Here, Thai legislation has rules on the usage of similar legal terms, which is the same as Japan. On the other hand, usage of Thai legal terms tends to be bound by outside-sentence contexts such as genre and year. Also, Thai legal terms sometimes appear with few adjacent words, which we do not normally observe in Japanese legal terms. Therefore, we apply additional features for Thai legal term correction to the Random Forest approach of the previous chapter.

In Chapter 6, we discuss the relationship between the studies and real-world data circulation from the viewpoints of the existence of data circulation in the studies and contributions that the studies bring.

In the final chapter, we summarize this thesis. We first organize discussions in the previous chapters and then we discuss future work and prospect of the studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we summarize the background and the importance of the studies in this thesis. In Section 1.1, we describe law and legislation, especially how they are used in Japan, and identify the current issues of the legislation process. In Section 1.2, we position the two studies of this thesis among related studies. In Section 1.3, we describe the contributions of this thesis in academic fields and the real world. Section 1.4 explains its structure.

1.1 Introduction

Laws are the fundamental components of society. They illumine the exemplary ways of economic and social activities, which guarantee the health and cultural lives of citizens. Laws are continuously updated in accordance with the changes of economic situations, values, and technology. Updated laws are expected to provide a fairer environment and to increase the material comfort of citizens. For example, the Japanese Civil Code provided a responsibility described as “瑕疵担保責任” (defect liability), which was amended to “契約不適合責任” (non-conformity liability) in May 2020. Here, defect liability denotes liability for a hidden crack that spoils an item’s value (e.g., a roof leak), and non-conformity liability denotes one for a state where the item itself contradicts the specifications described in the contract. That is, judging non-conformity liability is easier and more systematic, which accelerates rapid and infallible trades.

Statutes are laws written as documents, which are continuously revised in accordance with societal changes. Since statutes bind people by rights and duties, statutes must be described strictly so that they can be correctly understood. Statutes must be completely devoid of errors or inconsistencies. To meet this requirement, the language with which statutes (**statutory sentences**) are composed often contain specific wordings, technical terms, and sentence structures that are rarely used in daily written language. Overall, for understanding statutes, large amounts of knowledge and experience are required to interpret such description rules of statutory sentences.

Japan’s current statutory law system, which was established in the Meiji era [1], has basically remained unchanged for over 100 years. Japan established its legislative (interpreting, producing, and updating statutes) technique for the precise and uniform drafting, enacting, and modifying of statutes. This technique includes a number of concrete customs and rules (hereinafter **legislation rules**) stipulated by various workbooks [35, 37].

In its first layer, the legislation rules stipulate that a statute structure must be defined as a document. Here are some examples of such rules. A statute is required to have a title and must be divided into main and supplementary provisions. Main provisions should generally be divided into articles.

Along with statute structure definitions, the technique enacts rules for forming, modifying, and deleting particular sentences, figures, and tables. In addition to such editing guidelines, rules exist on the usage of *kanji* and *kana*. Furthermore, they define the distinct usage of similar legal terms. For example, three Japanese words,

[1]Japan established its law system by referencing Occidental countries’ law systems in this era after its opening to the world.

"者$_{(a)}$," "物$_{(b)}$," and "もの$_{(c)}$," are all pronounced *mono* and share the concept of "object." Term (a) only means a natural or a juristic person, term (b) only means a tangible object that is not a natural or a juristic person, and term (c) only denotes an abstract object or a complex of such objects. In ordinary Japanese written language, unlike in statutory sentences, term (c) can refer to "者" and "物". That is, we can use term (c) as "著作物 (work) を (ACC[2]) 創作する (create) もの (*mono*)" to express "a person who creates a work" in ordinary Japanese. On the other hand, this usage is prohibited in Japanese statutory sentences due to the above rules; therefore, we have no choice but to use term (a) in this situation.

One more example of such legal terms is "及び$_{(d)}$" (*oyobi*) and "並びに$_{(e)}$" (*narabini*), which mean "and" and express the coordination of phrases. The legislation rules provide special directives for these legal terms to express the **hierarchy** of coordination. Term (d) is used for coordinations of the smallest hierarchy, and term (e) is used for the coordinations of the non-smallest hierarchy. For example, "イチゴ 及び ミカン の ケーキ 並びに チョコケーキ" (strawberry *oyobi* orange cake *narabini* chocolate cake) describes two cakes: a cake with strawberry and orange and another with chocolate. We can ignore an interpretation of three things in parallel: strawberry, orange cake, and chocolate cake, since this interpretation contradicts the coordination rules. We call such a collection of rules simply **legislation rules**.

These legislation rules are not only applicable for the central government. Local governments enact ordinances and orders using identical legislation rules. Furthermore, such private entities as companies also use the rules to enforce contracts, agreements, articles of incorporation, patents, and other legal documents. Mastering legislation rules is critical for those who handle any kind of legal documents.

Under this context of Japanese statutory sentences, we identify two issues for handling them. The first issue is the strict compliance with legislation rules on drafting. These legislation rules are comprehensive. Legislation bureaus scrutinize draft statutory sentences, which force legislation officers to completely obey the legislation rules in drafting a statute. That is, they need to be cognizant of the formalism of the statutory sentences in addition to their substantial contents.

Another issue is the appearance of long, complex statutory sentences originated by hierarchical coordinate structure, which complicates their understanding. As an extreme example, Figure 1.1 shows a very long statutory sentence that has 680 words: Those who attempt to read such long and complex statutory sentences are required to have sufficient experience and knowledge reading them or they will most likely fail to identify and understand their sentence structure. Machines that handle such

[2]Accusative case marker

sentences will also probably fail to adequately analyze their sentence structure because of their complexity, and analysis errors will cause deterioration of further processes.

Therefore, in this thesis, we establish solutions for these two issues. For the issue of hierarchical coordinate structure, we propose a novel coordination analysis method specialized for Japanese statutory sentences. With coordination analysis, we can simplify long and complex statutory sentences, which supports any person and system that has trouble understanding long and complex statutory sentences. Thus, this study can be regarded as a quite fundamental study for further sentence processes. For the issue of legislation rules on drafting, we establish a legal term correction methodology as a first step toward establishing the proofreading of comprehensive statutory sentences. Our legal term correction finds misused legal terms and offers correction ideas for them, which helps legislation officers to write statutory sentences consistent with the legislation rules. For the latter, we apply the methodology to foreign statutory sentences (Thai) to judge its overall effectiveness.

Hierarchical coordinate structure analysis and legal term correction are tough issues in handling Japanese statutory sentences. According to Yamamoto's [96] analysis, factors that impede smooth interpretation and drafting of Japanese statutory sentences are listed as follows:

1. Difficulty of legal terms;
2. Difficulty of the style of statutory sentences: Especially, there still exist statutory sentences written in the classical Japanese language;
3. Appearance of hierarchical coordinate structures;
4. Appearance of nested parentheses;
5. Need of application mutatis mutandis and its replacement.

Among them, factor 2. is fading because we no longer write statutory sentences in classical Japanese (excepting references) and the existing provisions are gradually revised to the modern Japanese. For factor 4., Ogawa et al. [68] introduce an assumption that sequences inside and outside of parentheses independently form a sentence. Based on this assumption, we can simplify nested parentheses by shallow text processing, that is, extracting and organizing them by text matching. For factor 5., it is rather easy to identify provisions to be applied mutatis mutandis, because referring expression in Japanese statutory sentences is firmly formalized [35]. The remaining factors, legal terms and hierarchical coordinate structures, are not easy to handle because they require decisions based on context.

第二条第一項第五号、第七号、第九号又は第十一号に掲げる有価証券（政令で定めるものを除く。）で金融商品取引所に上場されているもの、店頭売買有価証券又は取扱有価証券に該当するものその他の政令で定める有価証券の発行者（以下この条から第百六十六条まで及び第百六十七条の二第一項において「上場会社等」という。）の役員（投資信託及び投資法人に関する法律第二条第十二項に規定する投資法人である上場会社等（第百六十六条において「上場投資法人等」という。）の資産運用会社（同法第二条第二十一項に規定する資産運用会社をいう。第百六十六条において同じ。）の役員を含む。以下この条から第百六十五条までにおいて同じ。）及び主要株主（自己又は他人（仮設人を含む。）の名義をもつて総株主等の議決権の百分の十以上の議決権（取得又は保有の態様その他の事情を勘案して内閣府令で定めるものを除く。）を保有している株主をいう。以下この条から第百六十六条までにおいて同じ。）は、自己の計算において当該上場会社等の第二条第一項第五号、第七号、第九号若しくは第十一号に掲げる有価証券（政令で定めるものを除く。）その他の政令で定める有価証券（以下この条から第百六十六条までにおいて「特定有価証券」という。）又は当該上場会社等の特定有価証券に係るオプションを表示する同項第十九号に掲げる有価証券その他の政令で定める有価証券（以下この項において「関連有価証券」という。）に係る買付け等（特定有価証券又は関連有価証券（以下この条から第百六十六条まで、第百六十七条の二第一項、第百七十五条の二及び第百九十七条の二第十四号において「特定有価証券等」という。）の買付けその他の取引で政令で定めるものをいう。以下この条、次条及び第百六十五条の二において同じ。）又は売付け等（特定有価証券等の売付けその他の取引で政令で定めるものをいう。以下この条から第百六十五条の二までにおいて同じ。）をした場合（当該役員又は主要株主が委託者又は受益者である信託の受託者が当該上場会社等の特定有価証券等に係る買付け等又は売付け等をする場合であつて内閣府令で定める場合を含む。以下この条及び次条において同じ。）には、内閣府令で定めるところにより、その売買その他の取引（以下この項、次条及び第百六十五条の二において「売買等」という。）に関する報告書を売買等があつた日の属する月の翌月十五日までに、内閣総理大臣に提出しなければならない。

From the Financial Instruments and Exchange Act (Act No. 25 of 1948)

Figure 1.1: A statutory sentence with lots of words

In our solutions, we incorporate machine learning technologies with the legislation rules. Both coordination analysis and legal term correction require decisions based on the context around the analysis target. In the previous example on coordination analysis, we naturally think that "strawberry *oyobi* orange cake" is a more reasonable interpretation than "strawberry *oyobi* orange cake" for the coordinate structure of "*oyobi*." Here we may utilize intuitions that a strawberry and an orange are more related than a strawberry and a cake, and that both a strawberry cake and an orange cake are common. Thus, using context around the coordinate structure enables us to conclude this interpretation. In the previous example on legal term correction, we can determine that "person" (者 $_{(a)}$) is the most likely to fill in the blank in "a ____ who creates a work" by observing the context around the blank such as that the word will be the subject of a relative clause "A ____ creates a work." However, it is not reasonable to list up all possible contexts and decisions for them because the number of possible context patterns is innumerable. Therefore, we utilize machine learning technologies that will learn contexts automatically through examples.

1.2 Positioning of Study

In this section, we position our two studies in this thesis: coordination analysis and legal term correction in Sections 1.2.1 and 1.2.2, respectively.

1.2.1 Coordination Analysis

Coordination analysis is an auxiliary task of syntax parsing. We need to pay careful attention to coordination analysis because it is substantially ambiguous, and thus coordination analysis errors produce inaccurate parse results. Studies of coordination analysis can be categorized into two groups: general purpose coordination analysis and domain-specific coordination analysis.

General Purpose Coordination Analysis

Coordination analysis methods for general sentences can be grouped into two types based on their functionality: coordination analysis-specific methods and integrated methods that are combined with parsing. One example of a coordination analysis-specific method is work by Ficler and Goldberg [23], which proposed a neural-based coordination analysis method that utilizes a long short-term memory (LSTM) network [34]. Teranishi et al. [84] proposed another neural-based coordination analysis method, which utilizes bi-directional LSTMs instead of uni-directional LSTMs.

Some methods utilize formal grammars to handle hierarchy. Hara et al. [31] proposed a method that incorporates a context-free grammar (CFG) designed for coordination analysis. Their method evaluates coordination candidates by word alignment with vectorized word features and vectorized context features. Teranishi et al. [85] embedded their bi-directional LSTM-based coordination analysis into a CFG parser.

For integrated methods, Kurohashi and Nagao [48] proposed a rule-based coordination analysis method that is motivated by parsing improvement. Kawahara and Kurohashi [40] proposed a probabilistic coordination method that is integrated with parsing and case structure analysis. A characteristic of Kawahara and Kurohashi [41]'s other coordination analysis method is that it does not utilize similarity among conjuncts. Instead of similarity, it uses the likelihood of dependency between the head of each conjunct and the words outside the coordinate structure. Hanamoto et al. [30] took the CFG approach of Hara et al. and integrated it with a semantic parser based on head-driven phrase structure grammar (HPSG) [56].

Treebanks with thousands of parsed sentences such as the Penn Treebank [22] have been used for the development of general purpose coordinate analysis methods. Genia Treebank [43], which is a compilation of bio-medical text, is also widely used for building a general purpose coordination analysis method.

Domain-specific Coordination Analysis

We next focus on studies of domain-specific coordination analysis. Here are instances of such studies: Yokoyama [98] proposed a coordination analysis method for Japanese patent sentences and focused on their suffix characteristics; Roh et al. [72] proposed a method specialized for English patent documents using regular expressions.

For the following two reasons, we argue that domain-specific coordination analysis is also necessary for Japanese statutory sentences. First, since coordinate structures in Japanese statutory sentences have domain-specific legislation rules that prescribe hierarchical structures, coordination analysis methods for general sentences struggle to properly identify hierarchical coordinations. Second, such hierarchical coordinate structures often result in long, complex statutory sentences, whose huge computational cost triggers the failure of syntax parsing. A domain-specific coordination analysis method can reduce this failure by simplifying such statutory sentences in prior to syntax parsing.

For parsing Japanese statutory sentences, Ogawa et al. [68] designed a tag schema that copes with the characteristics of Japanese statutory sentences. Their schema conforms to the regulations [47] of the Kyoto University Text Corpus [42, 49]. They also compiled a corpus with 592 parsed sentences in accordance with their proposed schema. This corpus contains coordination information that reflects the coordination rules in Japanese statutory sentences.

To the best of our knowledge, there are no studies on a coordination analysis method for Japanese statutory sentences other than Matsuyama et al. [54] and Yamakoshi et al. [95]. The former is a deterministic method that utilizes the legislation rules on hierarchical coordination. It scores coordinate structure candidates with word alignment with simple word-level scoring rules. The latter (my bachelor's thesis) proposed a method that utilizes a context-free grammar (CFG) to list all possible candidates. It also uses a word-alignment-based scoring strategy. Unlike these methods, our novel coordination analysis method incorporates neural language models as scoring candidates to overcome the weakness of the word-alignment-based approach.

1.2.2 Term Correction

Legal term correction is a specified proofreading task that focuses on the usage of legal terms. Here, proofreading is a work to find and correct errors in a pre-publication document. The targets of proofreading studies can be divided into three: taxonomy, corpus, and method.

Taxonomy

A number of studies defined a taxonomy to categorize the proofreading targets. As for an early study, Faigley and Witte [20] defined a method for analyzing revisions from the viewpoint of how they affect texts. They divided revisions into surface changes, which do not introduce new information to a text or remove old information, and meaning changes, which add new context or delete existing context. Corrections in legal term correction belong to meaning changes because each legal term has its distinct meaning.

Recent studies on proofreading taxonomy focus on the categorization of errors that are subject to be revised [9, 11, 38, 66, 81]. Among these studies, we discuss the taxonomy from Bryant et al. [9]. They proposed a taxonomy with 25 error categories, which we can divide into these four types:

- Inadequate choice of vocabulary, which is categorized by part-of-speech: adjective (e.g., big → wide), adverb (e.g., speedily → quickly), conjunction (e.g., and → but) and so on.
- Inadequate choice of grammatical category, namely, adjective form (e.g., bigger → biggest), wrong noun inflection (e.g., informations → information), noun number (e.g., cat → cats) and so on.

- Superficial corrections such as contraction (e.g., n't → not), orthography (e.g., Bestfriend → best friend), and spelling (e.g., genectic → genetic).
- Miscellaneous: word order (e.g., only can → can only), other correctable error (e.g., paraphrasing such as "at his best" → "well"), and uncorrectable error.

Since the legislation rules define the usage of similar legal terms, corrections in legal term correction belong to the first type in this taxonomy.

Corpus

Some studies compile corpora for proofreading tasks. The CoNLL shared task on grammatical error correction [62, 63] is an influential dataset for proofreading. It contains revision information in multiple proofreading categories: articles, prepositions, number, verbs, and their agreement in the 2013 version [63] as well as pronouns, phrasing, sentence structure, punctuation, capitalization, and so on in the 2014 version [62]. Another recent example is the BEA-2019 Shared Task on Grammatical Error Correction from Bryant et al. [8] that adopts its own taxonomy [9]. Nagata et al. [60] compiled a corpus from English texts written by Japanese students. This corpus contains error information and part-of-speech information. Cheng and Nagase [12] compiled a corpus from Japanese technical documents written by Chinese native engineers. Nguyen and Miyao [64] proposed a corpus for professional writing assistance, which contains word alignment information and error information. Lichtarge et al. [51] proposed an approach for generating large parallel datasets by utilizing Wikipedia edit history. Napoles et al. [61]'s proofreading corpus is unique since it has both error information and fluency scores.

There are a number of corpora designed for binary classification that judges the adequacy of sentences. Daudaravicius et al. [14] made a dataset for a shared task for judging adequacy. Its texts were acquired from the revision history of VTex, a cloud LaTeXservice. Afrin and Litman [1] compiled a dataset for a proofreading support system for students. It contained sets of two sentences before and after proofreading and a label that indicates the effectiveness of the proofreading.

For other cases, Yannakoudakis et al. [97] proposed a corpus for sentence adequacy ranking. Zhang et al. [100]'s corpus includes three versions of documents and two pieces of revision information between each version.

Most proofreading corpora are English, especially written by students and/or non-native people. These corpora are not applicable to Japanese legal term correction because both the language and domain are different. Cheng and Nagase's one is an exception, which is from Japanese technical documents written by Chinese native engineers. Although this is a Japanese corpus, we do not believe that it is applicable to Japanese legal term correction because the domain difference still remains.

Method

We next focus on studies of proofreading methods. From the viewpoint of approaches that output proofread ideas, we divide the studies into three: a rule-based approach, a classifier approach, and a generative model approach. Methods with rule-based approaches include Takeda et al.'s CRITAC [82] that targets Japanese, Chae's system [10] that aims at Korean, and de Smedt and Rosén's system [16] that focuses on Norwegian. These methods handle misspellings and term unification. Cheng and Nagase [12] proposed an example-based proofreading method for offshore development.

Concerning methods with a classifier approach, for example, De Felice and Pulman [15] proposed a method for articles and prepositions, Sawai et al. [74] proposed a method for verb usage, and Utsubo [89] described their neural method for punctuation.

For the generative model approach, Hitomi et al. [33] proposed a sequence-to-sequence neural network model that simultaneously generates proofread sentences and their editing information. Another study [75] utilizes BERT [18], a multi-purpose neural language representation model, which suggests a methodology for grammatical error correction. Some studies focus on special situations. Ikeda et al. [36]'s character-based proofreading method normalizes informal expressions and slang for smoother processes. Faruqui [21] proposed a sequence-to-sequence model that outputs an inflected word from its origin form and inflection information.

Mudge [59] proposed a combination method for his proofreading software service that incorporates language models, dictionaries, and rule bases.

Besides methods that directly offer correction instructions, other methods support proofreading. Zhang and Litman [101, 102] proposed a method that predicts the intention of a given revision. Liu and Liu [52]'s method is a sequence labeling model that categorizes the necessity of revising each word.

Finally, we look at proofreading methods for statutory sentences. A method proposed by Sugisaki [79] detects complex sentence structures in German statutory sentences using a binary classifier. Template-based document generation [17, 28] enables legislators to draft well-structured statutes by filling in a template. These methods share the goal of proofreading that is to make sentences correct because they prevent legislators from drafting malformed statutes that are against the legislation rules. To the best of our knowledge, no other study has focused on proofreading legal terms selected based on the legislation rules.

We position our legal term correction methods as a classifier approach. Although the generative model approach is gaining popularity since there are sophisticated neural generative models and large-scale proofreading datasets, we solve legal term correction as a classifier problem because no such proofreading dataset exists for Japanese statutory sentences.

1.3 Contributions

In this section, we discuss the contributions of this thesis. First, we look at them from the viewpoint of academic fields and then from the viewpoint of practical fields regarding real-world data circulation of statutes.

1.3.1 Contributions towards Academia

Our studies explore the incorporation of deterministic legislation rules and data-driven machine learning technologies to establish methods for issues in legislation. Through the studies in this thesis, we demonstrate that we can incorporate legislation rules with modern data-driven machine learning technologies by forming constraints regarding the rules of the definition of tasks or the input for machine learning models.

We argue for the following contributions of our whole study to academia:

- Study on coordination analysis
  - We establish a coordination analysis method for Japanese statutory sentences that incorporates neural language models with domain-specific heuristics for hierarchy identification. This method overcomes the weaknesses of existing domain-specific methods and general neural-based methods. It can consider context outside conjuncts and is resistant to influence by the difference of the lengths of paired conjuncts, which the conventional domain-specific methods do not achieve. Also, it does not need coordination or syntax information for training the model, which is a strong point against neural-based methods.
  - We experimentally show the effectiveness of our method, which performs especially well for coordinate structures with imbalanced length conjuncts. The existing methods struggle to cope with them.
- Study on legal term correction
  - To the best of our knowledge, this study is the first attempt to establish Japanese legal term correction.

- We formally define the legal term correction task and position it as a special case of a sentence completion test with choices.
- We propose two approaches for Japanese legal term correction. The first is Random Forest classifier-based and the second is BERT classifier-based.
- We argue that the legal term correction task has a two-layer class imbalance problem. Our second approach solves this problem with three techniques.
- We experimentally show both the effectiveness and the characteristics of the two approaches.
- We apply the methodology to Thai statutory sentences by identifying the differences between Japanese and Thai legal terms regarding term correction. Then we propose a method optimized for Thai legal terms.

1.3.2 Contributions towards Real-World Data Circulation

The real-world data circulation is an interdisciplinary field on the utilization of real data. The core objective of the real-world data circulation is a sustainable improvement of the real world by circular data utilization, that is, data acquisition, data analysis, and data implementation.

Legislation itself can be regarded as the practice of real-world data circulation centered on statutes. More specifically, while writing and enacting legislation, we notice problems in society with which the current laws cannot cope. This is the data acquisition. We then analyze the problems to identify what to draft, which is the data analysis. Based on the analysis results, we draft statutes, which is the data implementation as documentation. Finally we promulgate the statutes throughout society, which is the data implementation as law. Although society will change in accordance with the new statutes, such societal changes may eventually cause new problems. We thus repeatedly enact legislation, and the statutes and society continue to mutually change.

Coordination analysis and legal term correction, our study's subjects, support this data circulation by accelerating the legislation process. Coordination analysis interprets long and complex statutory sentences to help both humans and machines understand them more quickly and more accurately. Our coordination analysis study provides a more sophisticated method for Japanese statutory sentences and contributes to their dissemination by offering more accurate interpretations of statutory sentences.

Legal term correction, which creates ideas for improving drafts, directly and indirectly supports the legislation process. The direct aspect accelerates the drafting process because legislation officers can quickly find inaccurately used legal terms. It also admonishes legislation officers for incorrect legal term usage. This process eventually raises the skills of legislation officers who will be able to legislate more effectively. That process is an indirect aspect of support. Our study for legal term correction contributes to such circulation by defining this task and proposing approaches for it.

In addition to the contribution of the data circulation of laws, each study can constitute other types of data circulation. For coordination analysis, we can establish a data circulation of annotated electronic statute data. With a coordination analysis method, we append annotations to statute data available on the internet. Annotated data can be used for other methods that utilize syntactic information. One use case is the compilation of a legal term thesaurus, enabled by the characteristics of coordinate structures that enumerate similar phrases. For legal term correction, for example, we can establish the data circulation of legislation knowledge using feedback about corrections from users. Here knowledge originates in both machines and humans. Machine knowledge is updated by retraining the system by feedback data. Human knowledge is updated by summarizing ideas about corrections that indicate how legislators mistakenly use legal terms. Knowledge improves the overall legislation environment.

Establishment of legislation support technologies (including methodology, training data, trained models, etc.) in this thesis may proceed to further establishment of legislation support technologies, and thus a data circulation of technologies will be formed. One possible scenario is the application of trained machine learning models to other tasks. For example, we will construct neural language models in the coordination analysis study, and a domain-adapted language representation model in the legal term correction study. These models can be utilized for tasks that require semantic comparison of statutory sentences such as document retrieval [25, 78, 99]. The updated technologies trigger development of novel systems that improve the legislation process, which is the social implementation of this data circulation.

1.4 Thesis Structure

In Chapter 2, we describe the basic knowledge and technologies for the studies in this thesis. We first review Japanese legislation rules, especially those on coordination and the usage of legal terms. We then describe the language models and the classifiers with which we handle context. For each module, we explain their ideas and established methods.

In Chapter 3, we discuss our coordination analysis study. First, we review the need for the coordination analysis of Japanese statutory sentences and the characteristics of coordination analysis in such sentences. Next we introduce our neural-based coordination analysis method by comparing it with conventional methods. Finally, we discuss the performance of our experimental results to find strengths and weaknesses of this proposed method.

In Chapter 4, we discuss a legal term correction study for Japanese statutory sentences. First, we review the background and importance of legal term correction in Japanese legislation. We formally define the task and propose an algorithm to solve it. Then we explain our two approaches for this task, the Random Forest approach and the BERT classifier approach, and focus on the backbone of each one. Finally, we discuss their performances based on the experimental results of each approach and identify the characteristics of each one with method-specific analysis measures.

In Chapter 5, we apply legal term correction methodology to Thai statutory sentences. First, we overview the legislation and legal term correction in them. Next we describe Thai legal terms and consider how to cope with them by examining their characteristics. After that we propose a method based on the Random Forest approach to which we apply some modifications. Finally, we discuss its performance on Thai statutory sentences.

In Chapter 6, we describe our achievements for the real-world data circulation of our studies. Finally, in Chapter 7, we conclude and discuss future work regarding our studies.

Chapter 2

Technical Background

In this chapter, we describe the knowledge and techniques that are the basis of our proposed methods in this thesis. In Section 2.1, we review the rules of Japanese legislation and delve into the coordinate structures and legal terms. In Section 2.2, we describe the machine learning technologies used in our proposed methods. We address language models in Section 2.2.1 and classifiers in Section 2.2.2.

2.1 Legislation Rules

In this section, we review the legislation rules established during the practice of Japanese legislation. First we overview the legislation rules in Section 2.1.1. We focus on coordination and legal terms in Sections 2.1.2 and 2.1.3, which are the subjects of this thesis.

2.1.1 Overview

As we discussed in Section 1.1, Japan established its inclusive legislation technique as a collection of legislation rules. From the viewpoint of regulation targets, Japanese legislation rules can be classified as follows:

- Rules for statute structure: These rules contain the definitions of the elements in a statute and their hierarchical relationships, such as: "An article consists of paragraphs; a paragraph contains a provision and optional items." They also stipulate the order of these elements. With these rules, a statute becomes a well-structured document.

- Rules for implementing and operating provisions: Here the legislation rules stipulate expression manners rather than jurisprudential matters. For example, the rules contain the ways to amend some elements, including parts, sections, articles, paragraphs, items, and the partial expressions of such structures.

- Rules for document format: These rules include indention and line breaks and stipulate the number of leading spaces for each statute structure. For example, a header for an article should have two leading spaces.

- Rules for wording: These rules define the orthography of *kanji*, *okurigana*, numbers, foreign words, etc. They also list preferred words for statutory sentences and define distinct usage for some similar legal terms, which is most critical in this thesis among the legislation rules.

2.1.2 Legal Terms

Japanese legislation drafting rules define various sets of similar legal terms and their usage. Although there is no official list of such legal terms in the form of regulations, a well-known legislation workbook [35] define the usage of 26 legal term sets, which is operated in legislation as the de facto standard. Among them, we focus on the legal term sets whose adequate use can be identified from their surrounding context. The list below illustrates three of the legal term sets that are our study objectives:

- "規定 $_{(f)}$" and "規程 $_{(g)}$" (both pronounced as *kitei*):
  Both terms (f) and (g) are nouns that share the concept of "rules." However, term (f) denotes a particular rule defined in a paragraph in a statute, and term (g) suggests a suite of rules defined in a statute. According to the Standard Legal Terms Dictionary [86], term (f) must be translated as "provision" and term (g) must be translated as "rules," "procedures," or "regulations."

- "直ちに $_{(h)}$" (*tadachini*), "速やかに $_{(i)}$" (*sumiyakani*), and "遅滞なく $_{(j)}$" (*chitainaku*):
  These adverbs share the concept of "quickly or promptly." In Japanese statutory sentences, they express different degrees of alacrity or instantaneousness: terms (h), (i), and (j) express descending levels. This strict difference does not exist in general Japanese writing. According to the dictionary, terms (h), (i), and (j) must respectively be translated as "immediately," "promptly," and "without delay."

- "前項 の 場合 に おいて $_{(k)}$" (*zenko no baai ni oite*)
  and "前項 に 規定する 場合 に おいて $_{(l)}$" (*zenko ni kiteisuru baai ni oite*):
  Both of these phrases are conjunctions and share the concept of "mentioning the preceding paragraph." In Japanese statutory sentences, term (k) refers to the entire paragraph, and term (l) mentions the condition prescribed in the paragraph. According to the dictionary, terms (k) and (l) must be translated as "in the case referred to in the preceding paragraph" and "in the case prescribed in the preceding paragraph."

We exclude the legal term sets whose terms express a range of the amount from our study objectives. This is because we usually cannot determine their adequate use from the context. For example, there is a legal term set of two terms "以下" (*ika*) and "未満" (*miman*), where the former is equivalent to "less and equal than" and the latter is "less than."

Note that legal terms have wide grammatical diversity; they can be any part of speech. In addition, there are phrasal legal terms like terms (k) and (l). The frequencies of legal terms vary largely both inside a legal term set and among such sets. For example, in our experimental dataset, although term (f) occurs 401,381 times, term (g) only occurs 4,139 times. The legal terms in legal term set {term (f), term (g)} occur 405,520 times, and those in set {term (k), term (l)} occur only 3,159 times.

In addition to what we discussed above, legislation rules define the legal terms of the coordinate conjunctions that indicate the construction of coordinate structures,

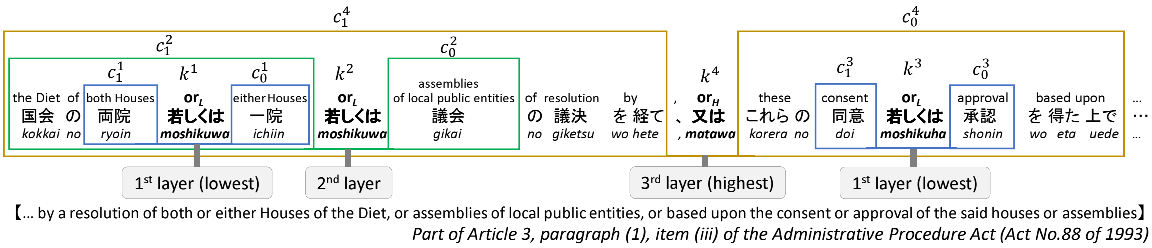

Figure 2.1: Japanese statutory sentence with "*matawa*" (or$_H$) and "*moshikuwa*" (or$_L$)

which we will describe in the next section.

2.1.3 Coordination

In this section, we explain the legislation rules for coordinate structures. The rules define the usage of two pairs of Japanese coordinators: "又は" (*matawa*) and "若しくは" (*moshikuwa*) that mean "or", and "及び" (*oyobi*) and "並びに" (*narabini*) that mean "and". They constitute hierarchical coordinate structures to disambiguate interpretations. The rules also define the usage of two coordinators, "その他" (*sonota*) and "その他の" (*sonotano*), whose meanings are very similar. Furthermore, they define another "and" coordinator "かつ" (*katsu*) to indicate specific nuances of meaning in the coordinate structure. In the final portion of this section, we review notable syntax restrictions specific to the coordination of Japanese statutory sentences.

"*matawa*" and "*moshikuwa*"

Both "*matawa*" and "*moshikuwa*" are generally translated as "or" in English. In general Japanese writing, these two coordinators are used interchangeably. However, in Japanese statutory sentences, "*matawa*" is strictly used for the highest coordinate structure, while "*moshikuwa*" is just used for other coordinate structures. Therefore, in this thesis, we translate "*matawa*" as "or$_H$" and "*moshikuwa*" as "or$_L$" in English to distinguish them in the context of coordination analysis. Figure 2.1 shows the usage of the two coordinators in a specific Japanese statutory sentence. A bunch of mutually connected boxes illustrates one coordinate structure. This figure also indicates conjuncts in the form of c^i_j, where i and j are their coordinate structure index and conjunct index, respectively, which we describe later in Section 3.2. k^i in this figure shows the coordinator of the i-th coordinate structure. Note that "*matawa*" is also used for "or" type coordinate structures without any connotation of hierarchy.

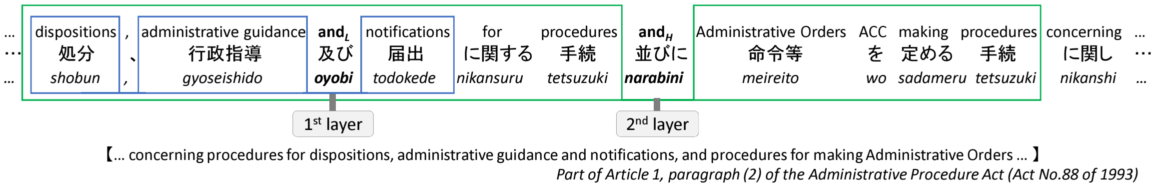

Figure 2.2: Japanese statutory sentence with "*oyobi*" (and$_L$) and "*narabini*" (and$_H$)

"*oyobi*" and "*narabini*"

Both "*oyobi*" and "*narabini*," which are generally translated as "and" in English, are also used interchangeably in general Japanese sentences. However, in Japanese statutory sentences, "*oyobi*" is strictly used for the lowest coordinate structure, and "*narabini*" is strictly used for other coordinate structures. In this thesis, we translate "*oyobi*" as "and$_L$" and "*narabini*" as "and$_H$" in English. Figure 2.2 illustrates the usage of the two coordinators. Note that "*oyobi*" is also used for "and" type coordinate structures without any connotation of hierarchy.

"*sonota*" and "*sonotano*"

Both "*sonota*" and "*sonotano*" are generally translated as "other," "any other," "such as," and so on in English, and they are also used interchangeably in general Japanese sentences. However, in Japanese statutory sentences, since these coordinators have different meanings, they represent the exact legal effects of statutory sentences. In this thesis, we translate "*sonota*" as "other$_1$" and "*sonotano*" as "other$_2$."

In view of the differences in meaning, the two coordinators constitute different types of coordinate structures [68]. The phrase followed by "*sonota*" (other$_1$) is a conjunct along with the preceding phrases. In Figure 2.3, the phrase "*hito no chikaku niyottewa ninshikisuru koto ga dekinai hoshiki*" (device unrecognizable to human senses) after coordinator "*sonota*" (other$_1$) is a conjunct along with "*denshiteki hoshiki*" (electronic device) and "*jikiteki hoshiki*" (magnetic device).

On the other hand, the phrase succeeding "*sonotano*" (other$_2$) is not a conjunct but just a hypernym of the preceding conjuncts. "*sonota*" in "*sonotano*" is also a conjunct. Here "*sonotano*" (other$_2$) is a compound word that consists of two morphemes: "*sonota*" (other) and "*no*" (of). In Figure 2.4, phrase "*kinmujoken*" (working conditions) after coordinator "*sonotano*" (other$_2$) is a hypernym of the preceding three conjuncts: "*kyuyo*" (salaries), "*kinmujikan*" (working hours), and "*sonota*."

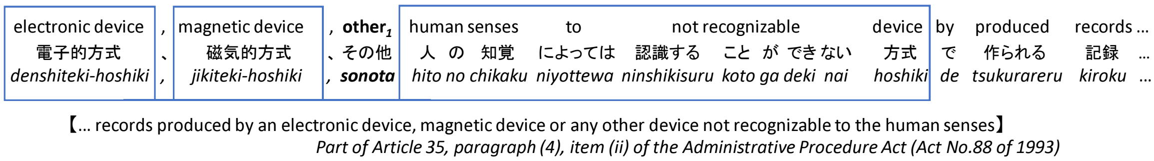

Figure 2.3: Japanese statutory sentence with "*sonota*" (other$_1$)

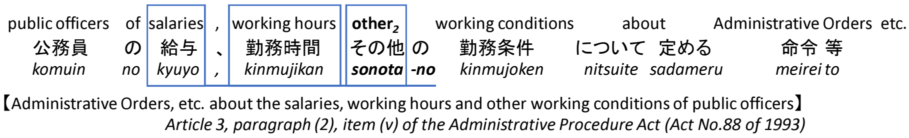

Figure 2.4: Japanese statutory sentence with "*sonotano*" (other$_2$)

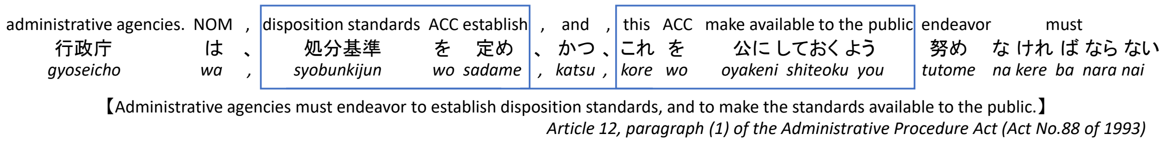

Figure 2.5: Japanese statutory sentence with "*katsu*" (and)

"*katsu*" (and)

"*Katsu*," which is translated as "and" in English, is identical to "*matawa*" and "*moshikuwa*." The difference between those coordinators is that "*katsu*" has a nuance that suggests the two conjuncts, which are connected by the coordinator, are semantically inseparable so that they only have one complete meaning after both are written. Since there is no hierarchical relationship between "*katsu*" and the other "and" coordinators, we simply translate "*katsu*" as "and." Figure 2.5 illustrates the usage of this coordinator.

Syntax of Coordinate Structure

Japanese legislation rules define the following syntax restrictions on coordinate structures.

- When connecting three or more conjuncts in a coordinate structure, the corresponding coordinator should be used only between the last two conjuncts, and Japanese commas (、) must be used between other conjuncts. The inner coordinate structure of Figure 2.2 shows this rule, which causes an important fact that every coordinate structure has at most one coordinator.

- When connecting verbal or adjective phrases, a comma should precede the coordinator. If "*katsu*" connects two verbal or adjective phrases, another comma must follow the coordinator in addition to the preceding comma. The coordinate structure of Figure 2.5 shows this rule.

2.2 Machine Learning Technologies

In this section, we describe the machine learning technologies utilized in our proposed methods. We address language models in Section 2.2.1 and classifiers in Section 2.2.2.

2.2.1 Language Model

A language model assigns the probability of generating sentence $W = w_1\ w_2 \dots w_{|W|}$. In many cases, a language model is designed to receive a word sequence with a fixed length as a context and outputs a probability distribution of a word that is to appear in the context. Therefore, to calculate a sentence's probability, we need to slice it into word sequences and gather the probabilities of each word sequence by the language model. For example, if a language model accepts two words as a context and outputs a word, we acquire the probability of W by chaining the probabilities of the sequences, such as $P(w_3|w_1, w_2) \cdot P(w_4|w_2, w_3) \cdot \dots \cdot P(w_{|W|}|w_{|W|-2}, w_{|W|-1})$.

In our research, we seek a way of utilizing language models for both coordination analysis and legal term correction. For coordination analysis, we use language models to evaluate the validity of coordinate structure candidates. For legal term correction, we use them to compare the likelihood of word sequences, each of which contains a candidate of similar legal terms. In both cases, we utilize language models to determine "correct word sequences."

In the following sections, we introduce three language model methods: an n-gram language model, a neural language model, and a general-purpose language representation model. The last is an **upwardly compatible** model, which can be used not only as a language model but also a sentence or word-level classifier.

n-gram Language Model

The n-gram language model, which is the simplest language model, predicts the next word of a given sequence with $n-1$ previous words. A simple implementation of the n-gram language model is to assign a relative frequency to each n-gram. That is, the probability of trigram "the black pen" $P(\text{pen}|\text{the}, \text{black})$ is $\frac{C(\text{the black pen})}{\sum_{w \in V} C(\text{the black } w)}$, where $C(w_1 w_2 \dots w_n)$ denotes the count of the n-gram $w_1 w_2 \dots w_n$ in the corpus. However, this implementation faces a critical problem with **zero frequency**. If a word sequence never appears in the corpus, a naive n-gram language model assigns such a sequence to probability zero. However, this treatment is inadequate because sentences containing zero-frequency sequences will be assigned zero probability, which naturally happens. Katz [39] solved this problem with a backoff mechanism, which interpolates probabilities of n-gram language models with different lengths.

Neural Language Model

A neural language model (NLM) works on a neural network [5]. Identical to an n-gram language model, it outputs a word's probability distribution that follows an input word sequence. Unlike an n-gram language model that calculates probabilities simply by counting the frequencies of n-grams, an NLM calculates probabilities by vector operations. It embeds input words into distributed representations (i.e., word vectors) and transforms these vectors to let them represent context from which it calculates a probability. Some studies prove that NLMs handle various linguistic properties. For example, word vectors connote a semantic relationship and the arithmetic operations of such vectors make a context drawn by words [55].

Many kinds of neural network architectures can be applied to a neural language model. Continuous Bag-of-words (CBOW) and Skipgram [55] are simple neural networks that have only three layers: embedding, context, and output. The behavior of CBOW and Skipgram is contrasted in terms of input and output. CBOW inputs a word sequence as a context and outputs a word probability in the context; Skipgram inputs a word and outputs a probability for the adjacent words. vLBL [57] and vLBL(c) [58] are derivatives of Skipgram, which assesses a word in the given context by calculating the similarity between the word vector and the context vector composed of the vectors of the context words.

Long short-term memory (LSTM) [34] is a neural network architecture with a high affinity for long sentences. A layer with LSTM cells constitutes a recurrent connection. That is, an LSTM cell uses the state of the previous step to determine the state of the current step. In natural language processing, a step corresponds to a word in a word sequence. Therefore, an LSTM layer processes a word in the word sequence by referring to the state at the previous word. This mechanism has a high affinity for sentences because the words in a sentence are dependent on each other. A typical LSTM cell has a constant error carousel (CEC), which retains the current value in steps, surrounded by three gates: input, forget, and output. This connection enables information to be propagated through distant steps. An LSTM neural network becomes a language model by allowing it to output a word probability for each input word. Figure 2.6 shows an example of an NLM constructed by an LSTM [80]. The four hidden layers with recurrent connections in themselves can

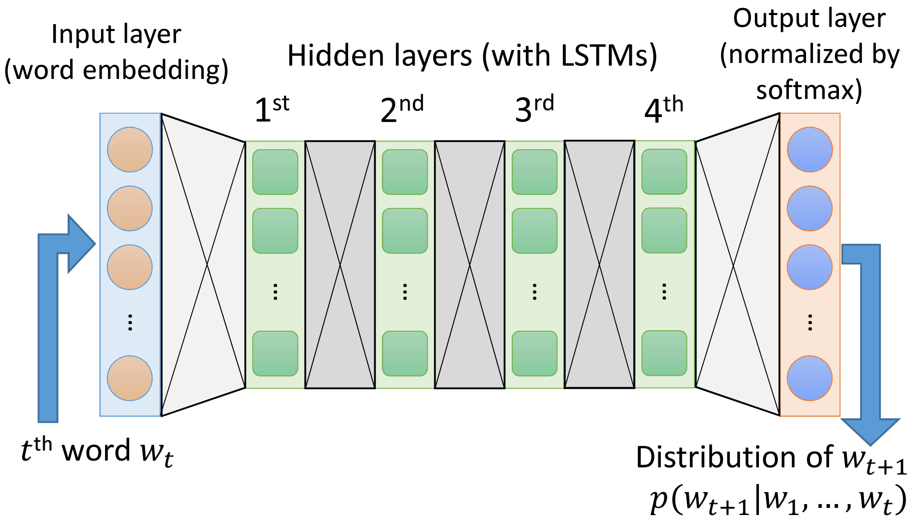

Figure 2.6: Example of LSTM-based neural language model

contain the context of the input word sequence.

General Purpose Language Representation Model

Such general-purpose language representation models as ELMo [70] and BERT [18] have recently attained remarkable performance in many natural language processing (NLP) tasks, including question answering [71] and language inference [93]. This good performance is due to two common key points. First, their word representations are pretrained using a huge amount of texts. Second, they are designed to be diverted to various NLP tasks by inheriting pretrained representations and attaching input and output modules to each target task.

Here we describe the architecture of BERT, which we used in the legal term correction task. Figure 2.7 shows the connection of a BERT model in the pretraining phase. BERT's main component is Transformer [90], which is a highly sophisticated neural network architecture. A BERT model has a number of Transformer layers whose units are mutually connected in the form of self-attention. The self-attention connection captures the relationship between input words by transforming the word vector into query, key, and value vectors. It calculates the inner products of a query vector with every key vector in the input sequence, and thus we get the weights of each word with the query vector's word. The weights show the relationship of the queried word and other words. It equally treats each word pair regardless of its distance; therefore, we usually add positional encodings to word vectors to indicate the words' positions. With these weights, new vectors are made for the input words by weighted-summation of the value vectors. The Transformer layers have a multiple number of

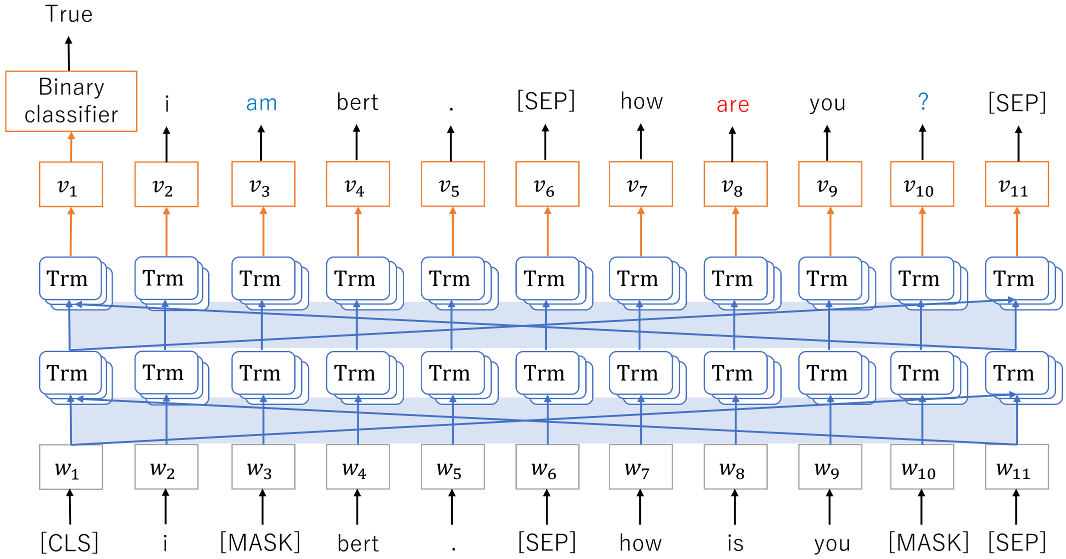

Figure 2.7: BERT model in pretraining

heads for this self-attention, which helps the model capture such different perspectives of linguistic knowledge as dependency and adjacent word information [91].

BERT pretraining is based on a multitask of the following two tasks: (1) masked language modeling and (2) next sentence prediction. In the masked language modeling task, the BERT model is required to answer accurate words of the input word sequence. In the authors' settings, 15% of the input words are subject to be predicted, where 80% of the selected words are covered with "[MASK]," 10% are replaced with a random word, and the remaining 10% are unchanged [18]. In the next sentence prediction task, the BERT model is required to answer whether a given sentence pair is continuous. 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 modeling: [you, i]
Label for next sentence prediction: True,
```

where "[CLS]" is the meta token for the sentence classification tasks, "[SEP]" is the separator of two sentences, and "[PAD]" is used for padding.

2.2.2 Classifiers

A classifier, in general, receives the features of an example and predicts the probability distribution of the classes to which the example may belong. In this study, we consider two classifiers in the legal term correction study: Random Forest [7] and BERT [18].

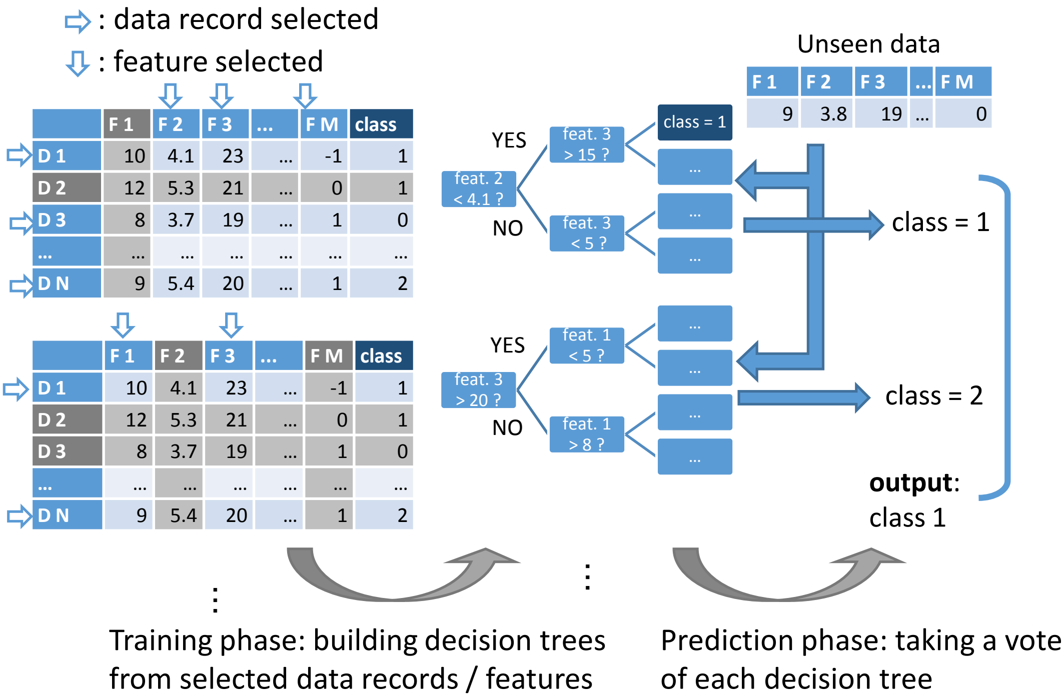

Figure 2.8: Procedure of Random Forest training and prediction

Random Forest

Random Forest [7] is a kind of ensemble machine learning method, which predicts classes by the integration of weak classifiers. Random Forest can be either a binary classifier or a multiclass classifier. For the legal term correction that we will discuss in Section 4, we use it as a multiclass classifier. We first explain the training and prediction processes of Random Forest. Figure 2.8 shows an overview of these processes.

In the training process, we built a Random Forest model by combining multiple decision trees that were separately trained. With the built model, we predict unseen examples by taking a majority of predictions from each decision tree. The trees in the middle of Figure 2.8 correspond to the decision trees. Each non-leaf node in a decision tree represents a condition, and one edge from the node is used when the condition is satisfied, and otherwise the other is used. A leaf node stands for one class, a decision of the tree.

One important Random Forest characteristic is to build decision trees using randomly sampled features and examples. With this technique, Random Forest performs equal to or better than other ensemble methods, such as Adaboost [24]. It is also robust to outliers and noises, and is easily parallelized [7]. One additional benefit of using Random Forest is that we can calculate feature importances by using out-of-bag

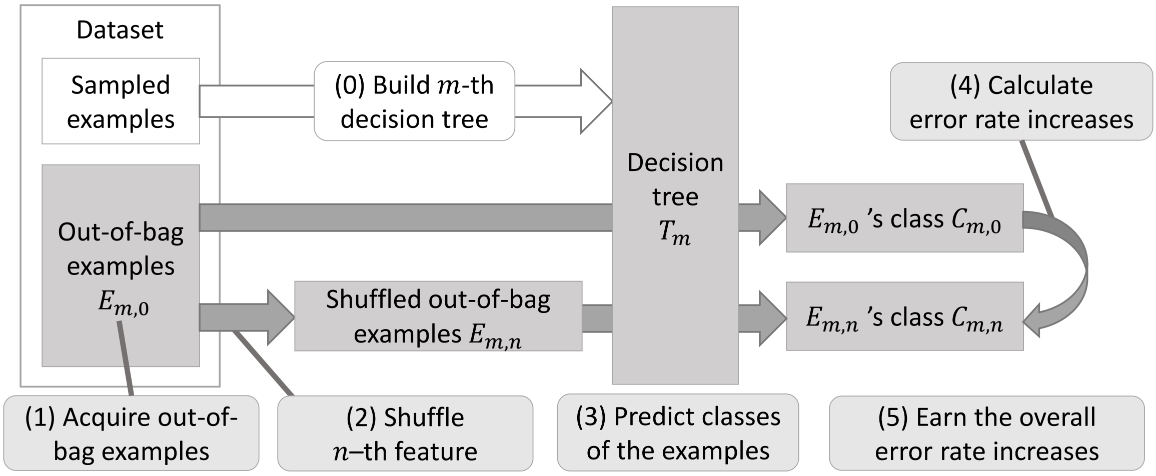

Figure 2.9: Procedure to calculate feature importances

training examples that are not used for training. The following shows a procedure for calculating such feature importances of the n-th feature.

1. Build m-th decision tree T_m by randomly sampled examples;
2. Acquire out-of-bag examples $E_{m,0}$ in T_m;
3. Compile set of examples $E_{m,n}$ where the n-th features in the examples are shuffled;
4. Using T_m, predict classes $C_{m,n}$ of shuffled examples $E_{m,n}$ and classes $C_{m,0}$ of original out-of-bag examples $E_{m,0}$;
5. Calculate the increase between error rates $a_{m,0}$ and $a_{m,n}$ that are calculated from the predictions of $C_{m,0}$ and $C_{m,n}$, ($a_{m,n}$ is usually somewhat bigger than $a_{m,0}$ since one feature was shuffled);
6. Earn overall error rate difference x_n by applying the above processes to all decision trees;
7. The n-th feature is important if x_n is big because shuffling the feature increased the prediction error.

Figure 2.9 depicts these processes.

BERT Classifier

We can build a BERT classifier by replacing certain output modules in Figure 2.7 with a classifier module for our desired classification task and finetuning it with its

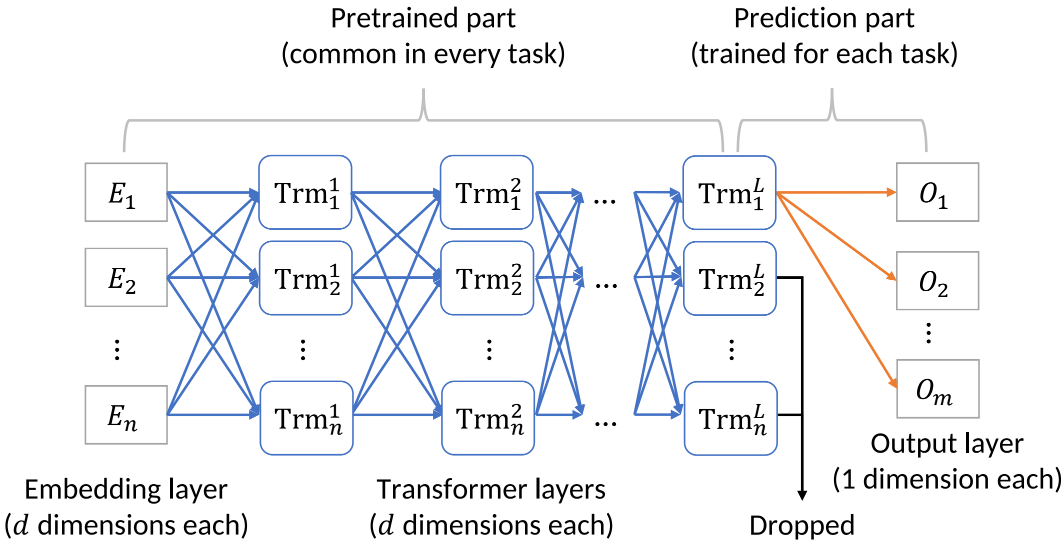

Figure 2.10: BERT classifier

training examples. In case of a sentence-level classification, we replace the binary classifier module with our classification module. In case of a word-level classification, we replace each word output module with our classification module. Figure 2.10 shows the construction of a BERT classifier. The BERT model inputs n words and outputs a probability distribution of m classes so the model is an m-class sentence-level classifier.

Chapter 3

Coordination Analysis

In this chapter, we describe the study for establishment of neural-based coordination analysis method for statutory sentences. First, we overview the background of coordination analysis in Section 3.1. In Section 3.2, we introduce a conventional method, from which our proposed method inherits the deterministic approach. In Section 3.3, we introduce the proposed method. In Section 3.4, we conduct experiments for the proposed method, and then we discuss the result in Section 3.5. Finally, we conclude this piece of study in Section 3.6.

3.1 Introduction

Japanese statutory sentences are not easy for non-experts to read. One reason for this difficulty is the frequent appearance of hierarchical coordinate structures that is supported by the Japanese legislation rules. For example, the sentence in Figure 2.1 in Section 2.1.3 contains four coordinate structures that compose a three-layered hierarchical coordinate structure. Such complex hierarchical coordinate structures also degrade the quality of automatic statutory document processing. Therefore, the development of a high-performance technique for coordination analysis is desired in statutory document processing, such as reading assistance with statutory sentences [94].

As we discussed in Section 1.2.1, various methods have been proposed for coordination analysis. However, we cannot expect these methods, especially those for general sentences, to work well for Japanese statutory sentences because they are not designed to consider the legislation rules. In fact, even a famous Japanese parser, KNP (v3.01) [40], could identify coordinate structures in Japanese statutory sentences at only 26 points of F-measure [54].

One existing method specialized for Japanese statutory sentences is from Matsuyama et al. [54]. This method deterministically identifies the hierarchy of coordinate structures and the scope of their conjuncts based on the legislation rules described in Section 2.1.3. Yamakoshi et al. [95] updated this method so that it utilizes a CFG parser for better hierarchy identification. However, these methods have a weak point in identifying conjuncts whose word length is very different from that of their adjacent conjuncts. This is because this method uses a scoring strategy based on one-to-one word alignment in identifying conjuncts. Also, these word alignment approaches do not consider context outside the coordinate structure, which may also degrade the performance.

From this background, we propose a new method for identifying hierarchical coordinate structures in Japanese statutory sentences using neural language models (NLMs) [5], especially Long-short term memory (LSTM) [34] based NLMs. Our method inherits the deterministic analysis strategy from Matsuyama et al.'s method. The difference is that ours uses LSTM-based NLMs instead of one-to-one word alignment for conjunct identification. We transform each conjunct candidate into a fixed-length vector so that our method can identify conjunct scopes without being affected by their length. Since our NLMs are trained by tokenized statutory sentences, our method does not rely on any annotated dataset such as Genia Treebank [43], which is friendly for a domain with limited resources such as Japanese statutory sentences.

3.2 Conventional Method for Japanese Statutory Sentences

In this section, we explain Matsuyama et al.'s coordination analysis method [54], from which we inherit the deterministic approach, describing the causes of the performance decrement that has been refined in our proposed method.

Figure 3.1 shows the processing flow of the conventional method. The method identifies all of the coordinate structures crd^i $(1 \leq i \leq N)$ in a sentence, where N is the number of coordinators included in the sentence.

The method assumes that any coordinate structure in Japanese statutory sentences can be formalized with the following extended regular expression:

$$crd^i = c^i_{n_i} \cdot \text{“、”} \cdot c^i_{n_i-1} \cdot \text{“、”} \cdot \ \ldots \ \cdot \text{“、”} \cdot c^i_1 \cdot \text{“、”?} \cdot \ k^i \cdot \text{“、”?} \cdot \ c^i_0, \tag{3.1}$$

where c^i_j $(0 \leq j \leq n_i,\ n_i \geq 1)$ is the j-th conjunct that constitutes crd^i, k^i represents the coordinator of crd^i, and "、" is a comma in Japanese. "·" represents the string concatenation operator, and "X?" means that "X" occurs at most once. Note that k^i occurs only between c^i_1 and c^i_0. Figure 2.1 also shows that each coordinate structure is expressed in accordance with Eq. (3.1).

The method identifies coordinate structures deterministically, that is, it identifies each conjunct and coordinate structure in a predetermined order and does not modify identified conjuncts or coordinate structures.

In the following subsections, we describe each process of the method in detail.

3.2.1 Coordinator Extraction and Ranking

All targeted coordinators shown in Table 3.1 are extracted from an input sentence. Two types of new coordinators "*to*" (and) and "*ya*" (and), which are also found in general Japanese sentences, have no specific usage in Japanese statutory sentences.

The analysis order of each of the extracted coordinators is decided based on two rules: (1) a coordinator with a smaller priority number in Table 3.1 comes first (Priority Rule), and (2) a coordinator that appears earlier in a sentence comes first among coordinators with the same priority (Position Rule).

3.2.2 Candidate Extraction

Candidates of a conjunct c^i_j are extracted as a set C^i_j. Although the way to extract them varies with the value of j, there are two common conditions independent of j.

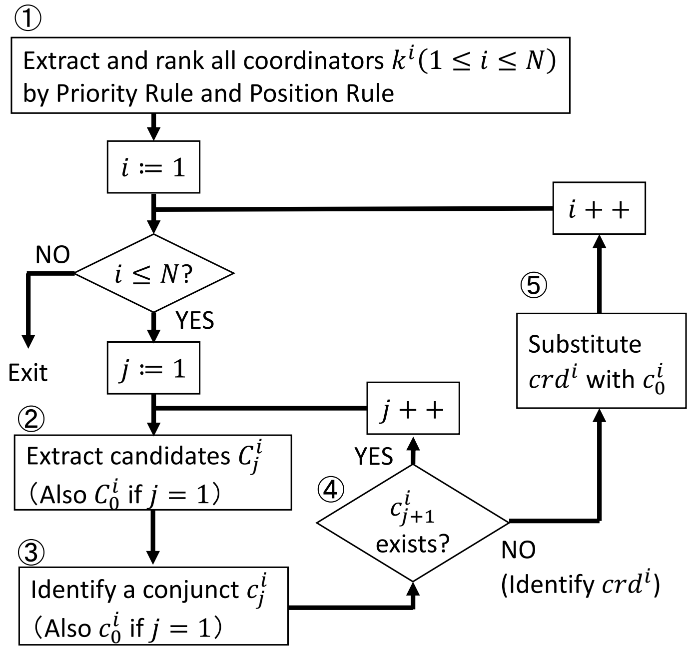

Figure 3.1: Processing flow of the conventional coordination analysis method

Table 3.1: Coordinators targeted in conventional method and their priority

Priority	Coordinator			
1	及び (*oyobi*; and$_L$)	若しくは (*moshikuwa*; or$_L$)		
2	並びに (*narabini*; and$_H$)	又は (*matawa*; or$_H$)		
3	その他 (*sonota*; other$_1$)	かつ (*katsu*; and)	と (*to*; and)	や (*ya*; and)

First, no candidate includes a comma or any unprocessed coordinator. Second, if a candidate includes any part of an already identified structure, the candidate has to include the whole of that structure.

In the case of C_1^i, the leftmost word of every candidate must be the leftmost word of a *bunsetsu*,[1] and the rightmost word must be a word (except a comma) just before the coordinator k^i.

In the case of C_0^i, the leftmost word of every candidate must be a word (except a comma) just after k^i. The rightmost word of every candidate must have the same POS (Part of Speech) as any one of the rightmost words of candidates of c_1^i and appear last in each *bunsetsu*. If the rightmost word is a noun, **Semantic Similarity** between the candidates of c_0^i and c_1^i is calculated based on a Japanese thesaurus [77],

[1] *Bunsetsu* is a linguistic unit in Japanese that roughly corresponds to a basic phrase in English. A *bunsetsu* consists of one independent word and zero or more ancillary words.

and the highest three candidates are chosen.

In the case of C^i_j $(j \geq 2)$, the leftmost word of every candidate must be a leftmost word of each *bunsetsu*, and the rightmost word must be a word (except a comma) before the leftmost word of c^i_{j-1}.

For example, in Figure 2.1, the two candidate sets C^1_1 and C^1_0 for the coordinator k^1 will be {"*kokkai no ryoin*" (both Houses of the Diet), "*ryoin*" (both Houses)} and {"*ichiin*" (either House)}, respectively.

3.2.3 Conjunct Identification

The two conjuncts c^i_1 and c^i_0 are identified simultaneously. Concretely, the method chooses a pair from $C^i_1 \times C^i_0$, which has the highest similarity between the two conjunct candidates. As for c^i_j $(j \geq 2)$, the similarity between each candidate in C^i_j and already identified c^i_{j-1} is calculated, and then the highest one is selected from C^i_j.

The similarity between the two conjunct candidates (hereafter, **Conjunct Similarity**) is calculated using one-to-one word alignment between the candidates. Concretely, the Conjunct Similarity is based on the following two criteria: (1) ratio of words that correspond to a word in the paired candidate and (2) the sum of similarity between two words that correspond to each other. Here, the similarity between two words is calculated based on their POSs and Semantic Similarity. The most appropriate word alignment is calculated by using dynamic programming.

However, this calculation method, which uses one-to-one word alignment, is weak in identifying a conjunct whose adjacent conjunct has a different word length. Figure 3.2 shows an example of a coordinate structure that the conventional method tends to analyze incorrectly. The correct conjuncts of this coordinate structure are "*shinsei wo shi yo to suru mono*" (persons planning to file applications) and "*shinsei sha*" (applicants), but their word alignment does not work well, as shown in Figure 3.3-(a). The words "*wo*", "*shi*", "*yo*", "*to*", "*suru*" (planning to) have no correspondence because no word can have correspondences to multiple words. Since the conventional method imposes a penalty on words that do not have correspondence, the Conjunct Similarity between unequal-length conjuncts tends to become low. On the other hand, as shown in Figure 3.3-(b), there is just one lone word in the word alignment between the two conjuncts "*mono*" (persons) and "*shinsei sha*" (applicants), and thus their Conjunct Similarity becomes larger than that of the two conjuncts "*shinsei wo shi yo to suru mono*" (persons planning to file applications) and "*shinsei sha*" (applicants). As a result, the conventional method incorrectly chooses the two conjuncts "*mono*" (persons) and "*shinsei sha*" (applicants).

Figure 3.2: Coordinate structure with unequal length of conjuncts

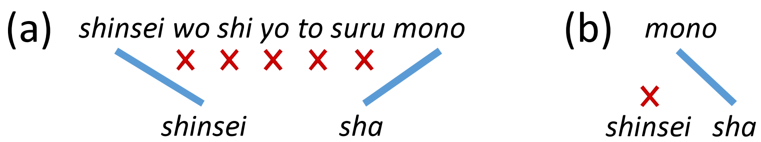

Figure 3.3: Word alignment between "*shinsei wo shi yo to suru mono*" (persons planning to file applications) and "*shinsei sha*" (applicant) in (a), and word alignment between "*mono*" (persons) and "*shinsei sha*" (applicant) in (b)

3.2.4 Further Conjunct Existence Judgment

After identifying conjuncts c^i_j, $c^i_{j-1}, \ldots, c^i_0$, the method judges that a next conjunct c^i_{j+1} exists if all of the following three conditions are satisfied: (1) the word just before the conjunct c^i_j is a comma, (2) the word just before the comma has the same POS as the rightmost word of c^i_j, and (3) the Semantic Similarity between the two words referred to in the second condition is above a certain threshold. The third condition is used only for a pair of noun words.

Assume that the method has chosen "*ryoin*" (both Houses) as c^1_1 in Figure 2.1. Then, the method judges the existence of c^1_2 and concludes that it does not exist since the word just before c^1_1, "*no*" (of), is not a comma.

3.2.5 Substitution of Coordinate Structure

If a conjunct candidate includes an already identified coordinate structure, the Conjunct Similarity with an adjacent conjunct candidate is underestimated because of their different word lengths. To avoid this, the method substitutes the entire identified coordinate structure crd^i with c^i_0.

In Figure 2.1, assuming that the method has identified crd^1, whose conjuncts are "*ryoin*" (both Houses) as c^1_1 and "*ichiin*" (either House) as c^1_0, the method substitutes "*ryoin moshikuwa ichiin*" (both Houses or$_L$ either House) with "*ichiin*" (either House) and uses the new sentence "*kokkai no <u>ichiin</u> moshikuwa gikai no* ..." (... of <u>either House</u> of the Diet or$_L$ assemblies) for the next process.

3.3 Proposed Method

We believe that the conventional method has problems in identifying conjuncts because its approach is based on word alignment. Therefore, our method uses LSTM-based NLMs instead of word alignment in identifying conjuncts. To remove the influence of unequal-length conjuncts, our method encodes a conjunct to a vector. On the other hand, we consider it reasonable to identify coordinate structures deterministically based on the legislation rules unique to Japanese statutory sentences. Therefore, our method identifies coordinate structures and their conjuncts deterministically by the same flow as the conventional method. In this section, we explain our method's procedure in each process in Figure 3.1, focusing on the points that differ from the conventional method.

3.3.1 Coordinator Extraction and Ranking

Table 3.2 shows the targeted coordinators and their priority in our method. Compared with the conventional method, our method adds two coordinators "から" (*kara*; from) and "その他の" (*sonotano*; other$_2$). "*Kara*" is often used to indicate a range (e.g., a range of reference articles) [53]. Judging from a syntactically tagged corpus of statutory sentences [68], we assign Priority 1 to "*kara*." As described in Section 2.1.3, "*sonotano*" (other$_2$) strictly has a different usage from "*sonota*" (other$_1$).

Coordinate structures other than "*kara*" and "*sonotano*" (other$_2$) are formalized by Eq. (3.1). Coordinate structures formed by "*kara*" (from) and "*sonotano*" (other$_2$) are formalized by Eqs. (3.2)[2] and (3.3)[3], respectively.

$$crd^i_{kara} = c^i_1 \cdot \text{"}kara\text{"} \cdot c^i_0 \cdot \text{"}made\text{"}, \tag{3.2}$$
$$crd^i_{sonotano} = c^i_{n_i} \cdot \text{"、"} \cdot c^i_{n_i-1} \cdot \text{"、"} \cdot \ \ldots \ \cdot \text{"、"} \cdot c^i_1 \cdot \text{"、"}? \cdot \ \text{"}\underline{sonota}_{c^i_0}no\text{"}. \tag{3.3}$$

3.3.2 Candidate Extraction

In principle, our method extracts a candidate set C^i_j of a conjunct c^i_j in the same manner as the conventional method. However, we do not impose constraints from the thesaurus on our method since they can be replaced with LSTM-based NLMs. Furthermore, our method introduces additional constraints to more broadly cover

[2]Equation (3.2) shows that if "*kara*" (from) constitutes a coordinate structure, it is always used with "*made*" (to).

[3]As described in Section 2.1.3, "*sonota*" in "*sonotano*" (other$_2$) becomes c^i_0.

Table 3.2: Coordinators targeted in our method and their priorities

Priority	Coordinator		
1	から (*kara*; from (A to B))		
2	及び (*oyobi*; and_L)	若しくは (*moshikuwa*; or_L)	
3	並びに (*narabini*; and_H)	又は (*matawa*; or_H)	
4	その他 (*sonota*; other_1)	その他の (*sonotano*; other_2)	
	かつ (*katsu*; and)	と (*to*; and)	や (*ya*; and)

the usage of coordinators in Japanese statutory sentences [37, 69]. The additional constraints are as follows:

- If k^i is "*oyobi*" (and_L), any candidate of c^i_j should not include any coordinate structure with "*oyobi*" (and_L), since the coordinator "*oyobi*" (and_L) is used only for the lowest-layer coordinate structure.

- If k^i is "*matawa*" (or_H), any candidate of c^i_j should not include any coordinate structure with "*matawa*" (or_H), since the coordinator "*matawa*" (or_H) is used only for the highest-layer coordinate structure.

- If k^i is "*matawa*" (or_H), any candidate of c^i_j should not leave any coordinate structure with "*moshikuwa*" (or_L) that can be included, since a coordinate structure with "*moshikuwa*" (or_L) should be subordinate to a coordinate structure with "*matawa*" (or_H).

3.3.3 Conjunct Identification

Our method identifies a conjunct based on two assumptions: (1) conjunct similarity — the conjuncts in any pair are alike and (2) conjunct interchangeability — the fluency of a sentence is maintained even if the conjuncts in a coordinate structure are swapped with each other. Previous works for coordination analysis [23, 31] use the first assumption. Teranishi et al. [84] uses also the second assumption. Instead of our second assumption, some previous works [23] used another assumption, i.e., that a coordinate structure can be replaced by any of its conjuncts. However, this assumption on replacement may not work well for an incomplete coordinate structure, which our method generates while identifying coordinate structures with more than two conjuncts. Therefore, we adopt our second assumption because it works well even for an incomplete coordinate structure.

In our method, conjuncts c_1^i and c_0^i are identified simultaneously by Eq. (3.4), while conjuncts $c_j^i (j \geq 2)$ are identified by Eq. (3.5):

$$\begin{aligned} (c_1^i, c_0^i) &= \underset{(c_l,c_r)\in C_1^i \times C_0^i}{\arg\max} \; sim(S,(c_l,c_r)) \times flu(S,(c_l,c_r)), && (3.4) \\ c_j^i &= \underset{c_l \in C_j^i}{\arg\max} \, sim(S,(c_l,c_{j-1}^i)) \times flu(S,(c_l,c_{j-1}^i)), && (3.5) \end{aligned}$$

where $sim(S,(c_l,c_r))$ is a score of the similarity between two conjuncts $c_l,\ c_r$ (hereafter, **Similarity Score**) and $flu(S,(c_l,c_r))$ is a score of the fluency of the sentence S after swapping two conjuncts $c_l,\ c_r$ (hereafter, **Fluency Score**). Referring to [23], our method calculates each score with two types of LSTM-based NLMs: (1) an **F-NLM** constructed by being fed words in forward order and (2) a **B-NLM** fed words in backward order. We use the two types of LSTM-based NLMs in order to use the context of the sentence even if a coordinate structure is at the beginning or end of the sentence.

Our LSTM-based NLMs are trained with raw word sequences, while other neural-based coordination analysis methods [23, 84, 85] are trained with parsed sentence data that have coordination information. There is no large-scale treebank that contains thousands of Japanese statutory sentences, and it is prohibitively costly to compile such a treebank because statutory sentences are highly technical, long, and complex. Therefore, our approach goes without training coordination itself.

Similarity Score

The Similarity Score $sim(S,(c_l,c_r))$ is calculated by Eq. (3.6):

$$sim(S,(c_l,c_r)) = sim_f(S,(c_l,c_r)) + sim_b(S,(c_l,c_r)), \qquad (3.6)$$

where

$$\begin{aligned} sim_f(S,(c_l,c_r)) &= 1 + sim_c(vec_f(W_{fl}), vec_f(W_{fr})), && (3.7) \\ sim_b(S,(c_l,c_r)) &= 1 + sim_c(vec_b(W_{bl}), vec_b(W_{br})), && (3.8) \end{aligned}$$

and $sim_c(\boldsymbol{u},\boldsymbol{v})$ is the cosine similarity between two vectors, $\boldsymbol{u}$ and $\boldsymbol{v}$. Adding one to sim_c prevents sim_f and sim_b from becoming negative. $vec_f(W)$ denotes the output vector of the hidden layer nearest to the output layer after inputting a word sequence W to the F-NLM, and $vec_b(W)$ denotes the output vector from the B-NLM. We expect that our method captures the similarity between word sequences in consideration of their contexts by calculating it based on the hidden layer's output vectors.

The four word sequences W_{fl}, W_{fr}, W_{bl}, W_{br} are generated by Eqs. (3.9-3.12), respectively.

$$\begin{aligned} W_{fl} &= W_f \cdot c_l, & (3.9)\\ W_{fr} &= W_f \cdot c_r, & (3.10)\\ W_{bl} &= c_l \cdot W_b, & (3.11)\\ W_{br} &= c_r \cdot W_b, & (3.12) \end{aligned}$$

where W_f represents the word sequence before c_l in S and W_b represents the word sequence after c_r in S. "·" represents the string concatenation operator. We assume that our method captures a broader context by using word sequences placed before and after the coordinate structure.

Assuming that we are calculating $sim(S, (\text{“}ryoin\text{”}, \text{“}ichiin\text{”}))$ in Figure 2.1, the four word sequences W_{fl}, W_{fr}, W_{bl} and W_{br}, respectively, become the following:

$$\begin{aligned} W_{fl} &= \text{“}\textit{kokkai no } \underline{\textit{ryoin}}_{c_l}\text{”}, & (3.13)\\ W_{fr} &= \text{“}\textit{kokkai no } \underline{\textit{ichiin}}_{c_r}\text{”}, & (3.14)\\ W_{bl} &= \text{“}\underline{\textit{ryoin}}_{c_l}\textit{ moshikuwa gikai no } \ldots\text{”}, & (3.15)\\ W_{br} &= \text{“}\underline{\textit{ichiin}}_{c_r}\textit{ moshikuwa gikai no } \ldots\text{”}. & (3.16) \end{aligned}$$

Fluency Score

The Fluency Score $flu(S, (c_l, c_r))$ is calculated by Eq. (3.17):

$$flu(S, (c_l, c_r)) = flu_f(W_s) + flu_b(W_s), \quad (3.17)$$

where W_s is a word sequence after swapping two conjuncts c_l and c_r in S. For example, W_s in calculating $flu(S, (\text{“}ryoin\text{”}, \text{“}ichiin\text{”}))$ in Figure 2.1 is "*kokkai no* *ichiin* *moshikuwa* *ryoin* *moshikuwa gikai no* ..." $flu_f(W_s)$ and $flu_b(W_s)$ denote the fluency of W_s based on the F-NLM and B-NLM, respectively. These fluencies are calculated by the following two equations:

$$\mathit{flu}_f(W_s) = \sqrt[|W_s|]{\prod_{t=1}^{|W_s|} P_f(w_t|w_1, w_2, \ldots, w_{t-1})}, \tag{3.18}$$

$$\mathit{flu}_b(W_s) = \sqrt[|W_s|]{\prod_{t=1}^{|W_s|} P_b(w_t|w_{|W_s|}, w_{|W_s|-1}, \ldots, w_{t+1})}, \tag{3.19}$$

where $P_f(w_t|w_1, w_2, \ldots, w_{t-1})$ is the probability that w_t appears after the word sequence $w_1, w_2, \ldots, w_{t-1}$ in the F-NLM. $P_b(w_t|w_{|W_s|}, w_{|W_s|-1}, \ldots, w_{t+1})$ is the corresponding probability in the B-NLM. We use the geometric mean to eliminate the effect of sentence length.

3.3.4 Further Conjunct Existence Judgment

Our method judges whether c^i_{j+1} appears before c^i_j in the same manner as the conventional method (see Section 3.2.4), except that we do not impose constraints based on Semantic Similarity on our method, since it does not use a thesaurus. In other words, our method judges that there is a further conjunct c^i_{j+1} next to the conjunct c^i_j if the following two conditions are satisfied:

- the word just before the conjunct c^i_j is a comma;
- the word just before the comma has the same POS as the rightmost word of c^i_j.

However, using only these conditions sometimes leads to an incorrect judgment. For example, we assume that our method identifies a noun phrase conjunct "*fusei kyoso no boshi*" (prevention of unfair competition) in Figure 3.4. Consequently, our method mistakenly identifies "*tame*" (in order to) as another conjunct, since the word just before "*fusei kyoso no boshi*" (prevention of unfair competition) is a comma and the word before the comma, which is "*tame*" (in order to), is a noun.

Nevertheless, even if our method incorrectly judges the further conjunct's existence, we do not consider imposing a constraint based on semantic similarity. This is because such a similarity-based constraint may reject words that are truly parts of conjuncts and lead to failure in identifying higher coordinate structures.

3.3.5 Substitution of Coordinate Structure

Our method substitutes crd^i with c^i_0, in common with the conventional method.

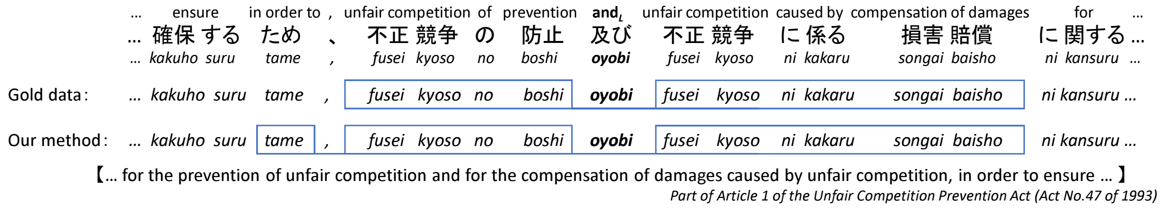

Figure 3.4: Incorrect coordination analysis by identifying an unnecessary conjunct

3.4 Experiment

To evaluate the effectiveness of our method, we conducted an experiment on identifying coordinate structures in two Japanese acts (Administrative Procedure Act and Unfair Competition Prevention Act) contained in a syntactically tagged corpus of statutory sentences [68].

3.4.1 Outline of Experiment

We constructed both input data and gold data from the above corpus [68]. In the corpus, each sentence is annotated by hand with information on morphological analysis, *bunsetsu* segmentation, and dependency analysis. We constructed the input data by removing information on dependency analysis and extracting each parenthesized expression as another independent sentence. The size of the input data is 592 sentences with 792 coordinate structures. We created the gold data using the dependency information in the corpus.

We built our LSTM-based NLMs [80] for our method. Based on our preliminary experiment, we decided that each model has four hidden layers and that each layer consists of 650 LSTM units. A sequence of one-hot word vectors is fed into the model. In creating the one-hot word vector, we used basic forms of words obtained from a Japanese morphological analyzer MeCab (v0.98) [46] with an IPA dictionary. We selected the 15,000 most frequent words for the vocabulary and added two special tokens, i.e., EOS (End Of Sentence) and UNK (UNKnown word), to the vocabulary. Thus, the numbers of input and output layer units are both 15,002. The NLMs were trained using Chainer (v1.15.0) [4]. We used softmax cross entropy as a loss function and updated the parameters by SGD (learning rate was 1) while clipping the L2 norm of gradient vectors to 5 and dropping out half of the randomly selected units. As the training data for our NLMs, we used 574,062 statutory sentences crawled from the Japanese Law Translation Database System (JLT)[5] as of Nov. 2016.

[4] http://chainer.org/
[5] http://www.japaneselawtranslation.go.jp/

Table 3.3: Experimental results of coordination analysis

		Our method	Matsuyama	Yamakoshi
Coordinate Structure	P	**66.1**% (463/700)	46.8% (312/667)	52.5% (336/640)
	R	**64.6**% (463/717)	43.5% (312/717)	46.9% (336/717)
	F	**65.2**	45.1	49.5
Conjunct	P	**83.0**% (1,372/1,653)	68.0% (1,019/1,499)	72.5% (1,114/1,536)
	R	**81.0**% (1,372/1,694)	60.2% (1,019/1,694)	65.8% (1,114/1,694)
	F	**82.0**	63.8	69.0

For comparison, we implemented Matsuyama et al.'s method [54] on our own and conducted the same experiment. The parameters of the method (e.g., threshold of the Semantic Similarity) were tuned using all of the input data. We also run Yamakoshi et al.'s method [95] that utilizes a CFG parser for hierarchy identification.

In the evaluation, we measured recall, precision, and F-measure for both conjuncts and coordinate structures. Precision P is the ratio of correctly identified conjuncts or coordinate structures among all automatically identified ones. Recall R is the ratio of correctly identified conjuncts or coordinate structures among all of those in the gold data. We judged that a conjunct has been correctly identified if its scope agreed completely with that of the gold data. Furthermore, we judged that a coordinate structure has been correctly identified if all of its conjuncts were correctly identified.

3.4.2 Experimental Results

Table 3.3 shows the experimental results of each method. "Matsuyama" and "Yamakoshi" mean the methods of Matsuyama et al. [54] and Yamakoshi et al. [95], respectively. Since our method was much superior to the conventional methods in all evaluation indexes, we confirmed its effectiveness for coordination analysis of Japanese statutory sentences.

Figure 3.5 shows an example of a coordinate structure correctly identified by our method. The conventional methods could not correctly identify the conjunct "*shinsei wo shi yo to suru mono*" (persons planning to file applications). As described in Section 3.2.3, this failure seems to be caused by a disparity of word length between the two conjuncts. On the other hand, our method transforms every conjunct candidate into a fixed-length vector by the LSTM-based NLMs, using context before and after the conjunct, and then directly calculates the score of a pair consisting of two conjunct candidates using the vectors. Therefore, we can conclude that our method correctly

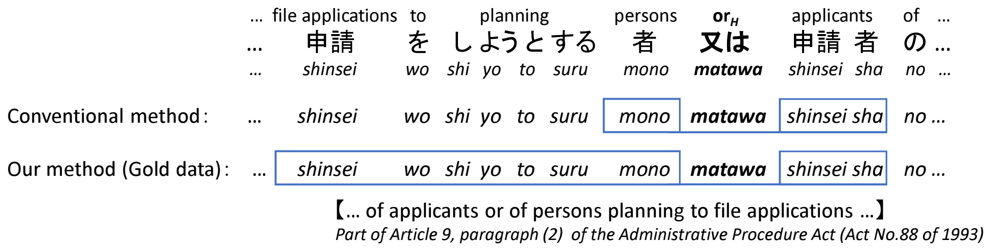

Figure 3.5: Successful coordination analysis case in our method

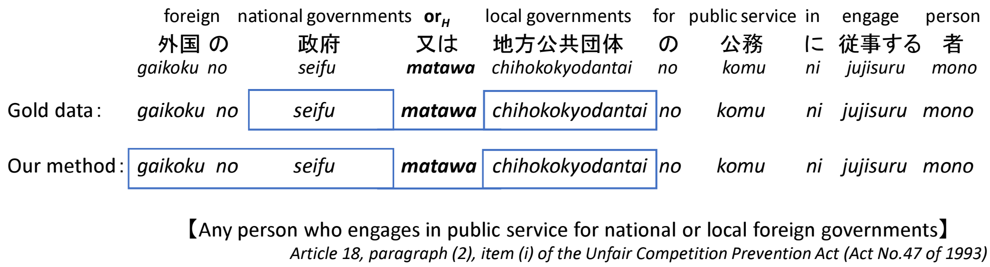

Figure 3.6: Failed coordination analysis case in our method

identified all of the conjuncts without being affected by their word lengths.

Next, Figure 3.6 shows an example of coordinate structures incorrectly identified by our method. In the gold data, "*seifu*" (national governments) and "*chihokokyodantai*" (local governments) are the conjuncts of the coordinate structure with "*matawa*" (or_H). However, our method identified "*gaikoku no seifu*" (foreign national governments) as the first conjunct by mistake.

3.5 Discussion

In this section, we reveal the characteristics and effectiveness of our method. First, we investigate our experimental results in more detail to evaluate our method as a coordination analysis method for Japanese statutory sentences. Second, we look further into the performance of the subtasks in our method.

3.5.1 Discussion of Experimental Results

The key characteristics of coordination in Japanese statutory sentences are the uniquely used coordinators and complex hierarchy. In this section, we look into our experimental results from the viewpoint of these characteristics.

Table 3.4: Results of coordination analysis by coordinator (F-measure)

Coordinator	Rate (Freq.)	Our method	Matsuyama	Yamakoshi
matawa (or$_H$)	36.7% (263)	**63.8**	49.4	49.6
moshikuwa (or$_L$)	25.2% (181)	**75.0**	61.5	54.8
oyobi (and$_L$)	15.9% (114)	**73.7**	57.0	66.7
narabini (and$_H$)	3.5% (25)	**54.9**	23.5	31.4
sonotano (other$_2$)	6.8% (49)	57.1	N/A	48.4
kara (from (A to B))	4.7% (32)	90.6	N/A	N/A
only comma	3.1% (22)	N/A	N/A	N/A
sonota (other$_1$)	1.8% (13)	**30.8**	0.0	15.4
katsu (and)	1.8% (13)	**34.8**	10.5	8.7
to (and)	0.7% (5)	0.0	0.0	N/A

Results by Coordinators

Our method introduces additional coordinators and rules to follow more closely the usage of coordinators in Japanese statutory sentences [37, 69]. We evaluated the performance of our method and that of the conventional methods by the type of coordinators to compare the two methods' characteristics more accurately.

Table 3.4 shows each method's F-measures for coordinate structures sorted by type of coordinator. The second column indicates the frequency and its rate (among all coordinate structures) of coordinate structures sorted by type of coordinator. Our method dramatically outperformed the conventional methods for all of the coordinators except "*to*" (and).

As described in Section 2.1.3, the four coordinators "*matawa*" (or$_H$), "*moshikuwa*" (or$_L$), "*oyobi*" (and$_L$), and "*narabini*" (and$_H$) constitute coordinate structures based on the legislation rules unique to Japanese statutory sentences. Since 81.3% of all coordinate structures are composed of these four coordinators, the results for them had a huge effect on the total substantial improvement in Table 3.3. From this result, we confirmed the effectiveness of our method on hierarchical coordination analysis for Japanese statutory sentences.

Comparing two fundamental coordinators "*matawa*" (or$_H$) and "*oyobi*" (and$_L$) that are also used in non-hierarchical coordinate structures, coordinate structures of the former were predicted worse. One potent reason is that identification of coordinate structures of "*matawa*" (or$_H$) with hierarchy may receive a bad influence from identification errors of their lower coordinate structures.

Table 3.5: Results of coordination analysis by layer (F-measure)

Height	Rate	(Freq.)	Our method	Matsuyama	Yamakoshi
1	71.4%	(512)	**74.1**	54.3	58.6
2	21.2%	(152)	**50.2**	17.9	20.2
3	5.4%	(39)	**26.2**	22.2	17.9
4	1.5%	(11)	**8.7**	0.0	0.0
5	0.4%	(3)	0.0	0.0	0.0

Results by Height of Coordinate Structure

Complex hierarchical coordinate structures appear frequently in Japanese statutory sentences, and it is not easy to completely identify their scope and hierarchy. Therefore, we evaluate how well our method can identify such hierarchical coordinate structures.

Table 3.5 shows each method's F-measures sorted by height of coordinate structure. Since our method outperformed the conventional methods at almost every height except the fifth one, we confirmed the effectiveness of our method for hierarchical coordinate structures frequently appearing in Japanese statutory sentences.

However, our method, as with the conventional methods, tended to lose performance in higher coordinate structures. Three reasons seem to account for this result:

- Failure to identify lower coordinate structures affected the identification of the higher ones.
- The higher the coordinate structure, the longer its conjunct tends to become, since the structure also contains a lower coordinate structure. Therefore, the number of conjunct candidates tends to increase, and the probability of incorrect choice increases.
- When coordinators with the same priority appear in succession, there exists a hierarchical coordinate structure that our method cannot identify due to our rules on the analysis order of coordinators. We will delve into this in Section 3.5.2.

Therefore, the difficulty of identifying a higher coordinate structure remains in our method.

3.5.2 Discussion of Subtask Performance

Our method identifies **hierarchical coordinate structures** by sequentially executing the following two subtasks: (1) **hierarchy** identification and (2) **conjunct** identification. The hierarchy identification determines the hierarchy among the coordinate structures in a sentence. The conjunct identification determines the scopes of all conjuncts in a coordinate structure. In the following, we analyze the performance of each subtask.

Performance of Hierarchy identification

Our method roughly identifies a hierarchy among coordinate structures through coordinator ranking as described in Section 3.3.1, and then it identifies a complete hierarchy among them through conjunct identification as described in Sections 3.3.2—3.3.3. Since Japanese statutory sentences are written strictly according to Japanese legislation-drafting manuals, our coordinator ranking method is based on the heuristic rules obtained from the manuals, which consist of the Priority Rule and Position Rule defined in Section 3.2.1.

To evaluate our coordinator ranking method, we measured the percentage (hereafter, ranking accuracy) of the pairs consisting of two coordinators whose order relation in priority is correct. If two coordinate structures do not have hierarchy, we consider that any order between their coordinators is correct, since our method can correctly identify such coordinate structures regardless of their coordinator ranking. For example, if there are four coordinators in a sentence and two pairs have correct order relation in priority, then the percentage is 33.3% ($= 2/{}_4C_2$).

The micro-averaged ranking accuracy for all of the 592 experimental input sentences in Section 3.4.1 was 95.7% ($= 1,040/1,087$). We can confirm that our heuristic coordinator ranking method can achieve high ranking accuracy and our heuristic rules are effective for coordinator ranking.

On the other hand, our coordinator ranking method did not succeed completely. Table 3.6 shows 47 pairs of coordinators whose order relation in priority were incorrectly identified by our method. Out of the 47 pairs, 29 pairs failed in identification because of the Priority Rule, while the other identification failures resulted from the Position Rule.

First, we discuss the cause of the failures from the Priority Rule shown in Table 3.2. In view of the Japanese legislation-drafting manuals referred to in Section 2.1.3, there is specific priority between two pairs of coordinators "*matawa*" (or$_H$) and "*moshikuwa*" (or$_L$), and "*oyobi*" (and$_L$) and "*narabini*" (and$_H$). However, there is no priority between other pairs of coordinators than the two pairs mentioned above. For example, there is no priority between "*matawa*" (or$_H$) and "*sonotano*" (other$_2$). This means that either a coordinate structure with "*matawa*" (or$_H$) or that with "*sonotano*" (other$_2$) can be the upper-layer one. Actually, the coordinate structure

Table 3.6: Combinations and numbers of coordinators our method could not identify correctly

Larger coordinator	Smaller coordinator	Rule violated	#
oyobi (and$_L$)	*to* (and)	Priority	2
	matawa (or$_H$)	Priority	5
moshikuwa (or$_L$)	*sonotano* (other$_2$)	Priority	1
	moshikuwa (or$_L$)	Position	13
	matawa (or$_H$)	Priority	4
narabini (and$_H$)	*sonotano* (other$_2$)	Priority	1
	to (and)	Priority	2
	matawa (or$_H$)	Position	4
matawa (or$_H$)	*katsu* (and)	Priority	1
	sonotano (other$_2$)	Priority	13
katsu (and)	*sonotano* (other$_2$)	Position	1
Total			47

with "*sonotano*" (other$_2$) becomes the upper-layer one in sentence (a) of Figure 3.7, while that with "*matawa*" (or$_H$) becomes the upper-layer one in sentence (b) of Figure 3.7. The Priority Rule cannot capture the difference between such examples because the rule judges the priority only by the type of coordinator.

As another example, coordinate structures with "*kara*" (from) rarely have lower coordinate structures like that in Figure 3.8, and such a case does not appear in our test data [68]. Our method cannot identify such hierarchical coordinate structures correctly since the priority of "*kara*" (from) is 1.

Next, we discuss the cause of the failures from the Position Rule. When plural coordinators have the same priority, our method regards a coordinator that comes earlier as the one that constitutes a lower-layer coordinate structure. However, there are some examples of hierarchical coordinate structures that violate this rule. In Figure 3.9, among four "*moshikuwa*" (or$_L$) in this sentence, the second one constitutes the highest layer coordinate structure in the gold data. As described above, the Position Rule cannot assume that the second "*moshikuwa*" (or$_L$) constitutes a higher layer coordinate structure than that of the third "*moshikuwa*" (or$_L$).

Since the deterministic approach has limits for hierarchy identification, we need to adopt a more effective approach.

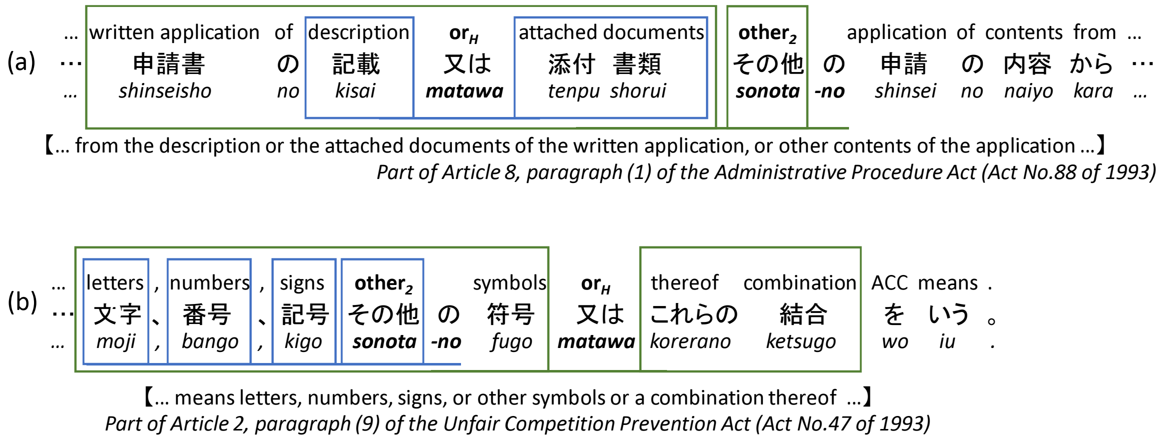

Figure 3.7: Sentences having coordinate structures with "*matawa*" (or$_H$) and "*sonotano*" (other$_2$): the coordinate structure with "*sonotano*" (other$_2$) is upper-layer in sentence (a), while that with "*matawa*" (or$_H$) is upper-layer in sentence (b).

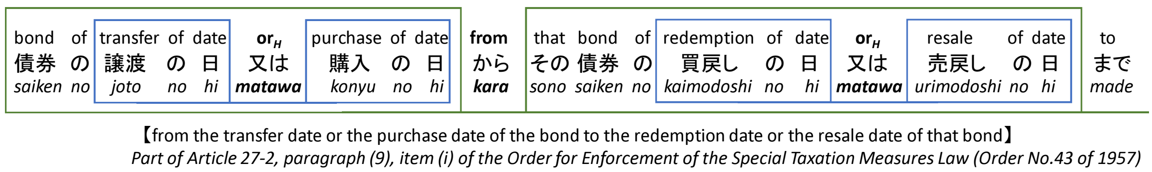

Figure 3.8: Sentence having a coordinate structure with "*kara*" (from), which has lower coordinate structures

Performance of Conjunct Identification

After identifying hierarchy among coordinate structures, our method deterministically identifies an internal structure of each coordinate structure, as described in Sections 3.3.2—3.3.5. In this section, we evaluate the performance of the conjunct identification from two points of view.

First, we conducted an experiment on coordination analysis under the condition of using an oracle coordinator ranking instead of the coordinator ranking generated in

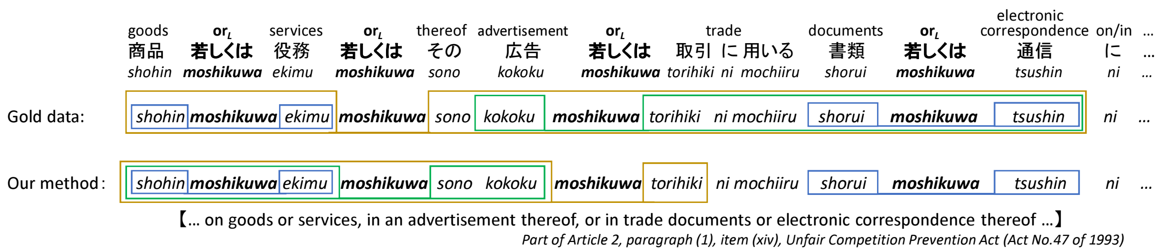

Figure 3.9: Coordination analysis of a statutory sentence with multiple "*moshikuwa*" (or$_L$)

Table 3.7: Performance of coordination analysis with oracle coordinator ranking: Performance without an oracle is the same as that of our method in Table 3.3

		Without oracle		With oracle	
Coordinate Structure	P	66.1%	(463/700)	**71.3%**	(492/690)
	R	64.6%	(463/717)	**68.6%**	(492/717)
	F	65.2		**69.9**	
Conjunct	P	83.0%	(1,372/1,653)	**85.6%**	(1,386/1,619)
	R	81.0%	(1,372/1,694)	**81.8%**	(1,386/1,694)
	F	82.0		**83.7**	

Section 3.3.1. We created the oracle from the hierarchy among coordinate structures acquired from the gold data. We used the same experimental settings and evaluation measurement as Section 3.4.1, except for applying the oracle.

Table 3.7 shows the experimental results. Even if the hierarchy among coordinate structures is given, the F-measure of coordination analysis was less than 70 points. We can confirm that conjunct identification is an inherently difficult subtask. On the other hand, coordination analysis with oracle coordinator ranking identified 29 more coordinate structures correctly and achieved 4.7 points higher performance at F-measure than the analysis without oracle. Therefore, we can confirm that it is fruitful to improve the coordinator ranking method.

Second, we conducted an ablation study to evaluate the contributions of the Similarity Score (see Section 3.3.3) and the Fluency Score (see Section 3.3.3) used for conjunct identification. We implemented two experimental methods that (1) identifies each conjunct only with the Similarity Score and (2) identifies each conjunct only with the Fluency Score. Then we conducted an experiment on coordination analysis using the two methods described above and using our proposed method that uses both the Similarity Score and the Fluency Score. We used the same experimental settings and evaluation measurement as Section 3.4.1.

Table 3.8 shows the experimental results. Our proposed method using both functions achieved the best performance in every evaluation measurement. From this result, we can confirm that it is effective to use both the Similarity Score and the Fluency Score.

3.6 Summary

We proposed a coordination analysis method for Japanese statutory sentences using neural language models. Our method identifies the hierarchy of coordinate structures and the scope of all conjuncts in them. For hierarchy identification, we applied the

Table 3.8: Performance of coordination analysis per scoring function: "Both" is the same as that of our method in Table 3.3

		Similarity Score	Fluency Score	Both
Coordinate Structure	P	60.3% (422/700)	54.9% (384/699)	**66.1**% (463/700)
	R	58.9% (422/717)	53.6% (384/717)	**64.6**% (463/717)
	F	59.6	54.2	**65.2**
Conjunct	P	78.5% (1,264/1,611)	77.2% (1,273/1,649)	**83.0**% (1,372/1,653)
	R	74.6% (1,264/1,694)	75.1% (1,273/1,694)	**81.0**% (1,372/1,694)
	F	76.5	76.2	**82.0**

legislation rules on hierarchical coordination. For conjunct identification, we utilized LSTM-based NLMs to score each conjunct candidate. The experimental results show that our method was much more effective than the conventional methods.

Chapter 4

Japanese Legal Term Correction

In this chapter, we describe the study for Japanese legal term correction. First, we overview the background of legal term correction in Section 4.1. In Section 4.2, we define legal term correction task and consider a general algorithm for this task. In Section 4.3, we propose our methods that come from two different approaches: the Random Forest approach and the BERT approach. In Section 4.4, we conduct experiments for the proposed methods, and then we discuss the result in Section 4.5. Finally, we conclude this piece of study in Section 4.6.

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 that creatively expresses thoughts or sentiments】

Figure 4.1: Phrases in Japanese statutory sentences with legal term (underlined), from the Copyright Act (Act No. 48 of 1970)

4.1 Introduction

Legislation drafting requires careful attention. The Japanese government deals with this task by thorough legislation rules and final inspection by the Cabinet Legislation Bureau. The legislation rules regulate the document structures, orthography, and phraseology of the statutes. Among them, they explicitly define distinct usage and meaning to many mistakable **legal terms**. For example, we have a set of three legal terms "者 (a)," "物 (b)," and "もの (c)." They are all pronounced *mono* and share the concept of "object." Term (a) only means a natural or juristic person, term (b) only means a tangible object that is not a natural or juristic person, and term (c) only means an abstract object or a complex of these objects. Figure 4.1 displays phrases including these legal terms. In ordinary Japanese sentences, unlike in statutory sentences, term (c) can refer to term (a) and term (b). That is, we can use term (c) as "著作物 を 創作する もの" to express "a person who creates a work" in ordinary Japanese. However, this usage is not allowed in Japanese statutory sentences because of the rules mentioned above.

To avoid errors and inconsistencies, we should inspect statutory sentences in accordance with the legislation rules. However, inspections of statutory sentences are still conducted mainly by human experts in legislation, which requires deep knowledge and an enormous amount of labor. The legislation rules are applied to not only statutes but also ordinances and orders of local governments. Furthermore, legal documents in a broad sense, such as contracts, terms of use, and articles of incorporation, are also written in compliance with the rules. Manual inspections are also dominant in these domains.

From this background, we aim to establish a proofreading method for legal terms. Although there are a lot of studies on proofreading methods as we discussed in Section 1.2.2, to our knowledge, no study other than ours focuses on legal terms that are used separately on the basis of context. Therefore, we initially provide a definition of this task so that we can search for a solution. Concretely, we define it as a special case of the multi-choice sentence completion test. With this setting, we first propose an approach that uses Random Forest classifiers [7], each of which is optimized for each set of similar legal terms. The classifiers input words adjacent to the targeted legal term and output the most adequate legal term in the targeted legal term set.

We then propose another approach that uses a classifier based on BERT (Bidirectional Encoder Representations from Transformers) [18]. A BERT classifier captures an abundant amount of linguistic knowledge by fine-tuning a "ready-made" model that is pretrained by a large quantity of text. Furthermore, it utilizes more contexts than the conventional classifiers in prediction, since BERT classifiers can handle whole sentences (128 tokens maximum in our experiment).

Here, we consider the **two-level infrequency** of the legal term correction task: **term-level infrequency** and **set-level infrequency**. Term-level infrequency refers to large frequency gaps between legal terms in a legal term set. It causes a class imbalance problem, where the classifiers tend to choose frequent terms. Set-level infrequency means the infrequency of a legal term set. This causes an underfitting problem since classifiers suffer from a shortage of training examples.

To cope with the two-level infrequency, we apply three training techniques in the BERT approach. The first technique is to preliminarily adapt the pretrained BERT model to Japanese statutory sentences. This technique contributes to the improvement of the overall performance by providing prior knowledge of the statutory sentences to the proposed method. The second technique is to undersample training examples softly and repetitively to cope with the term-level infrequency. The third technique is to unify classifiers for individual legal term sets into one model to cope with the set-level infrequency by sharing common knowledge of legal term correction. Moreover, this technique reduces the total model size, which is critical in our method because a BERT model is quite huge (more than 1 GB in our case).

4.2 Definition of Legal Term Correction Task

In this section, we review the legal term correction task that we defined and a general algorithm for the task in Section 4.2.1 and Section 4.2.2, respectively.

4.2.1 Task Definition

The legal term correction task is defined as follows:

- A sentence $W = w_1 w_2 \ldots w_{|W|}$ ($w_i \in V$) and a set of similar legal terms $T = \{t_1, t_2, \ldots, t_{|T|}\} \subseteq V^+$ are given, where V^+ is the Kleene plus of vocabulary V, that is, a legal term $t \in T$ can be either a word or multiple words;

- Each 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.

This task is regarded as a kind of multi-choice sentence completion test by introducing the following ideas:

- W^l ____ W^r is a sentence with a blank, where ____ is a blank, and W^l and W^r are two word sequences adjacent to the left and right of t, respectively.

- T is the choices, one of which adequately fills in the blank in the sentence.

4.2.2 Generic Algorithm

A general algorithm for this task is shown in Algorithm 1, where $\mathit{score}(W^l, t, W^r)$ is any scoring function that calculates the likelihood of t.

Algorithm 1 Algorithm for legal term correction

Input: W, T
Output: Suggests
  Suggests $\leftarrow \emptyset$
  for all (i, j) such that $w_i\ w_{i+1} \ldots w_j = t \in T$ **do**
    $W^l \leftarrow w_1\ w_2 \ldots w_{i-1}$
    $W^r \leftarrow w_{j+1}\ w_{j+2} \ldots w_{|W|}$
    $\hat{t} \leftarrow \arg\max_{t' \in T} \mathit{score}(W^l, t', W^r)$
    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

For example, suppose that the statutory sentence W and the legal term set T are as follows:

$$W = \begin{array}{cccccc} \textit{chosakubutsu} & \textit{wo} & \textit{sosakusuru} & \textit{mono} & \textit{no} & \textit{hogo} \\ \text{著作物} & \text{を} & \text{創作する} & \text{もの}_{\text{(c)}} & \text{の} & \text{保護}, \\ \text{work} & \text{ACC} & \text{create} & \text{a.object} & \text{of} & \text{protection} \end{array} \tag{4.1}$$

$$T = \{\text{者}_{\text{(a)}}, \text{物}_{\text{(b)}}, \text{もの}_{\text{(c)}}\}. \tag{4.2}$$

The algorithm finds term (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:

$$score\left(\begin{array}{ccccc} \textit{chosakubutsu wo sosakusuru}, & \textit{mono}, & \textit{no hogo} \\ \text{著作物 を 創作する}, & \text{者}_{\text{(a)}}, & \text{の 保護} \\ \text{work ACC create}, & \text{person}, & \text{of protection} \end{array}\right), \tag{4.3}$$

$$score\left(\begin{array}{ccccc} \textit{chosakubutsu wo sosakusuru}, & \textit{mono}, & \textit{no hogo} \\ \text{著作物 を 創作する}, & \text{物}_{\text{(b)}}, & \text{の 保護} \\ \text{work ACC create}, & \text{t.object}, & \text{of protection} \end{array}\right), \tag{4.4}$$

$$score\left(\begin{array}{ccccc} \textit{chosakubutsu wo sosakusuru}, & \textit{mono}, & \textit{no hogo} \\ \text{著作物 を 創作する}, & \text{もの}_{\text{(c)}}, & \text{の 保護} \\ \text{work ACC create}, & \text{a.object}, & \text{of protection} \end{array}\right). \tag{4.5}$$

We expect the algorithm to choose the first option and to output a suggestion that "もの $_{\text{(c)}}$" in W be replaced into "者 $_{\text{(a)}}$".

4.3 Approaches

In this section, we introduce two approaches for solving the legal term correction task of Japanese statutory sentences. Section 4.3.1 describes the first approach that uses Random Forest classifiers. Section 4.3.2 describes the second approach that uses a BERT classifier.

4.3.1 Approach 1 — Using Random Forest Classifiers

In this approach, we use Random Forest classifiers as the scoring function. Here, we prepare distinct sets of decision tree classifiers for each legal term set T. Concretely, the scoring function for T using Random Forest classifiers $score_{RF_T}$ is defined as follows:

$$score_{RF_T}(W^l, t, W^r) = \sum_{d \in D_T} P_d(t | w^l_{|W^l|-N+1}, \ldots, w^l_{|W^l|-1}, w^l_{|W^l|}, w^r_1, w^r_2, \ldots, w^r_N), \quad (4.6)$$

where D_T is a set of decision trees for the legal term set T and $P_d(t | w^l_{|W^l|-N+1}, \ldots, w^l_{|W^l|-1}, w^l_{|W^l|}, w^r_1, w^r_2, \ldots, w^r_N)$ is the probability (actually 0 or 1) that $d \in D_T$ chooses a legal term $t \in T$ based on features $w^l_{|W^l|-N+1}, \ldots, w^l_{|W^l|-1}, w^l_{|W^l|}, w^r_1, w^r_2, \ldots, w^r_N$. 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). Here, the right-most N words of W^l are used because they are the nearest words to t.

For the following reasons, we decided not to employ the neural language models described in Section 2.2.1 but Random Forest for this task.

We can optimize hyperparameters of the Random Forest classifier for each legal term set: Especially, we consider that it is fruitful to determine an optimal window size for each legal term set. For example, judging use of “直ちに $_{\text{(h)}}$” (*tadachini*), “速やかに $_{\text{(i)}}$” (*sumiyakani*), and “遅滞なく $_{\text{(j)}}$” (*chitainaku*) usually requires to observe the whole context of the sentence. Therefore, a wide size of windows will be needed for this legal term set. On the other hand, neural language models are typically trained with a unified dataset; that is, they need to use the same parameters for the prediction of any legal term set.

Classifier can easily cope with multi-word legal term sets: Classifiers including Random Forest treat a legal term as one class regardless of its word count. On the other hand, language models including n-gram model, CBOW, vLBL, and even BERT's masked language model architecture are designed to predict a single word from the given context. Especially, CBOW, vLBL, and BERT's masked language model that predict a word from bi-directional adjacent words will suffer from the output of multiple words at once. Therefore, to handle multi-word legal terms with a language model, we need to apply a special treatment. One approach is to concatenate such legal terms to one word in the preprocessing phase. Another approach is to calculate probabilities of words forming a multi-word legal term and then combine them. However, the latter approach may overestimate the probabilities since word sequences of multi-word legal terms can be more likely to appear than other non-phrasal word sequences.

4.3.2 Approach 2 — Using a BERT Classifier

In this section, we describe the second approach that utilizes a BERT classifier. First, we overview this method. In this method, we apply three training techniques to cope with an infrequency issue that comes from two levels of infrequency. We explain that two-level infrequency first, and then we explain the training techniques for the infrequency.

Overview

We utilize a pretrained BERT classifier for the scoring function $score(W^l, t, W^r)$. The pretrained BERT model solves the Random Forest classifiers' problem of dropping linguistic knowledge. Our BERT classifier inputs a "masked" sentence where the targeted legal term t is masked 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.

$$score_{BERT}(W^l, t, W^r) = BERT(t|S), \tag{4.7}$$

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

$$S = pp(w_1^l w_2^l \dots w_{|W^l|}^l \text{ [MASK] } w_1^r w_2^r \dots w_{|W^r|}^r), \tag{4.8}$$

where $pp(W)$ is a function to truncate the input sentence W on the masked legal term "[MASK]" that was originally t. Even when this BERT classifier inputs a sentence, 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 in our experiment).

Two-level infrequency

Practically, we handle multiple numbers of legal term sets each of which contains multiple numbers of legal terms. This setting will cause an infrequency that consists of two levels.

At the first level, the **term-level infrequency** is located, which refers to frequency gaps between legal terms in a legal term set. Here, we call legal terms with less frequency in their legal term sets as **infrequent legal terms**. For example, legal term "規程 $_{(g)}$" of legal term set { 規定 $_{(f)}$, 規程 $_{(g)}$} can be regarded as an infrequent legal term. In our dataset, the former occurs 401,381 times while the latter occurs only 4,139 times. The term-level infrequency causes a class imbalance problem.

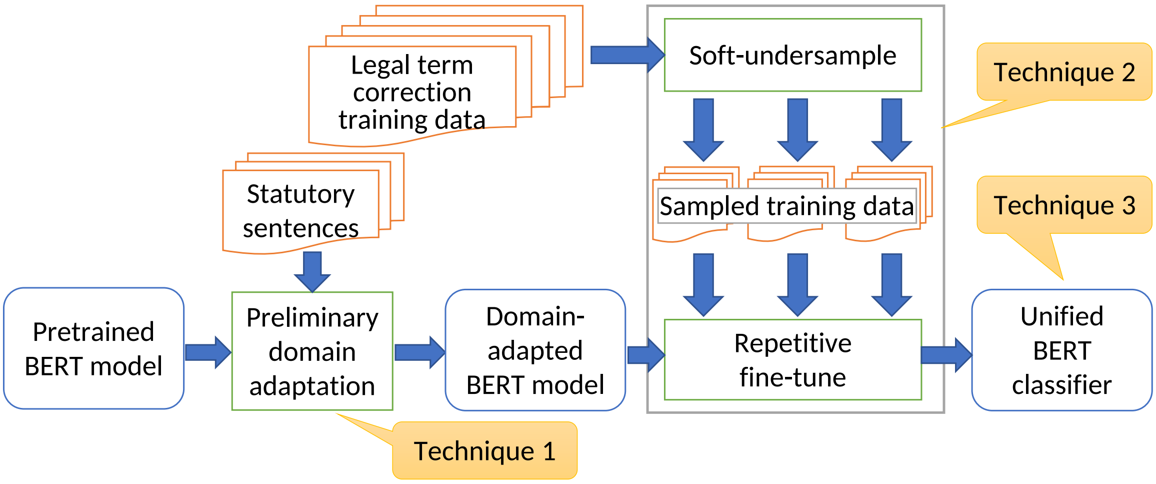

Figure 4.2: Training scheme for our BERT legal term classifier

The **set-level infrequency** is located on the second level, which is the infrequency of a legal term set. For example, legal term set { 前項の場合において $_{(k)}$, 前項に規定する場合において $_{(l)}$} in Section 2.1.2 can be regarded as an infrequent legal term set since the legal term set occur only 3,159 times in our dataset. The set-level infrequency 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.

Three Training Techniques

We design a training scheme for our BERT classifier, where we cope with the two-level infrequency. Figure 4.2 shows the scheme. Concretely, we introduce three training techniques: preliminary domain adaptation, repetitive soft undersampling, and classifier unification. We describe each training technique below.

Preliminary Domain Adaptation: We adapt a general-purpose BERT pretrained model to statutory sentences prior to training with legal term correction examples. We consider that this technique copes with both the term-level infrequency and the set-level infrequency for the following two reasons: First, we can feed prior knowledge of 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 [90] 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 the 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 modeling 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 modeling: [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.

Repetitive Soft Undersampling: We apply an undersampling technique, which aims at the term-level infrequency. In our undersampling technique, we first apply "soft" undersampling that "weakens" the magnitude correlation among frequent classes and infrequent classes. We expect that weakening the correlation assists classifiers to predict well both of frequent class examples and infrequent class examples. Concretely, we undersample examples from the example set E_t that corresponds to a legal term $t \in T$ as much as the value of the following function:

$$s(E_t, E_{\mathrm{all}}; \beta) = |E_t| \cdot \left(\frac{\min\{|E| \mid E \in E_{\mathrm{all}}\}}{|E_t|} \right)^{\frac{1}{\beta}}, \tag{4.9}$$

where $|E|$ is the number of examples in example set E and $E_{\mathrm{all}} = \{E_t | t \in C\}$ 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. This function reduces the number of t's examples to be sampled by the ratio of the number of examples for the least frequent legal term $\min\{|E| \mid E \in E_{\mathrm{all}}\}$ to the number of t's examples $|E_t|$. Here, the hyperparameter β controls the strength of reduction. $\beta = 1$ creates naive undersampling, which biases the classifier toward the infrequent class. In contrast, large enough β (i.e., $1/\beta \approx 0$) creates no undersampling, which biases the classifier toward frequent classes. The larger the β, the weaker the undersampling, which balances infrequent and frequent classes.

Next, to cover as many examples as possible, we resample training examples from the whole dataset after certain iterations. This procedure resembles an ensemble training framework such as Bagging [6] and Boosting [24]. However, unlike them, we do not create an ensemble of BERT models since the size of a BERT model is quite huge. Algorithm 2 shows the training algorithm of our model. $\mathit{choice}(E, n)$ is a function that randomly chooses n items from the example set E. I is the number of iterations.

Algorithm 2 Repetitive soft undersampling

Input: E_{all}, β, I
Output: Model
Initialize parameters of Model
for $i \leftarrow 1$ to I **do**
  $E_{\text{sus}} \leftarrow \{\}$
  for E in E_{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_{sus}
end for

As far as we are aware, there are few cases of applying undersampling to BERT classifiers. One case is from Anand et al. [2], where majority examples are undersampled and minority examples are copied randomly. However, they did not repeat the undersampling process during training; thus they trashed a large number of majority examples, which may include important examples for classification performance. In contrast, we use repetition to cover a large part of the majority examples; therefore, our method has less chance of missing any important examples.

Classifier Unification: The Random Forest approach uses classifiers separately built for each legal term set; however, this may cause two problems. One is the underfitting problem caused by the set-level infrequency. The other, a more severe one, is a storage problem; that is, we need an enormous amount of storage to keep all classifiers, especially when we use BERT (more than 1 GB $\times$ 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 legal term sets. That is, the parameters of the classifier are shared among all legal terms so that the classifier can use broader knowledge in predicting the legal terms of an infrequent legal term set.

For the output layer, we consider two approaches: global classification and merged classification. Figure 4.3 and Figure 4.4 show our unified classifier models with global classification and merged classification, respectively. For global classification, the model outputs the 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 the 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

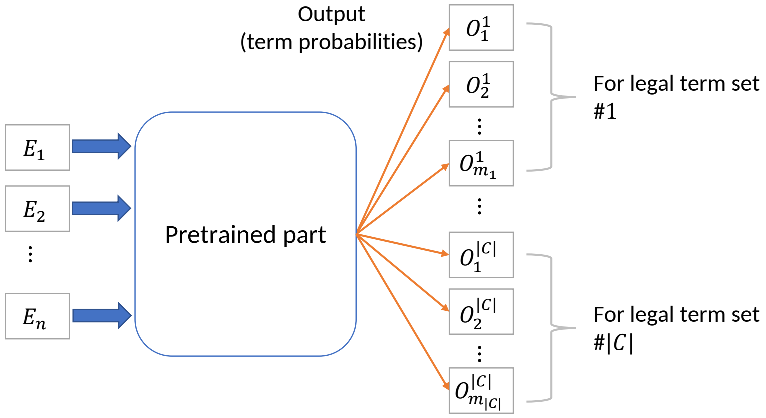

Figure 4.3: Unified BERT classifier (global classification)

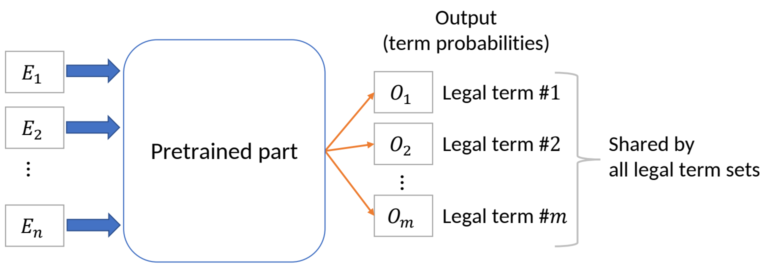

Figure 4.4: Unified BERT classifier (merged classification)

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, quux}. Then, the output layer has three units O_1, O_2, and O_3, where each should output likelihoods of "foo" and "baz," "bar" and "qux," and "quux," respectively.

4.4 Experiment

We conducted experiments on predicting legal terms in Japanese statutory sentences to examine the performance of the two proposed approaches.

4.4.1 Experimental Settings

We compiled a statutory sentence corpus from e-Gov Statute Search[1] provided by the Ministry of Internal Affairs and Communications, Japan. We acquired 3,983 Japanese acts and cabinet orders on May 18, 2018. Next, we tokenized each statutory sentence in the corpus by MeCab (v.0.996), which is a Japanese morphological analyzer. The statistics of the corpus are as follows: the total number of sentences is 1,223,084,

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

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.

We defined 69 legal terms in 27 legal term sets by referencing the Japanese legislation manual [35, 37]. Table A.1 to Table A.4 in Appendix A show all legal term sets. There are 7,072,599 and 251,085 legal term frequencies in the training data and the test data, respectively. Figure 4.5 and Figure 4.6 show the statistics of the legal term frequencies and the legal term set frequencies, respectively. Concretely, they show the counts, average, and median of the legal terms (Figure 4.5) and the legal term sets (Figure 4.6) in the entire corpus (blue) and in the test data (yellow). There is a difference between the average and the median in both legal terms and legal term sets, mainly because the occurrence of a legal term "の" (*no*; of) (in Table A.2) is quite frequent.

We compared our BERT classifier and Random Forest classifiers (abbreviated as RF) with the following classifiers and language models: CBOW [55], Skipgram [55], vLBL [57], vLBL(c) [58], vLBL+vLBL(c) (abbreviated as vLBL+) [58], and n-gram language. To test a neural-based model whose complexity is between BERT and the neural language models, we additionally tested TextCNN [44], which is a sentence classifier based on a convolutional neural network.

Our BERT classifier is based on a publicly available BERT model pretrained by Japanese Wikipedia text [2]. The model's specs are as follows: 12 Transformer layers, 768 hidden vectors, and 12 heads. It has a vocabulary of 32,000 subwords and receives a maximum of 128 tokens; therefore, we truncate each example to 128 words.

For the preliminary domain adaptation, using a script provided by the authors of BERT, we 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 (150,000 $\times$ 32 / 467,382). For the repetitive soft undersampling, we set β in Eq. (4.9) to 3 and undersample iterations I to 100. In this setting, we trained 42,721,500 examples (including duplication). For the classifier unification, we adopted the global classification. We will evaluate these settings in the training techniques in Section 4.5.2.

At each iteration of the repetitive soft undersampling, we fine-tuned the model by the following settings: the number of epochs is 5, minibatch size is 512, warmup

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

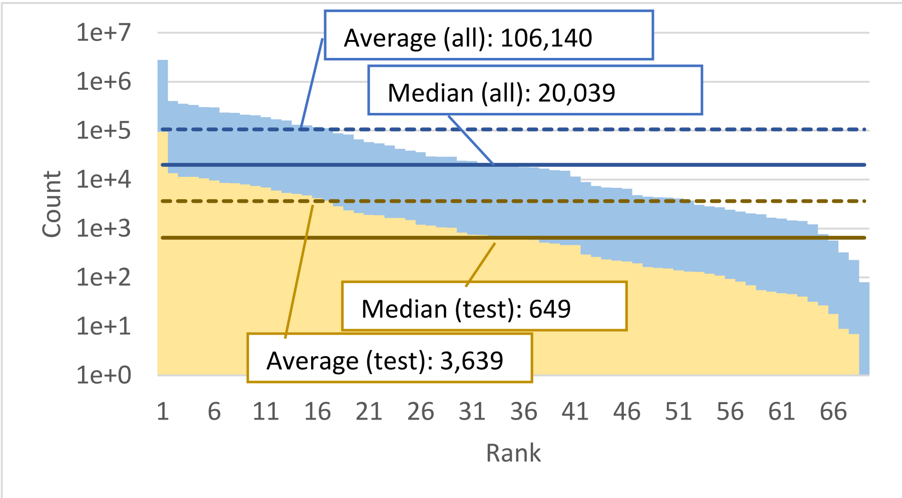

Figure 4.5: Counts of legal terms

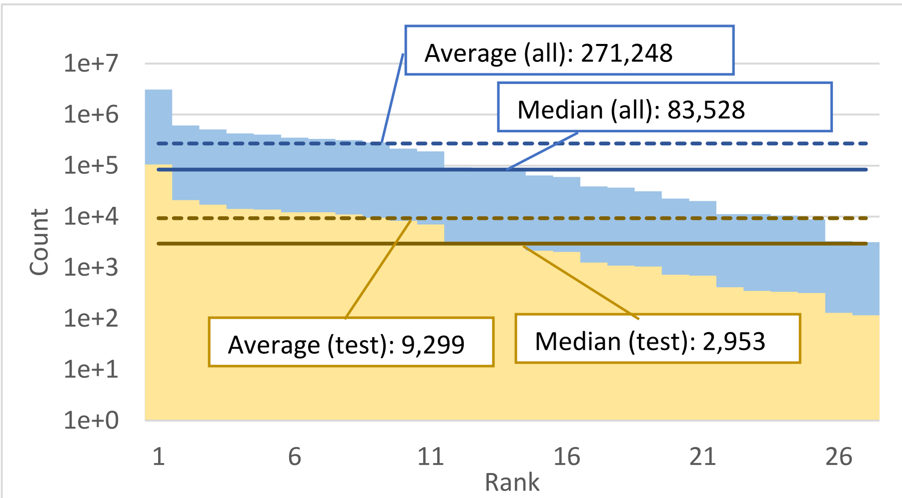

Figure 4.6: Counts of legal term sets

proportion is 0.1, and learning rate is 2e-5. We used TensorFlow on Colaboratory [3] for the implementation, training, and testing.

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, unlimited} by grid search with five-fold cross-validation. Implementation, training, and testing were done by Scikit-learn (v.0.19.1).

For TextCNN, we build a classifier for each legal term set. We set the number of vector dimensions, the sequence length, and the number of epochs to 200, 128, and 5, respectively. We implemented, trained, and tested the model on Colaboratory.

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 [45]. 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 [39], referencing Zweig and Burges [103].

Since neural language models and n-gram models are designed to predict a single word, we combined legal terms with multiple words (e.g., “前項_の_場合_において”) into single words (e.g., “前項の場合において”) by the longest match principle. We added these combined legal terms to the vocabulary. The total number of tokens in the corpus thus becomes 45,213,528. The word counts in Table A.1 to Table A.4 in Appendix A reflect this operation. 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.

4.4.2 Comparison in Classifiers

Table 4.1 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 by maximum likelihood estimation (MLE), in which the most frequent legal terms in the training 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 $acc_{\text{macro-T}}$ is 92.56%, which is 7.88 points better than Random Forest. TextCNN achieved the second-best performance in all of the criteria

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

Table 4.1: Overall performance of legal term correction

Classifier	acc_{micro}	$acc_{\text{macro-S}}$	$acc_{\text{macro-T}}$
BERT (approach 2)	**97.57%**	**96.15%**	**92.56%**
RF (approach 1)	95.37%	93.22%	84.68%
TextCNN	95.99%	94.12%	86.28%
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+	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%

as it performed better than Random Forest.

Next, we evaluated the accuracies of minority and majority legal terms. Here, we considered three types of partitioning as follows:

- **Interset term partitioning** (interset), which separates majority and minority legal terms in all legal terms regardless of legal term sets. We separated them by the median of counts of all legal terms (649 times or lesser as the minority). Accuracies of the minority legal terms in this partitioning indicate the overall improvement of the infrequencies.

- **Intraset term partitioning** (intraset), which separates majority and minority legal terms in its belonging legal term set. We regard the least frequent legal term and the most frequent legal term in a legal term set as the minority and the majority, respectively. Accuracies of the minority legal terms in this partitioning indicate improvement of the term-level infrequency.

- **Set-level partitioning** (set), which separates majority and minority legal term sets in all legal term sets. We separated minority and majority legal term sets by the median frequency among all legal term sets (2,953 times or lesser as the minority). Accuracies of the minority legal term sets indicate improvement of the set-level infrequency.

Table 4.2 shows the macro-average accuracies of the majority legal terms acc_{majority} and minority legal terms acc_{minority} in the interset partitioning, 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 other

Table 4.2: Accuracy of legal term correction by interset majority and minority

Classifier	acc_{majority}	acc_{minority}	Difference
BERT	**96.15%**	**89.06%**	7.09
RF	91.27%	78.27%	13.00
TextCNN	93.47%	79.29%	14.18
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+	84.47%	77.86%	6.61
Trigram	78.15%	60.82%	17.33
4-gram	80.82%	64.57%	16.25

Table 4.3: Accuracy of legal term correction by intraset majority and minority

Classifier	acc_{majority}	acc_{minority}	std_{minority}
BERT	**97.36%**	**87.40%**	18.89%
RF	96.11%	72.69%	26.62%
TextCNN	97.02%	75.05%	24.76%
CBOW	89.26%	60.95%	29.45%
Skipgram	63.42%	70.74%	30.70%
vLBL	76.44%	73.64%	29.45%
vLBL(c)	88.03%	72.99%	31.19%
vLBL+	86.06%	76.89%	29.26%
Trigram	93.06%	45.34%	37.25%
4-gram	94.12%	49.81%	35.59%

methods. By comparing the difference between the two accuracies, BERT has less difference than Random Forest and n-gram, along with Skipgram and the vLBL family.

Table 4.3 and Table 4.4 show the macro-average accuracies by the intraset partitioning and the set partitioning, respectively. std_{minority} in each table means the standard deviation of the accuracies of the minority legal terms/sets. This measure indicates the prediction performance stability for minority terms or sets. BERT also achieved the best performance in both the partitioning and in both the majority and minority. In particular, it improved acc_{minority} in intraset by more than 10 points compared to Random Forest. Also, std_{minority}s of the BERT classifier are smallest in the two tables. From these results, our BERT classifier coped with both the term-level infrequency and the set-level infrequency.

Finally, we look at the accuracies for particular legal term sets. Table 4.5 shows the accuracy of a legal term set $\{$ 者$_{\text{(a)}}$, 物$_{\text{(b)}}$, もの$_{\text{(c)}}\}$. The BERT classifier achieved

Table 4.4: Accuracy of legal term correction by set majority and minority

Classifier	acc_{majority}	acc_{minority}	std_{minority}
BERT	**96.78%**	**95.40%**	6.01%
RF	95.35%	91.25%	9.40%
TextCNN	95.98%	92.40%	8.75%
CBOW	88.05%	81.49%	16.46%
Skipgram	74.57%	52.39%	28.70%
vLBL	83.35%	68.14%	27.70%
vLBL(c)	89.89%	83.00%	13.47%
vLBL+	89.57%	81.95%	15.12%
Trigram	86.07%	85.58%	13.46%
4-gram	88.30%	87.39%	12.90%

the best accuracy in every legal term in this legal term set. It well predicted the legal term 物$_{\text{(b)}}$, the least frequent legal term in the set, while the Random Forest classifier dropped its accuracy. Table 4.6 shows the accuracy of a legal term set {直ちに$_{\text{(h)}}$, 速やかに$_{\text{(i)}}$, 遅滞なく$_{\text{(j)}}$}. The BERT classifier achieved the best accuracy in acc_{micro}, followed by the Random Forest classifier. It appears that the broader context contributed to predicting these legal terms accurately. The BERT classifier handles a larger context (i.e., 128 tokens) better than the Random Forest classifier (20 tokens for this legal term set). In contrast, the n-gram classifiers that utilize a few previous words achieved poor accuracies, especially in minor legal terms (h) and (i). Since it is difficult to identify the adequate legal term with a few previous words in this legal term set, the classifiers seem to have heavily relied upon the frequencies of the legal terms. Actually, the trigram classifier predicted 84.15% (69/82) of term (h) as term (j) and 80.00% (88/110) of term (i) as term (j), where term (j) is the most frequent legal term. Table 4.7 shows the accuracy of a legal term set {要求$_{\text{(m)}}$ (*yokyu*; requirement), 要請$_{\text{(n)}}$ (*yosei*; request)}, which is the least frequent legal term set. The BERT classifier was best in overall accuracy and the accuracy for the minority legal term (m). In this legal term set, the Random Forest classifier did not work outstandingly, that is, the language model-based methods tended to work well. One possible reason is that these methods dealt with the set-level infrequency as our BERT classifier did because their language models are trained by whole sentences regardless of legal term sets and are uniformly used for all legal term sets.

For roundup of assessing accuracies of particular legal term sets, Table 4.8 shows the number of legal term classes and legal term set classes that each classifier (except MLE) predicted best. BERT classifier has the largest number of best predictions both in legal terms and legal term sets. It achieved the best accuracy in 23 out of 27 legal

Table 4.5: Accuracy of legal term set { 者$_{\text{(a)}}$, 物$_{\text{(b)}}$, もの$_{\text{(c)}}$}

Classifier	者$_{\text{(a)}}$	物$_{\text{(b)}}$	もの$_{\text{(c)}}$	acc_{micro}
BERT	**98.50%**	**98.25%**	**97.35%**	**98.03%**
RF	95.46%	88.80%	92.84%	94.05%
TextCNN	97.70%	96.68%	95.47%	96.75%
CBOW	86.59%	86.88%	85.71%	86.25%
Skipgram	76.26%	81.28%	74.41%	75.79%
vLBL	78.98%	94.84%	88.90%	83.81%
vLBL(c)	88.81%	97.29%	94.02%	91.36%
vLBL+	92.22%	97.11%	92.25%	92.50%
Trigram	84.97%	92.39%	85.37%	85.54%
4-gram	92.53%	93.35%	87.07%	90.39%
Freq.	11,440	1,143	8,389	

Table 4.6: Accuracy of legal term set { 直ちに$_{\text{(h)}}$, 速やかに$_{\text{(i)}}$, 遅滞なく$_{\text{(j)}}$}

Classifier	直ちに$_{\text{(h)}}$	速やかに$_{\text{(i)}}$	遅滞なく$_{\text{(j)}}$	acc_{micro}
BERT	68.29%	**62.73%**	90.05%	**78.45%**
RF	68.29%	37.27%	93.67%	73.61%
TextCNN	40.24%	33.64%	95.93%	68.28%
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+	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%
Freq.	82	110	221	

Table 4.7: Accuracy of legal term set {要求$_{\text{(m)}}$, 要請$_{\text{(n)}}$}

Classifier	要求$_{\text{(m)}}$	要請$_{\text{(n)}}$	acc_{micro}
BERT	**79.17%**	92.75%	**87.18%**
RF	66.67%	79.71%	74.36%
TextCNN	68.75%	88.41%	80.34%
CBOW	58.33%	92.75%	78.63%
Skipgram	58.33%	59.42%	58.97%
vLBL	54.17%	98.55%	80.34%
vLBL(c)	60.42%	94.20%	80.34%
vLBL+	64.58%	**100.00%**	85.47%
Trigram	72.92%	85.51%	80.34%
4-gram	77.08%	94.20%	**87.18%**
Freq.	48	69	

Table 4.8: Number of best predictions by classifier

Classifier	# Legal terms	# Legal term sets
BERT	**32**	**23**
RF	13	4
TextCNN	18	4
CBOW	5	0
Skipgram	7	0
vLBL	6	0
vLBL(c)	10	2
vLBL+	11	0
Trigram	9	1
4-gram	6	2
Total classes	69	27

term sets, which is outstanding. Compared to the best predictions in legal term sets, the number of best predictions in legal terms is low. We reason this from the fact that other classifiers often have strong biases to frequent legal terms. One case we saw is "遅滞なく$_{(j)}$", which Trigram classifier predicted best while it predicted other legal terms poorly.

4.5 Discussion

In this section, we evaluate the characteristics of the two proposed approaches. In Section 4.5.1, we discuss that of the Random Forest approach. In Section 4.5.2, we discuss that of the BERT approach.

4.5.1 Evaluation of Random Forest Approach

In this section, we discuss the experimental result of Random Forest classifiers to reveal its characteristics for the legal term correction task. First, we investigate the window size selection to reveal its effectiveness. Next, we analyze the feature importance calculated by classifiers to find how the classifiers utilize context in each legal term set.

Effectiveness of Adaptive Window Size Selection

In this section, we discuss the effectiveness of the adaptive window size selection. Concretely, we look at the following topics:

- Tendency of selected window sizes;

Table 4.9: Window sizes and their corresponding legal term sets

WS	Legal term set ID
2	1, 6, 8, 13, 15, 19
5	2, 3, 4, 5, 7, 9, 12, 18, 20, 22, 25
10	11, 14, 17, 21, 23, 24
15	10, 16, 26, 27

Table 4.10: Accuracy of legal term correction per window size selected

WS	# legal terms	Fixed-window RF	Adaptive-window RF
2	140,098	98.15%	**98.25%**
5	99,360	**95.42%**	95.40%
10	5,310	91.01%	**91.79%***
15	6,317	93.51%	**96.15%***

- Comparison of fixed window size and adaptive window size;
- Optimality of the window size selection done by grid search.
- Relationship between window size in a neural language model and its performance.

Tendency of Selected Window Sizes: First, we look at the tendency of window sizes selected by the method. By applying the grid search settings described in Section 4.4.1, the method chose the window size shown in Table 4.9 for each legal term set. Here, WS in the table means window size and IDs in the table correspond to the legal term set IDs defined in the appendix. Table 4.9 proves that all window sizes are used. Among the four window sizes, window size 5 was the most frequently selected, which indicates the rationality of choosing that window size for neural language models.

Comparison of Fixed Window Size and Adaptive Window Size: Next, to investigate the effectiveness of adaptive window size selection, we classified legal term sets by window size that the method selected for the legal term set. We then calculated accuracies (micro averaged) of each window size class from two methods: Random Forest with adaptive window size selection (adaptive-window RF; the proposed approach) and Random Forest with the fixed window size of 5 for every legal term set (fixed-window RF) Table 4.10 shows the result. * indicates there was a significant difference ($p < 0.05$) between adaptive-window RF and fixed-window RF. Adaptive-window RF achieved significantly high performance than fixed-window RF

Table 4.11: Accuracy of vLBL+vLBL(c) in different window sizes

WS	acc_{micro}	$acc_{\text{macro-s}}$	$acc_{\text{macro-w}}$
2	89.82%	82.46%	78.24%
5	90.95%	85.62%	81.12%
10	**92.39%**	84.64%	81.28%
15	92.34%	**86.63%**	**81.50%**
Optimal	93.05%	89.47%	83.76%
Adaptive-window RF	95.20%	92.86%	83.57%
Fixed-window RF	95.37%	93.22%	84.68%

in window sizes 10 and 15. From this result, we claim the effectiveness of using a longer context for the legal term sets belonging to these classes. Also, although it is not significant, 0.10 points of improvement were observed in window size 2. 0.02 points of degradation were in window size 5, which seems to be caused by random noise because of its insignificance.

Optimality of the Window Size Selection done by Grid Search: Next, we evaluate the optimality of window size selection by adaptive-window RF. We summed up the number of legal term sets whose selected window size agrees with its optimal window size. As a result, adaptive-window RF chose optimal window sizes in 81.4% (22/27) of legal term sets. A finding is that disagreement tended to happen in legal term sets with fewer examples such as ID10 ({前項に規定する場合において, 前項の場合において}, 3,159 examples) ID21 ({直ちに, 速やかに, 遅滞なく}, 11,192 examples). Therefore, adding new examples will lead to improvement. We then compared the prediction accuracy of adaptive-window RF and optimal-window RF (i.e., RF using the optimal window sizes). As a result, adaptive-window RF and optimal-window RF are achieved 95.37% and 95.40%, respectively, whose difference was only 0.03 points.

Window Size in Neural Language Model and its Performance: Finally, we investigated accuracies of vLBL+vLBL(c) in different window sizes, shown in Table 4.11. The model achieved the best acc_{micro} in window size 10, and the best $acc_{\text{macro-s}}$ and $acc_{\text{macro-w}}$ in window size 15. However, every setting achieved worse performance than the Random Forest method. Even if we give the optimal window sizes to the classifier, acc_{micro} was 93.05%, which is lower than the Random Forest.

Feature Importance Analysis

As we discussed in Section 2.2.2, feature importances are calculated during building a Random Forest classifier. Here, we can find the importance of context position (more concretely, relative word position) for prediction, because the method uses relative words as features.

Table 4.12 shows such feature importances in each legal term set. Relative position is a word position centered by the position of the target legal term. For example, relative position -1 indicates the very previous word of the legal term. The sum of feature importances of each legal term set will be 1. We can categorize four types of tendency from Table 4.12:

(1) Strong dependency to nearest previous words: ID1 ({規定, 規程}), ID2 ({場合, とき, 時}), ID15 ({この限りでない, 妨げない}) belong this category. As for ID1, "規定" (provision) often appears with a postposition like "の" or a verb like "掲げる" (listed), because the legal term tends to be used with the number of the provision (e.g., "第三条の規定" (the provisions in Article 3) "前号に掲げる規定" (the provisions listed in the preceding item)). Actually, 78.0% (302,191 examples) of the preceding word of "規定" were "の." On the other hand, "規程" (rules) tends to be used with the target of regularization (e.g., "業務規程" (operational rules) and "事務規程" (administration rules)). Therefore, in many cases, we can earn much evidence for judgment by focusing on the previous word. However, "に関する" (regarding) can be placed in both cases like "経過措置に関する規定" (provisions regarding the transitional measures) and "業務の運営に関する規程" (rules regarding business operation); therefore, further words were given with some amount of importance.

(2) Dependency to previous words: Many legal term sets belonging to this category are verbal legal terms sets, for example, ID11 ({推定する, みなす}) and ID14 ({することができる, しなければならない, するものとする}). This outcome is natural because verbs appear at the end of phrases in Japanese sentences.

(3) Dependency to following words: Some of the instances are ID26 ({前項の場合において, 前項に規定する場合において}), ID27 ({ただし, この場合において}). These legal terms are conjunctions or conjunctional phrases that supplement the previous sentence. Therefore, the following words are clues for prediction in many cases.

Table 4.12: Feature importances in each legal term set

ID	Relative position -15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.07	0.46	0.03	0.04	0.02	0.02	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
2	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.04	0.04	0.42	0.14	0.09	0.03	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
3	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.05	0.06	0.16	0.08	0.07	0.09	0.04	0.03	0.03	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01
4	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.02	0.03	0.03	0.03	0.04	0.05	0.07	0.07	0.13	0.05	0.05	0.04	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02
5	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.07	0.08	0.10	0.07	0.07	0.04	0.03	0.03	0.02	0.03	0.02	0.02	0.01	0.01	0.01	0.01	0.01
6	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.04	0.05	0.05	0.07	0.10	0.13	0.07	0.04	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
7	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.03	0.05	0.04	0.05	0.07	0.08	0.10	0.06	0.03	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.05	0.08	0.12	0.08	0.16	0.18	0.09	0.03	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00
9	0.01	0.00	0.00	0.01	0.01	0.01	0.01	0.02	0.00	0.03	0.02	0.04	0.05	0.08	0.21	0.10	0.12	0.10	0.05	0.03	0.03	0.03	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.00
10	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.05	0.08	0.12	0.22	0.02	0.03	0.04	0.04	0.02	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00
11	0.06	0.05	0.06	0.06	0.06	0.07	0.06	0.06	0.09	0.11	0.11	0.10	0.04	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.07	0.12	0.14	0.21	0.12	0.05	0.03	0.03	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00
13	0.04	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.05	0.06	0.05	0.08	0.12	0.07	0.07	0.03	0.03	0.03	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00
14	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.08	0.08	0.14	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.01	0.00	0.06	0.09	0.39	0.38	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.01	0.01	0.02	0.03	0.02	0.03	0.03	0.04	0.19	0.05	0.13	0.03	0.12	0.02	0.02	0.01	0.06	0.01	0.01	0.04	0.01	0.01	0.01
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.06	0.06	0.22	0.18	0.14	0.15	0.03	0.02	0.02	0.01	0.05	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.03	0.11	0.12	0.05	0.04	0.04	0.04	0.03	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02
19	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.03	0.03	0.03	0.04	0.05	0.07	0.23	0.10	0.06	0.04	0.03	0.03	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01
20	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.05	0.05	0.15	0.12	0.09	0.05	0.04	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
21	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.03	0.03	0.02	0.02	0.09	0.04	0.04	0.04	0.04	0.04	0.03	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02
22	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.04	0.03	0.06	0.05	0.06	0.08	0.05	0.04	0.04	0.03	0.04	0.03	0.03	0.03	0.03	0.03	0.03	0.02	0.02	0.03	0.02
23	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.05	0.04	0.05	0.04	0.04	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
24	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.04	0.06	0.04	0.06	0.06	0.06	0.05	0.04	0.05	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.02	0.02	0.02
25	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.04	0.06	0.13	0.08	0.07	0.05	0.04	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02
26	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.00	0.01	0.08	0.06	0.06	0.06	0.07	0.06	0.08	0.07	0.06	0.07	0.06	0.06	0.06	0.06
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.03	0.06	0.13	0.07	0.08	0.06	0.06	0.05	0.05	0.07	0.06	0.05	0.04	0.05	0.05	0.04

Table 4.13: Effectiveness of preliminary domain adaptation

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

(4) Even dependency to previous and following words: Many legal term sets belong to this category, for example, ID3 ({ 者, 物, もの }), ID9 ({ 科する, 処する, 課する }), ID18 ({ 当該, その }), ID23 ({ 又は, 若しくは }). Different from legal term sets of type (1), these sets do not have strong clues to distinguish in the nearest words. Also, they do not tend to appear at the beginning or end of the sentence. Therefore, the classifier seems to need to look at whole the context.

From these results, we found out that Random Forest classifiers are trained with careful attention to the context around legal terms.

4.5.2 Evaluation of BERT Approach

In this section, we evaluate the effects of the proposed three techniques.

Preliminary Domain Adaptation

To evaluate the effect of the preliminary domain adaptation, we first compare the overall accuracy of the domain-adapted BERT model and the pretrained BERT model. Table 4.13 shows the acc_{micro}, acc_{majority} (macro average), and acc_{minority} (macro average) between the domain-adapted model and the pretrained model. We used the interset partitioning for acc_{majority} and acc_{minority} since the preliminary domain adaptation is a technique for overall performance. From this result, we find that preliminary domain adaptation contributed to overall improvement.

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 4.7 shows the transition of accuracy in the domain-adapted BERT model and the pretrained BERT model. The domain-adapted model achieved better performance overall, especially in the 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% that is 0.29 points worse. Thus, the preliminary domain adaptation accelerates convergence.

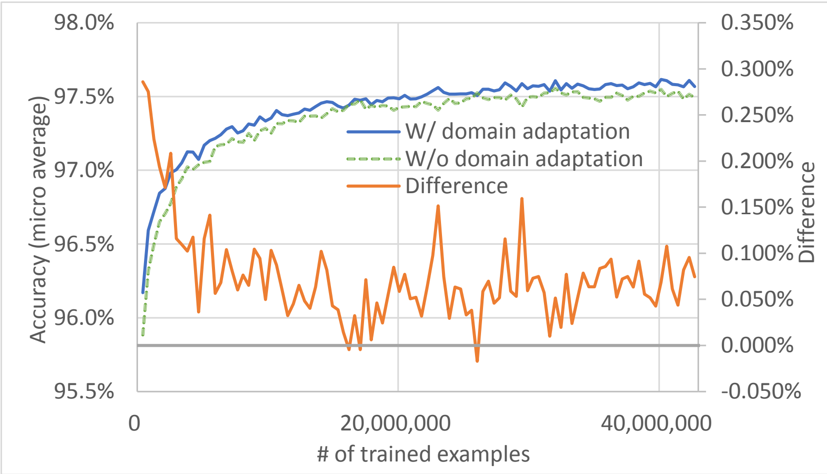

Figure 4.7: Transition of Accuracy in the domain-adapted model and the pretrained model

Repetitive Soft Undersampling

To evaluate the effect of soft undersampling, we made BERT classifiers with different β values the undersampling strength. Here, we applied $\beta \in \{1, 2, 3, 5, 10, 50, 100, \infty\}$. Each BERT model trained approximately 10,000,000 examples in total, where at most 500,000 examples were sampled in each iteration.

Table 4.14 shows acc_{micro}, acc_{majority} (macro average), and acc_{minority} (macro average) in the different β. We used the intraset partitioning for acc_{majority} and acc_{minority} since the repetitive soft undersampling is a technique for term-level infrequency. The model with $\beta = 5$ performed best in acc_{micro} and acc_{majority}, while the one with $\beta = 1$ did best in acc_{minority}. The gap in acc_{minority} is large (6.58 points between the best and worst one) compared to the gap in acc_{majority} among the models (1.23 points). Also, a small β contributes to better acc_{minority}, which is consistent with the fact that a smaller β brings about stronger undersampling. On the other hand, acc_{majority} of the model with $\beta = 1$ is the worst. Therefore, setting a moderate β promotes good performance for both majority legal terms and minority legal terms.

However, there is a problem with the result in Table 4.14. The model with $\beta = 1$ could correctly answer the question of a legal term "に関係する$_{(o)}$" (*ni kankeisuru*; regarding), whereas others could not, and that result blurred the overall accuracy of the minority legal terms. As Table A.2 indicates, we have only one example for the legal term in the test data. Therefore, success in answering the example is a big bonus; from another perspective, it excessively raises acc_{minority}.

To solve this problem, we introduced a fixed acc_{minority} in which term (o) is not included. Table 4.15 shows the fixed acc_{minority} between different

Table 4.14: Effectiveness of soft undersampling

β	acc_{micro}	acc_{majority}	acc_{minority}
1	95.26%	96.09%	**89.73%**
2	97.20%	96.99%	87.89%
3	97.35%	97.14%	87.95%
5	**97.40%**	**97.32%**	86.58%
10	97.37%	97.22%	84.16%
50	97.31%	96.91%	84.03%
100	97.31%	97.28%	83.39%
∞	97.32%	97.11%	83.15%

Table 4.15: Legal term correction result of fixed acc_{minority}

β	Fixed acc_{minority}
1	89.33%
2	91.27%
3	**91.33%**
5	89.91%
10	87.39%
50	87.26%
100	86.60%
∞	86.35%

$\beta \in \{1, 2, 3, 5, 10, 50, 100, \infty\}$. In this criterion, the model with $\beta = 1$ achieved the fourth-best accuracy.

To assess the stability of the prediction for minority legal terms, we investigated the transition of the fixed acc_{minority} in different β and training amounts. Figure 4.8 shows the result. According to the graph, a smaller β tends to have contributed to better accuracy for minorities since the early iterations.

Next, we evaluate the effect of repetitive undersampling. Figure 4.9 shows the transition of the overall accuracy acc_{micro} in a BERT classifier with repetitive undersampling and one without repetitive undersampling. In the latter model, we sampled examples once and then used the same examples throughout all iterations. We applied soft undersampling with $\beta = 3$ to both models. From Figure 4.9, it is clear that the model with repetitive undersampling raised its accuracy as iterations proceeded, while the accuracy for the model without repetitive undersampling slightly degraded. We believe this deterioration came from the overfitting of the training examples. In summary, we found that the repetitive undersampling worked favorably for better accuracy.

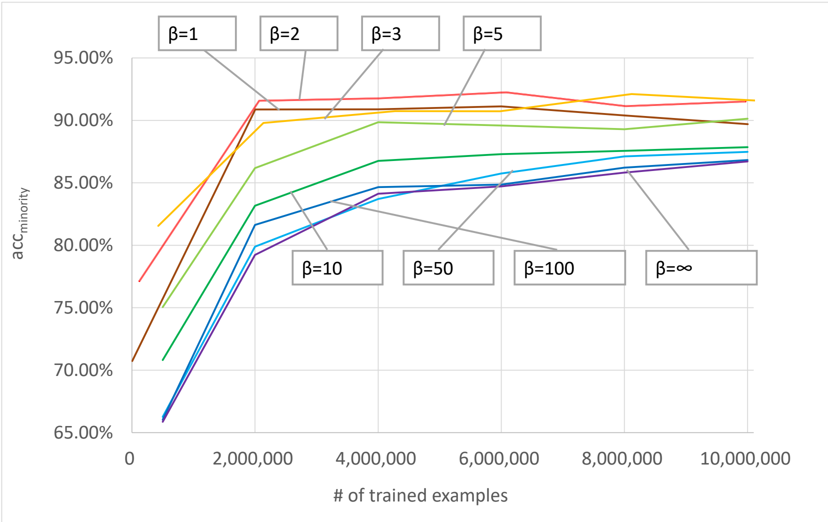

Figure 4.8: Transition of acc_{minority} in different β (sampled per approx. 2 M training examples)

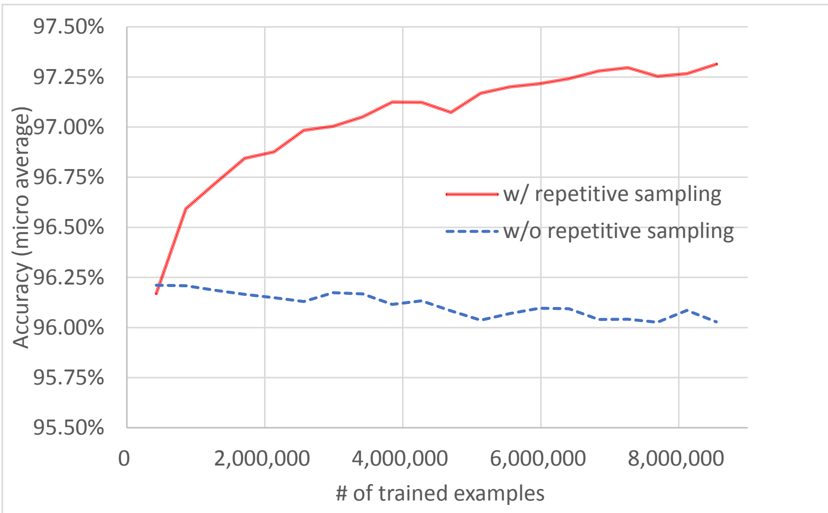

Figure 4.9: Transition of acc_{micro} in the model with repetitive undersampling and model without repetitive undersampling

Table 4.16: Effectiveness of classifier unification

Applied?	acc_{micro}	acc_{majority}	acc_{minority}
Yes	97.32%	96.38%	**94.10%**
No	**97.39%**	**96.81%**	93.74%

Table 4.17: Effectiveness of classifier unification in an infrequent legal term set {要求$_{\text{(m)}}$, 要請$_{\text{(n)}}$}

Applied?	要求$_{\text{(m)}}$	要請$_{\text{(n)}}$	acc_{micro}
Yes	**66.67%**	89.86%	**80.34%**
No	55.32%	**92.75%**	77.59%
Freq.	48	69	

Classifier Unification

To evaluate the effectiveness of the classifier unification, we made a separate BERT classifier for each legal term set and compared it with the unified BERT classifier. Here, both models are domain-adapted. We set $\beta = \infty$ to the unified classifier to exclude effects from soft undersampling. We fed 10 M examples (500 k examples $\times$ 20 resamplings) to the unified model.

Table 4.16 shows acc_{micro}, acc_{majority} (macro average), and acc_{minority} (macro average) of the unified BERT classifier (proposed) and the separate BERT classifiers. We used set partitioning for acc_{majority} and acc_{minority} since the classifier unification is a technique for set-level infrequency. Although the separate BERT classifier achieved better accuracies in the overall and majority sets, the unified one achieved better performance in the minority set at 0.36 points above. As for improvement in an infrequent legal term set, Table 4.17 shows the accuracies of legal term set {要求$_{\text{(m)}}$, 要請$_{\text{(n)}}$}. This legal term set has only 3,277 examples in the whole dataset, which is 1/100 of average occurrence. The separate classifier achieved only 55.32% accuracy in term (m), the infrequent legal term, while the unified classifier achieved 66.67% accuracy that is 11.35 points better. In summary, the classifier unification had a certain effect that improved performance for minority legal term sets.

Next, we compare the global classification and the merged classification. Table 4.18 shows the accuracy of global classification and merged classification. We used set partitioning for acc_{majority} and acc_{minority}. In both cases, we used the repetitive soft undersampling with $\beta = 3$ and trained 43 M examples in total (427,215 examples $\times$ 100 resamplings). The global classification overall performed better, specifically, 2.14, 2.77, and 3.81 points better in accuracy in acc_{micro}, acc_{majority}, and acc_{minority}, respectively. It appears that sharing output positions caused a disturbance in train-

Table 4.18: Comparison of global classification and merged classification

Type	acc_{micro}	acc_{majority}	acc_{minority}
Global	**97.57%**	**96.78%**	**95.40%**
Merged	95.43%	94.01%	91.59%

ing 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.

4.6 Summary

In this chapter, we tackled the legal term correction task. First, we defined the task as a special case of the sentence completion test. Then, we proposed two approaches: the Random Forest approach and the BERT approach. As for the first approach, we applied Random Forest classifiers. Concretely, each classifier is trained for its responsible legal term set with optimal parameters. The experiment showed that the Random Forest method outperforms the light-weighted neural language models and that it brings interesting findings on selected window sizes and attended context in each legal term set. As for the second approach, we applied a BERT classifier. Here, we claimed the two-level infrequency and three training techniques for it, namely, preliminary domain adaptation, repetitive soft undersampling, and classifier unification. The experiment demonstrated that the BERT classifier outperforms all the classifiers and the three training techniques are effective for the problems ascribed to the two-level data infrequency.

A BERT classifier garners linguistic knowledge through pretraining with a huge amount of text, which is not easily applicable to Random Forest classifiers. Also, it can flexibly capture relationships between distant words by its self-attention mechanism. On the other hand, Random Forest classifiers will suffer distant contexts because it treats relative positions discretely. From the experimental results and these facts, we insist that a BERT classifier is a more adequate classifier for this task.

Chapter 5

Application of Legal Term Correction to Thai Statutory Sentences

In this chapter, we describe an attempt to apply the legal term correction methodology to Thai statutory sentences. Like Japanese legislation, Thai legislation also has rules for distinct usage of some mistakable legal terms. Through this attempt, we learn whether we can apply the methodology for Japanese statutory sentences to Thai statutory sentences in the same manner.

First, we overview the background of legislation and legal terms of Thailand in Section 5.1. In Section 5.2, we learn several sets of Thai legal terms that have distinct usage. In Section 5.3, we introduce our proposed method for Thai legal terms. Our method receives three additional features to cope with the characteristics of Thai legal terms. In Section 5.4, we conduct experiments and then we discuss the experimental result. Finally, we conclude this study in Section 5.5.

5.1 Introduction

Legislative drafting requires careful scrutiny. Not only Japan but also every country regards this requirement as necessary in its legislation. To meet this requirement, for example, Thailand stipulates legislation rules [67], which is publicized by the Office of the Council of State, the bill examining authority. The rules define allowable usage of similar legal terms, which is similar to the situation in Japanese legislation.

For example, there are two similar Thai legal terms " อย่างหนึ่งอย่างใด" (*yang-nueng-yang-dai*; lit. thing-one-thing-any) and " อย่างใดอย่างหนึ่ง" (*yang-dai-yang-nueng*; 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 [67]. Drafters must not misuse any legal term in statutory sentences; otherwise, they 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.

In the previous chapter, we have established a methodology of legal term correction, where we focused on Japanese legal terms. However, it is not trivial that this methodology is directly applicable to Thai legal terms. Actually, we have found that the usage of some Thai legal terms is changed over time (hereinafter we call this **year dependency**) and such legal terms have domains where they likely to appear (**genre dependency**), which cannot be seen in that of Japanese legal terms. Also, we have observed Thai legal terms being used in an item with a few adjacent words (**little context**), which is problematic for our methodology that utilizes adjacent words as context.

We, therefore, attempt to establish a legal term correction method optimized for Thai statutory sentences. In the same manner as the previous chapter, 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 [7] to score the likelihood of each candidate for the legal term. To cope with the characteristics of Thai legal terms, we newly introduce additional features from outside of the statutory sentence, namely, year, title keyword, and section keyword. We expect that the year feature copes with year dependency, the title keyword feature copes with genre dependency, and the section keyword feature helps classifiers identifying legal terms with little context.

Section 100 Any person who has x of the following attributes must not be a chairman of the committee:
(1) Being a stakeholder of the company;
(2) Being a worker of the company;
(3) Being a political servant.

Figure 5.1: A typical section with items in Thai statutes

5.2 Thai Legal Terms

In this section, we explain several sets of Thai legal terms whose usage is defined in the Thai legislation manual [67].

5.2.1 *Yang-nueng-yang-dai* and *yang-dai-yang-nueng*

"อย่างหนึ่งอย่างใด" (*yang-nueng-yang-dai*) is literally "thing-one-thing-any," while "อย่างใดอย่างหนึ่ง" (*yang-dai-yang-nueng*) 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.

Let us explain with the passage in Figure 5.1. Both "*yang-nueng-yang-dai*" and "*yang-dai-yang-nueng*" can be placed at the position x. When we use "*yang-nueng-yang-dai*" at x, the passage prohibits the person being the chairman of the committee if the person meets at least one of the three attributes. On the other hand, when we use "*yang-dai-yang-nueng*" at x, the passage prohibits the person being the chairman if the person meets just one of the three attributes. In other words, if the person was not only a worker of the company but also a political servant, the passage should not prohibit the person.

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

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

"อำนาจหน้าที่" (*amnat-nathi*), "อำนาจและหน้าที่" (*amnat-lae-nathi*), and "หน้าที่และอำนาจ" (*nathi-lae-amnat*) consist of "อำนาจ" (*amnat*; power), "หน้าที่" (*nathi*; duty), and "และ" (*lae*; 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 [87]. 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, which is an example of year dependency. Also, the constitution of Thailand has used only "*amnat-nathi*", which is an example of genre dependency.

5.2.3 *Panakngan-chaonathi* and *chaonathi*

Both " พนักงานเจ้าหน้าที่" (*Panakngan-chaonathi*; competent authority [87]) and " เจ้าหน้าที่" (*chaonathi*; 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.

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

Both " ข้าราชการการเมือง" (*kharachakan-kanmueang*; lit. official-politics) and " ผู้ดำรง-ตำแหน่งทางการเมือง" (*Phu-damrong-tamnaeng-thang-kanmueang*; 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.

5.3 Proposed Method

In this section, we show our proposed method for the legal term correction task. Our method is based on the methodology in Section 4.3.1 that we use Random Forest as a scoring function. Unlike that method, our method introduces three additional features from outside of the sentence to augment prediction performance. We describe these features in Section 5.3.1, followed by our prediction model in Section 5.3.2.

1 มาตรา ๒๗ ผู้มีลักษณะอย่างใดอย่างหนึ่งดังต่อไปนี้ ต้องห้ามมิให้เป็นประธานกรรมการหรือกรรมการ คือ

2 (๑) มีส่วนได้เสียในสัญญากับการรถไฟแห่งประเทศไทย หรือในกิจการที่กระทำให้แก่การรถไฟแห่งประเทศไทย ทั้งนี้ ไม่ว่าโดยตรงหรือโดยทางอ้อม เว้นแต่จะเป็นเพียงผู้ถือหุ้นของบริษัทที่กระทำการอันมีส่วนได้เสียเช่นว่านั้น

3 (๒) เป็นพนักงานของการรถไฟแห่งประเทศไทย

4 (๓) เป็นข้าราชการการเมือง

5 (๔) ขาดคุณสมบัติหรือมีลักษณะต้องห้ามตามกฎหมายว่าด้วยคุณสมบัติมาตรฐาน สำหรับกรรมการ และพนักงานรัฐวิสาหกิจ

Figure 5.2: A Thai legal term (underlined) with few adjacent words

5.3.1 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 year dependency. 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.

- **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. The use of Thai legal terms has more genre dependency than Japanese legal terms. One example is that the constitution of Thailand has used only "*amnat-nathi*" and has not used "*amnat-lae-nathi*" or "*nathi-lae-amnat*". Another example is "*Panakngan-chaonathi*", which may be related with a minister.

- **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 little context. Figure 5.2 demonstrates such an example. In the case of Figure 5.2, "ข้าราชการการเมือง" (*kharachakan-kanmueang*) 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

Algorithm 3 Our algorithm

Require: W, y, K^t, K^s, T
Ensure: Suggests
  Suggests $\leftarrow \emptyset$
  for all (i, j) such that $w_i\ w_{i+1} \dots w_j = t \in T$ **do**
    $W^l \leftarrow w_1\ w_2 \dots w_{i-1}$
    $W^r \leftarrow w_{j+1}\ w_{j+2} \dots w_{|W|}$
    $t_{\text{best}} \leftarrow \arg\max_{t' \in T} \text{score}(W^l, t', W^r, y, K^t, K^s)$
    if $t \neq t_{\text{best}}$ **then**
      Suggests $\leftarrow$ Suggests $\cup$ {suggestion that t in position (i, j) should be replaced into t_{best}}
    end if
  end for

case, the sentence in line 1 is the main sentence of the section (มาตรา; matra), so that keywords of this sentence are used as the section keyword feature.

5.3.2 Prediction Model

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

- Statutory sentence $W = w_1\ w_2 \dots w_{|W|}$, 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 bool vectors that indicate the existence of the keywords in the title and the section, respectively;

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

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

$$\text{score}(W^l, t, W^r, y, K^t, K^s) = \sum_{d \in D} P_d(t|F) \tag{5.1}$$

$$= \sum_{d \in D} P_d(t|w^l_{|W^l|-N+1}, \dots, w^l_{|W^l|}, w^r_1, \dots, w^r_N, y, k^t_1, \dots, k^t_{|K^t|}, k^s_1, \dots, k^s_{|K^s|}),$$

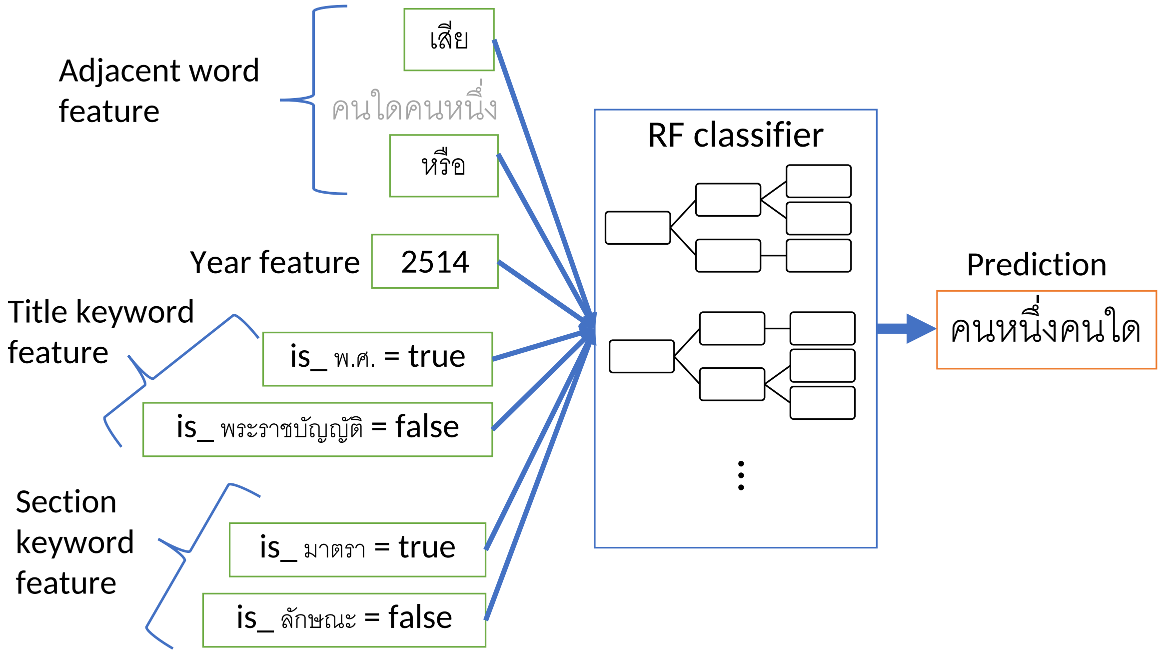

Figure 5.3: Legal term correction model for Thai legal terms

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_i^l and w_i^r are the i-th words of W^l and W^r, respectively. y is the year feature, and k_i^t and k_i^s are the existence of the i-th keyword in the title sentence and section sentence, respectively. N is the window size. Figure 5.3 expresses the input and output of this model.

5.4 Experiment

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

5.4.1 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

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

Table 5.1: Thai legal terms for experiment

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

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 [67]. Table 5.1 shows each legal term and its number of total occurrences.

We compared our method (Random Forest with additional features; RF+) with Random Forest without the additional features (RF) and BERT [18]. 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 [18]. The pretrained model has 12 Transformer

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

Table 5.2: Experimental results of Thai legal term correction

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%**

layers and each layer's unit contains 768 hidden values. We replaced the target legal term in every example into a meta token "ˆ" 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. Note that we do not apply the three techniques introduced in Section 4.3.2 for this BERT classifier.

5.4.2 Experimental Results

Table 5.2 shows the experimental results of each model. RF+ achieved the best accuracy in Set2, Set4-1, and overall accuracy. In every legal term set, RF+ achieved better performance than RF.

5.4.3 Result Analysis

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 5.3 shows the accuracies of each legal term (averaged in results of five-fold cross-validation). According to Table 5.3, 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 accuracies of less frequent legal terms are generally lower than those of the BERT method.

Table 5.3: Accuracy per Thai 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 5.4: Most important features in Thai legal term correction

#	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

Next, we look at the feature importance of Random Forest classifiers. Table 5.4 shows the 10 most important features for each legal term set. In Table 5.4, "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 can be explained by that "*amnat-lae-nathi*" is a newer legal term (refer to Section 5.2.2). Also, "รัฐธรรมนูญ" (*ratthamnun*; constitution) is an important keyword in the legal term set, which can be explained by that constitutions only use "*amnat-nathi*" out of the three legal terms. These facts show that the additional features could capture the outside-the-sentence characteristics of Thai 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.

5.5 Summary

In this chapter, 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.

Chapter 6

Real-World Data Circulation in Legislation

In this chapter, we discuss the relationship between the studies in this thesis and real-world data circulation. In Section 6.1, we describe the general concept of the real-world data circulation. In Section 6.2, we find data circulations in legislation and discuss contributions that our two studies — coordination analysis and legal term correction — bring.

6.1 Real-World Data Circulation

The real-world data circulation is a recent interdisciplinary study field on the utilization of real data. The core objective of the real-world data circulation is a sustainable improvement of the real world by circular data utilization. The circular data utilization is typically categorized into three phases of data processing: data acquisition, data analysis, and data implementation. In the data acquisition phase, we get live data from the real world in a computer-readable form. In the data analysis phase, we analyze the acquired data to figure out knowledge that may affect the real world favorably. In the data implementation phase, we return the knowledge to the real world, which brings about a more comfortable world. Here, it is necessary to sustain this data processing, or it cannot be circulation.

Among many tasks where we can establish a real-world data circulation, we can consider autonomous driving as a typical example. In the data acquisition phase, for example, we identify the terrain around the car from sensors like LiDAR (Light Detection and Ranging) or GPS (Global Positioning System), where the former can directly identify the terrain by point cloud while the latter is to associate with prebuilt terrain data from a database. Using such data, we find the best trajectory for the car that avoids hitting obstacles and that brings about a smooth movement, which is the data analysis phase. Finally, we control the car autonomously according to the computation, which is the data implementation phase. By storing the terrain data from the sensors, we can continue improving our computation model using it. That is, our autonomous driving system is sustainable in terms of continuous improvement.

6.2 Examples of Real-World Data Circulation in Legislation

Data Circulation of Statutes in the Society: Fundamentally, legislation itself can be regarded as the data circulation of statutes in the real world (Figure 6.1). Here two circulations are mutually related: the circulation of societal conditions and the circulation of legislation.

The circulation of societal conditions basically consists of two states: healthy and problematic. A healthy society becomes problematic when it violates the human rights of its citizens. This typically happens due to such environmental changes as technology innovation, climate changes, and economic changes. Legislation copes with the problems by updating the current statutes so that they guarantee the human rights. For example, the 2020 amendment to the Act on the Protection of Personal

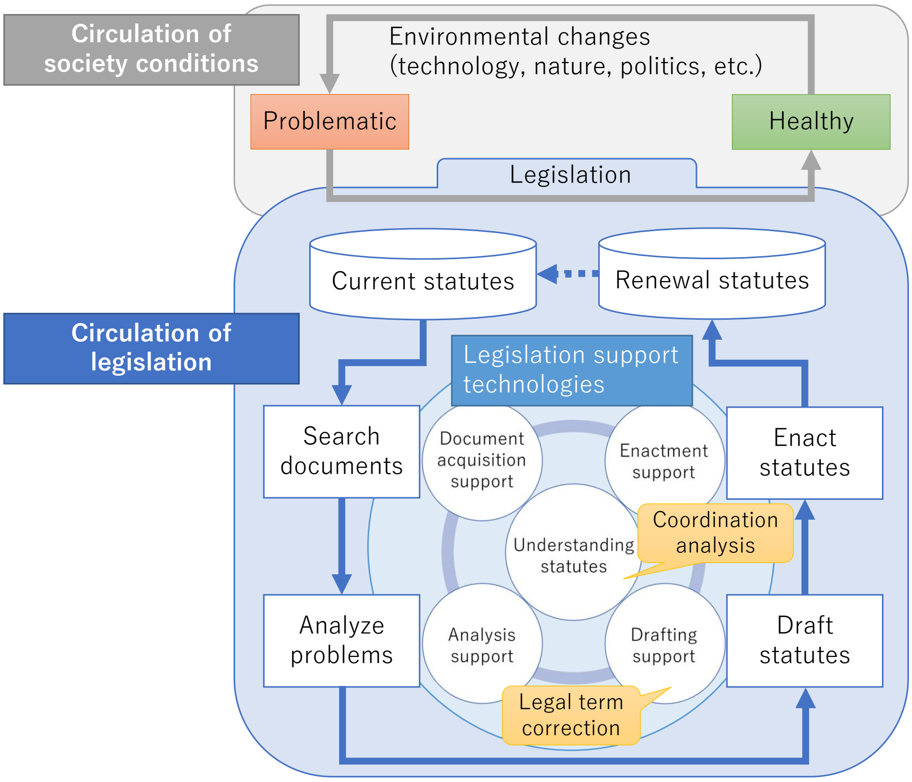

Figure 6.1: Data circulation in legislation

Information improves the scheme of HTTP cookie usage [83]. Advertisement services predict customer segments such as the age range and interests by HTTP cookies of a user gathered from websites he/she accessed, which has been enabled by the innovation of big data analysis. In the current scheme, cookies can be transferred between the advertisement services and websites without the user's prior consent because such data do not directly identify an individual. However, certain websites can augment personal data in their membership system without consent by linking the cookies and the personal data. To solve this vulnerability of privacy protection, the new amendment stipulates that the website shall find the agreement of the users on the cookie data transfer to another entity beforehand.

The circulation of legislation consists of four phases: search, analysis, drafting, and enactment. In the search phase, we thoroughly research related statutes, a phase that resembles the data acquisition phase in the context of real-world data circulation. In the analysis phase, we decide what kind of statute is needed or what statutes should be changed by analyzing the current statutes and current social situations. In the drafting phase, we verbalize the renewed statute in accordance with the legislation rules, which is one perspective of the implementation phase that focuses on documentation. In the enactment phase, we spread the revised statutes to the public, and thus people conduct economic and social activities healthily under the revised statutes. This is another perspective of the implementation phase that focuses on society. Therefore legislation resembles data circulation.

Legislation support technologies make this legislation process rapid and smooth. Each circulation phase has its corresponding technologies that support the phase. For instance, the search phase has document retrieval systems (e.g., [25, 78, 99]), the analysis phase has legal reasoning systems (e.g., [4, 13]), the drafting phase has statute generation systems (e.g., [17, 28]), and the enactment phase has statute databases such as e-Gov Statute Search.

Our two studies, coordination analysis and legal term correction, also contribute to this environment. First, coordination analysis is an important auxiliary process of syntax parsing that supports circulation. The syntax structure of Japanese statutory sentences is useful information for any practical methods for the search, analysis, drafting, and enactment of statutory sentences because it interprets them. Therefore, coordination analysis supports the data circulation of statutes through parsing. The result of coordination analysis itself also helps people to understand statutory sentences and directly accelerates the analysis of the data circulation of statutes. Our coordination analysis method holds a certain position among coordination analysis methods for Japanese statutory sentences because it is compatible with the hierarchical coordinate structures specific in Japanese statutory sentences and provides better predictions by utilizing LSTM-based language models.

Second, legal term correction directly supports the drafting part of the circulation of statutes. It tells legislation officers incorrectly using legal terms, making the drafting process quicker and more efficient than manually inspecting such legal terms. We additionally believe that legal term correction can indirectly contribute to the analysis of the circulation of statutes, which originates from the self-growth of legislation officers. Although a legal term correction system offers correction ideas, the final decision of the legal term use is left to legislation officers who must eventually consider the validity of legal term use by themselves, which is a good resource for legislation training. Once they improve their legislation skills, interpreting statutes will become more efficient. This is the contribution of legal term correction for the analysis part of the circulation of statutes.

Data Circulation of Statutes as Electronic Data: Coordination analysis forms another data circulation centralized by annotated electronic statute data (Figure 6.2). Nowadays, we easily acquire statutes as electronic data from such law database sys-

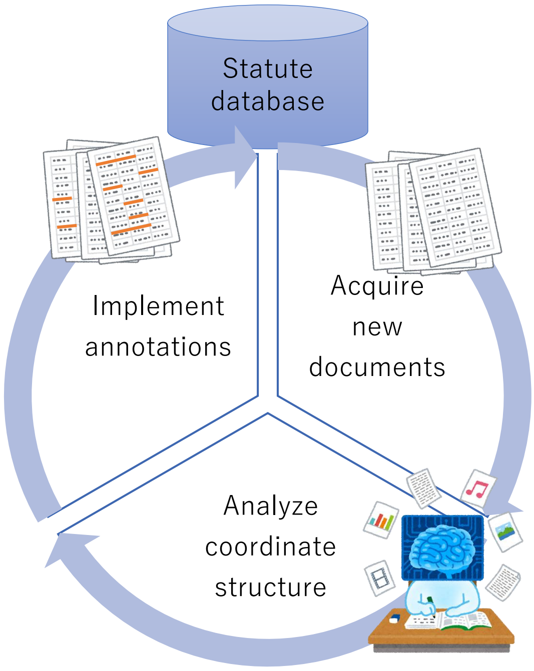

Figure 6.2: Data circulation in statutes as electronic data

tems as e-Gov Statute Search in a computer-friendly form like XML. Utilizing such data sources satisfies the data acquisition role. We then apply coordination analysis to the acquired data to get the coordinate structure information of the statutory sentences, which is the analysis phase. We then append the coordinate structure information to the electronic statute data and upload them to a database. The annotated statute data can be utilized for further tasks, including parsing, visualization, and simplification. That is, they have become resources for legislation support technologies. This is the implementation phase of the data circulation. Since ordinary users do not need raw coordination information in their work, it is reasonable to establish another database system for the annotated statute data in the implementation phase.

Data Circulation of Legal Term Correction: If we utilize the feedback of legal term correction from legislators, we can establish a data circulation centralized by correction history data. Figure 6.3 shows such a circulation. A legal term correction system receives a statute draft and outputs corrected ideas using the correction model, for example, the classifiers in our proposed method. In this circulation, we ask skilled legislators to judge the correction results of the model. We put their feedback into the correction history of the system. The correction history then contributes two ways of data circulation. The first way is regarding the system, which updates the

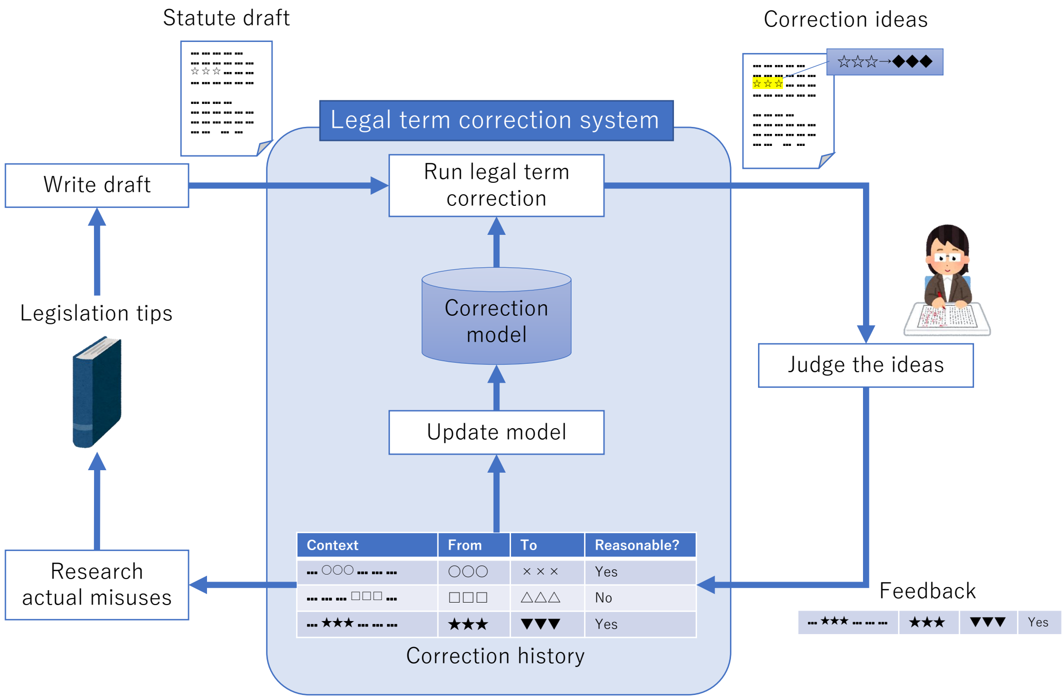

Figure 6.3: Data circulation in legal term correction

correction model from the history. The second is regarding legislation practices, where we analyze the history from the perspective of how legislators made mistakes. This analysis result is compiled as legislation tips, which teach legislators better ways to use legal terms. Both ways produce a better legislation environment so that the system can provide an improved prediction model and reduce mistakes in legislation.

Data Circulation of Technology: Generally, establishment of technologies (including methodology, training data, trained models, etc.) proceeds to further establishment of technologies, which can be regarded as a data circulation (Figure 6.4). We introduce an established technology, research a novel technology based on the introduced one, and then propose the technology. Each action can be regarded as the acquisition phase, the analysis phase, and the implementation phase of technology.

Legislation support technologies in this thesis may also proceed to further establishment of legislation support technologies. One possible scenario is the application of trained models to other tasks. For example, we constructed neural language models in the coordination analysis study. We also domain-adapted a BERT language representation model in the legal term correction study. These models can be utilized for tasks that require semantic comparison of statutory sentences such as document retrieval [25, 78, 99] by encoding a statutory sentence according to word relationships.

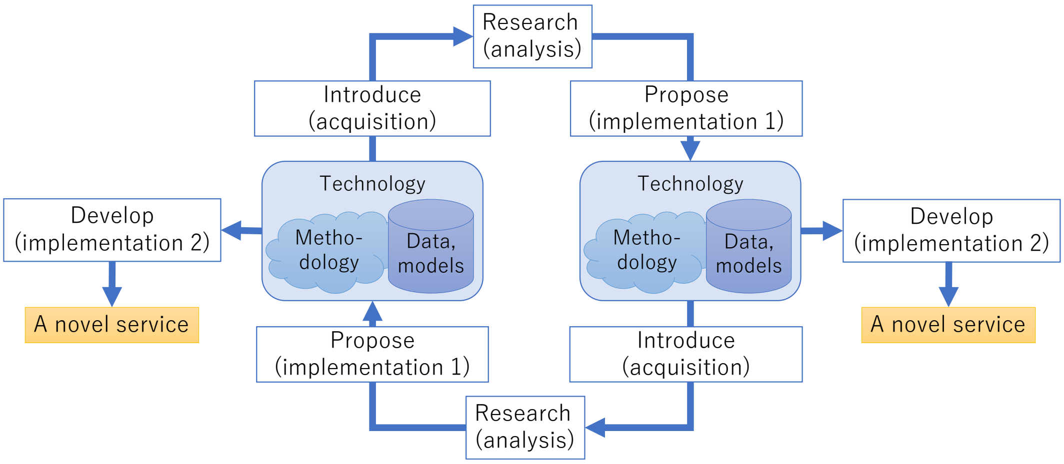

Figure 6.4: Data circulation of technology

The methodology for a task also triggers the development of another technology. For example, the methodology of legal term correction can be used for term unification in statutory sentence translation because both legal term correction and term unification have situations in common: There are a number of legal term sets and their firm usages.

The updated technologies trigger the development of novel systems that improve the legislation process, which is the social implementation of this data circulation.

LegalAI Project and its Data Circulation: For a practical activity regarding real-world data circulation, I am participating in the LegalAI project where I have been developing an online contract review system. Figure 6.5 shows an overview of the LegalAI system, which has three modules: anonymization, comment generation, and database.

The legal term correction methodology proposed in this study is implemented in this system as a comment generation module. Here the legal term correction model is trained by contracts. In addition to legal term correction, the system offers several other comment generation modules based on natural language processing technologies, such as risky clause detection, contract density judgment, and misspelling detection.

With this system, we aim to establish an organic data circulation centralized by feedback from users by automatically updating the judgment rules, the legal term set, and the classifier based on the feedback. With this plan, the system can achieve data circulation, where it acquires feedback, analyzes it to establish more sophisticated knowledge, and implements the knowledge as system updates. When we establish a data circulation that uses texts in contracts, the anonymization module is crucial. Un-

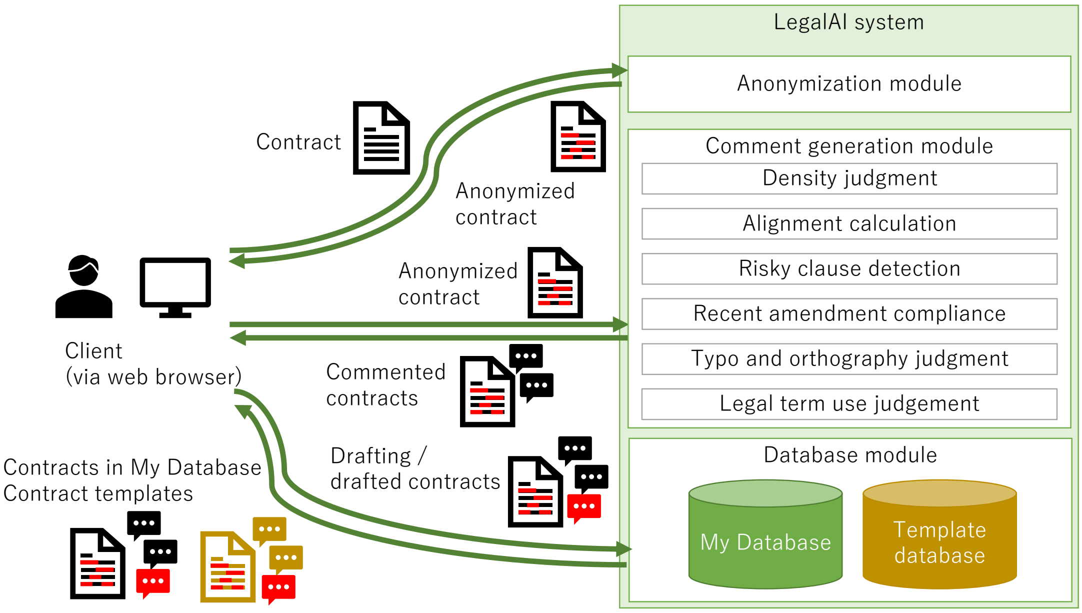

Figure 6.5: LegalAI system

like statutes, contracts often contain sensitive expressions, such as names, addresses, account numbers, transaction amounts, and so forth. Therefore, we must replace such sensitive expressions with anonymized expressions when we reuse the text data even if we are using them for training machine learning models. Even though LegalAI currently offers an anonymization module as a web service, we do not consider it to be optimal. To maintain maximum security, we want to separate the module as an offline service that users can execute without requiring internet access.

As another utilization of real-world data regarding contracts, we aim to offer statistical facts of uploaded contracts such as frequently commented clauses, frequently appearing clauses in certain contract categories, comparisons of clause amounts between users and averages, and so on. This plan will encourage users to reflect and improve their practices when they draft contracts. Here lies another real-world data circulation for users' drafting skills. We collect contracts and their clauses in the acquisition part and analyze statistical facts from the data in the analysis part. We then highlight the habits of users based on the statistics in the implementation part. They will eventually learn better ways to draft contracts, resulting in improved contracts.

Chapter 7

Conclusion

In this chapter, we summarize our entire work. In Section 7.1, we summarize our findings and the contributions of this thesis. In Section 7.2, we discuss future work for our two studies.

7.1 Overall Summary

In this thesis, we studied two topics regarding statutory sentences: coordination analysis and legal term correction.

In Section 1, we overviewed the characteristics of statutory sentences and identified two issues for handling Japanese statutory sentences. The first issue is the strong influence of legislation rules on drafting. With these rules, the legislation bureau scrutinizes the phrasing of Japanese statutory sentences. This motivates our legal term correction study. The second is the appearance of difficult statutory sentences that have complex coordinate structures, which led us to study coordination analysis. Among the existing studies, we identified academic and practical value of the two studies.

In Section 2, we explained the knowledge and techniques that are the keys for our two studies. We first reviewed Japanese legislation rules. We described the legal terms and coordination in Japanese statutory sentences and their characteristics. We then reviewed the language models and classifiers and their characteristics to deepen the discussions in subsequent chapters.

In Section 3, we tackled the coordination analysis task of Japanese statutory sentences and proposed a new coordination analysis method based on neural language models. Such models judge conjunct scope candidates from two hypothetical properties of coordination: similarity and interchangeability of paired conjuncts. For the identification of coordination hierarchy, the method uses the legislation rules of coordinators for indicating hierarchy. Among neural-based coordination analysis methods, we identified one strong point in our method: it does not require training data with syntax or even coordination information. This feature is friendly to Japanese statutory sentences because it is prohibitively expensive to compile a large-scale corpus of statutory sentences with syntax information because of the complexity. From experiments, we showed that this method outperformed the existing methods for Japanese statutory sentences based on word matching.

In Section 4, we tackled legal term correction tasks. First, we established the definition and an algorithm for legal term correction tasks and showed that legal term correction can be solved in the framework of sentence completion tests. Next we introduced two approaches for such tasks. The first approach utilizes Random Forest classifiers, each of which is specialized for each responsible legal term set. This approach outperformed neural language models that are typically used for sentence completion tests. From the result of the classifiers' optimization process, we also obtained such findings as the tendency of context selection by legal term sets. However, we encountered a two-layered class imbalance problem, which originated from relatively infrequent legal terms in legal term sets and absolutely infrequent legal term sets. To solve this problem and improve our predictions, we took the second approach that utilizes a BERT classifier. We applied three techniques, domain adaptation, repetitive soft undersampling, and classifier unification, to solve the two-layered class imbalance problem. We experimentally showed that the BERT classifier achieved the best performance among them as well as the validity of the three techniques by ablation studies.

In Section 5, we applied our legal term correction methodology to Thai statutory sentences to judge how well the methodology works globally. First, we identified the characteristics of Thai legal terms and their differences from Japanese legal terms. We identified three characteristics in Thai legal terms that may impede directly applying the methodology: year dependency, genre dependency, and insufficient context. Next we proposed a legal term correction method for Thai statutory sentences and introduced three additional features: year, title, and section features that respectively handle the three characteristics. We experimentally showed the method's effectiveness and validity.

In Section 6, we discussed real-world data circulation around our studies and found that legislation itself forms a data circulation centered on statutes. Our proposed methods contribute to this circulation by accelerating the legislation process. Plus, we discussed that both a coordination analysis system and a legal term correction system will also build a real-world data circulation in their systems.

7.2 Future Work

In this section we discuss the remaining tasks in both the coordination analysis study and the legal term correction study.

7.2.1 Future Work on Coordination Analysis

In the coordination analysis study, the nearest remaining task is to improve the identification mechanism. For hierarchy identification, we saw examples whose coordination hierarchy cannot be correctly identified because of the deterministic identification rules. One solution is to utilize a CFG parser to list every possible hierarchy pattern, as Yamakoshi et al. [95] did with a word-based scoring method. However, the computation cost is this solution's bottleneck, since using LSTM-based neural language models and applying a CFG parser to long sentences are both heavy processes. A possible solution is to replace our LSTM-based language models with lightweight models such as Skipgram [55]. Knowledge distillation [32] can be utilized to train small language models effectively.

For sophisticated conjunct identification, we can find methods to score conjunct scope candidates better than LSTM-based neural language models. One choice is a general-purpose language representation model such as BERT, which we use as a language model. The following are the advantages of using it instead of the LSTM-based language models. We can utilize a pretrained model that has plenty of linguistic knowledge. We can apply domain adaptation with Japanese statutory sentences. The self-attention mechanism can directly capture word relationships between distant words, although LSTM-based language models fail to directly capture such relationships unless we introduce an attention mechanism [3].

Even though integrating coordination analysis with sentence parsing [30, 40, 41, 48] may be the ultimate formation of coordination analysis, we argue that it is more suitable for Japanese statutory sentences to conduct coordination analysis prior to sentence parsing. This is because Japanese statutes often contain prohibitively long sentences, and in such a situation we will not be able to apply parsing. A solution for this situation is to disassemble the sentences into fragments and individually apply parsing, which needs coordination analysis.

From this point of view, we need to reduce the computational cost of coordination analysis for practical use. Our method uses forward and backward LSTM-based language models that require heavy computation. Furthermore, the method feeds to the language models four sentences (W_{fl}, W_{fr}, W_{bl}, and W_{br}) to calculate the similarity score and one sentence W_s to calculate the fluency score. As same as the incorporation with a CFG parser, one way to reduce the computational cost is applying such small language models as Skipgram [55] instead of LSTM-based language models.

After establishing a fast coordination analysis method for Japanese statutory sentences, the method may accelerate the development of legislation support technologies. First, automatic syntax annotation enables us to utilize annotated statutory sentences for data-driven learning methods. For example, we can apply a neural machine translation method that incorporates syntax information [19]. As a more practical usage, we can achieve a completely automatic visualization system for Japanese statutory sentences [94], where the authors must manually revise the parsed data output by a general purpose parser. Another practical usage is the compilation of a legal term thesaurus [29]. Since a coordinate structure refers similar entities, we can regard conjuncts in a coordinate structure as synonyms. Furthermore, we may acquire hypernyms and hyponyms from hierarchical coordinate structures and coordinate structures with the coordinator "*sonota* (other$_1$)." Such hypernym-hyponym relationships reflect the word usage specific in statutory sentences, which will be a novel resource for study.

7.2.2 Future Work on Legal Term Correction

In the legal term correction study, the biggest issue is to be able to cope with any kind of proofreading objectives (misspellings, word usage, phrase usage, sentence structure, etc.) of the statutory sentences. One approach is to build a generative model like Hitomi et al.'s one [33] that reforms a pre-proofread sentence into a proofread sentence. There are two problems to adopt this approach for statutory sentence proofreading. First, it requires a proofreading dataset that consists of a huge number of pre- and post-proofread sentence pairs, which is complicated to acquire. Second, it is not straightforward to identify and consider the intent of each revision done by the generative model. Since statutory sentences are highly technical sentences bound by strict writing rules, we need to connect each revision to its ground.

One intermediate improvement of the legal term correction methodology is to correct "non-legal terms" that should be expressed as legal terms. For example, we revise "...したら" (*shi tara*) to "...したとき" (*shi ta toki*), where *tara* (when) is colloquial and replace it with a legal term, "とき" (*toki*). Here multiple expressions should be written as "とき," such as "なら" (*nara*) and "ならば" (*naraba*). One possible solution for this task is to introduce a new classifier that identifies which word or words in the input statutory sentence should be revised and to what.

Another approach to improve the current method is to enlarge the evaluation target. We need to establish a method that automatically detects sets of confusing phrases from a statutory sentence corpus. This method can be used not only for legal term correction but also for jurisprudential studies on the views of incorrect word usage. Although similarity of context can be a strong clue to find such mistaken phrases, some incorrect properties seem to come from outside of the context similarity. For example, in English, homophones are two words with similar pronunciations (principle and principal) whose meanings are quite different. Similarly, in Japanese two words with similar appearances (既出 and 概出) can also be confusing. A proofreading corpus can be a resource to identify such confusing expressions for many different linguistic sensibilities.

Our classification methodology of legal term correction can be utilized for other tasks. One potential case is term unification in statutory sentence translation. The Japanese government maintains a Standard Legal Terms Dictionary [86], which defines a number of legal term sets that are appropriate usages for translators. For example, legal term "*kisoku*" (規則) has two translations: "regulation" and "rule."

The former denotes a statute from government ministries, and the latter means internal practical guidelines for boards and courts. Since the dictionary defines legal term sets by their usages, we can solve this task with our classification methodology.

Word correction can be found not only in writing statutory sentences but also in writing daily sentences. Many domains, such as reports and news articles, require very strict word use. Even when writing daily sentences, since people may mistake word use from ignorance or misspellings, automatic word usage correction will also be beneficial in daily sentences. As we discussed in Section 1.2.2, many proofreading methods have been proposed. However, their word usage correction can be improved. According to the rankings from the BEA-2019 shared task [8], the best prediction model, which is Transformer-based, achieved a 43.48 $F_{0.5}$ score in adjective correction, 49.41 in adverb correction, 48.67 in conjunct correction, and 41.17 in noun correction. These scores are quite worse than ones in easily correctable error categories such as determiner correction (75.67 $F_{0.5}$ score), noun inflection correction (91.95), and so on. There will be room for our classifier methodology in improving such challenging error categories.

Building a language representation model is one beneficial work for legal term correction and overall statutory sentence processing. Many tasks in statutory sentence processing discussed can be regarded as sentence-level classification (like important sentence extraction [73]) or word-level classification (like semantic role labeling [26, 27, 76, 65, 88, 92]), which a language representation model can handle directly or with a small modification. In the pretraining process, room exists for considering the pretraining task design specialized for statutory sentences. The BERT pretraining process adopts a masked language model task and a next sentence prediction task; other pretraining schemes have also been proposed. For example, Lewis et al. [50] designed five pretraining tasks, token masking, token deletion, text infilling, sentence permutation, and document rotation, for their sequence-to-sequence based language representation model called BART. One concern in building a pure domain-specified language representation model is the amount of text resources. As described in Section 4.4, although we built a Japanese statutory sentence corpus of approximately 44 million words, Devlin et al. pretrained a BERT model [18] with a corpus of 3.3 billion words, which is 75 times larger than ours. One solution is to use statutory sentences in a broad sense, for example, ordinances and contracts, where we need to heed their qualities and their unique characteristics.

Appendix A

Legal Terms

Table A.1 to Table A.4 show (1) nominal legal term sets, (2) adjective and adverbial legal term sets, (3) verbal legal term sets, and (4) conjunctional legal term sets that we defined, respectively. The frequencies of legal terms indicated in these tables are values after word concatenation. English translations in these tables are taken from the Standard Legal Terms Dictionary (March 2018 edition) [86] provided by the Ministry of Justice, Japan, except for asterisked items. Subfixes of certain legal terms correspond to ones in the main chapters.

Table A.1: List of nominal legal term sets

Legal term (pronunciation; meaning)	Total	In training	In test
規定 $_{(f)}$ (*kitei*; provision)	401,381	387,756	13,625
規程 $_{(g)}$ (*kitei*; rules, procedure, regulation)	4,139	3,975	164
とき (*toki*; if, when, whenever (condition))	127,861	123,082	4,779
場合 (*baai*; if, when, whenever (condition))	188,970	182,116	6,854
時 (*toki*; when (meaning a certain time))	17,808	17,290	518
者 $_{(a)}$ (*mono*; natural or juristic person*)	353,279	341,839	11,440
物 $_{(b)}$ (*mono*; tangible object*)	29,689	28,546	1,143
もの $_{(c)}$ (*mono*; abstract object*)	231,715	223,326	8,389
許可 (*kyoka*; permission, license)	24,145	23,394	751
認可 (*ninka*; authorization, approval, permission, confirmation)	15,677	15,182	495
届出 (*todokede*; notification, report, registration)	22,021	21,366	655
認証 (*ninsho*; certification, accreditation)	1,949	1,818	131
通知 (*tsuchi*; notice)	17,894	17,304	590
通報 (*tsuho*; notification, report, information)	768	741	27
報告 (*hokoku*; report)	16,675	16,071	604
連絡 (*renraku*; contact, liaison*)	1,475	1,434	41
命令 (*meirei*; order, direction)	15,176	14,715	461
指揮 (*shiki*; direction, command)	570	552	18
指示 (*shiji*; instruction*)	3,034	2,882	152
監督 (*kantoku*; supervision)	3,732	3,638	94
要求 $_{(m)}$ (*yokyu*; requirement*)	1,606	1,558	48
要請 $_{(n)}$ (*yosei*; request*)	1,671	1,602	69
施行 (*seko*; enforcement*)	233,058	224,535	8,523
適用 (*tekiyo*; application)	83,062	80,723	2,339
準用 (*jun'yo*; application mutatis mutandis*)	2,041	1,909	132

Table A.2: List of adjective and adverbial legal term sets

Legal term (pronunciation; meaning)	Total	In training	In test
当該 (*togai*; that, the, referenced, relevant)	297,904	288,300	9,604
その (*sono* ; that*)	213,114	205,694	7,420
に 係る (*ni kakaru*; pertaining to*)	161,564	156,226	5,338
に 関する (*ni kansuru*; regarding*)	120,076	115,912	4,164
に 関係する $_{(o)}$ (*ni kankeisuru*; regarding*)	80	79	1
の (*no*; of*)	2,813,563	2,718,160	95,403
に 規定する (*ni kiteisuru*; provided for in, prescribed in*)	169,742	163,721	6,021
の 規定 に よる (*no kitei ni yoru*; pursuant to, under the provisions of*)	113,123	109,404	3,719
直ちに $_{(h)}$ (*tadachini*; immediately)	2,414	2,332	82
速やかに $_{(i)}$ (*sumiyakani*; promptly)	2,229	2,119	110
遅滞なく $_{(j)}$ (*chitainaku*; without delay)	6,549	6,328	221
に基づき (*ni motozuki*; based on*)	8,883	8,589	294
により (*niyori*; by*)	205,212	197,244	7,968

Table A.3: List of verbal legal term sets

Legal term (pronunciation; meaning)	Total	In training	In test
処する (*syosuru*; punish*)	4,841	4,605	236
科する (*kasuru*; impose (fine or punishment)*)	1,212	1,166	46
課する (*kasuru*; impose (tax)*)	4,320	4,269	51
適用する (*tekiyosuru*; apply)	21,119	20,378	741
準用する (*jun'yosuru*; apply mutatis mutandis)	66,303	64,231	2,072
例による (*reiniyoru*; is governed by)	4,368	4,228	140
推定する (*suiteisuru*; presume)	228	221	7
みなす (*minasu*; deem)	20,039	19,353	686
とする (*to suru*; shall be)	131,314	126,186	5,128
である (*de aru*; be*)	58,020	56,110	1,910
する こと が できない (*suru koto ga dekinai*; may not, be unable to)	6,783	6.588	195
して は ならない (*shite ha naranai*; must not, is prohibited)	4,457	4,301	156
する こと が できる (*suru koto ga dekiru*; may)	29,348	28,301	1,047
しなければ ならない (*shinakereba naranai*; must, shall)	42,679	41,023	1,656
する もの と する (*suru mono to suru*; is to)	11,501	11,043	458
この 限り で ない (*kono kagiri de nai*; does not apply to*)	7,380	7,117	263
妨げない (*samatagenai*; does not preclude)	1,419	1,364	55
なお 従前 の 例による (*nao juzen no reiniyoru*; prior laws continue to govern)	36,402	35,345	1,057
なお 効力 を 有する (*nao koryoku wo yusuru*; remain in force*)	2,734	2,702	32
改める (*aratameru*; revise (an expression))	6,941	6,728	213
改正する (*kaisei suru*; revise (a statute))	24,349	23,519	830

Table A.4: List of conjunctional legal term sets

Legal term (pronunciation; meaning)	Total	In training	In test
又は (*matawa*; or$_H$)	337,058	325,711	11,347
若しくは (*moshikuwa*; or$_L$)	88,241	85,393	2,848
及び $_{\text{(d)}}$ (*oyobi*; and$_L$)	301,460	290,921	10,539
並びに $_{\text{(e)}}$ (*narabini*; and$_H$)	49,584	47,944	1,640
その他 (*sonota*; other$_1$)	29,163	27,959	1,204
その他の (*sonotano*; other$_2$)	55,391	53,525	1,866
前項 の 場合 に おいて $_{\text{(k)}}$ (*zenko no baai ni oite*; in the case referred to in the preceding paragraph)	2,834	2,714	120
前項 に 規定する 場合 に おいて $_{\text{(l)}}$ (*zenko ni kiteisuru baai ni oite*; in the case prescribed in the preceding paragraph)	325	316	9
ただし (*tadashi*; provided, however, that ⋯)	39,234	37,737	1,497
この 場合 に おいて (*kono baai ni oite*; in this case)	20,788	20,139	649

Publications

Peer-reviewed Journal Papers	Related Chapter
Takahiro Yamakoshi, Takahiro Komamizu, Yasuhiro Ogawa, and Katsuhiko Toyama. Japanese Mistakable Legal Term Correction using Infrequency-aware BERT Classifier. Transactions of the Japanese Society for Artificial Intelligence : AI, Vol.35, No.4, pp.E-K25_1-17, 2020.	Chapter 4
Takahiro Yamakoshi, Yasuhiro Ogawa, Takahiro Komamizu, and Katsuhiko Toyama. Japanese Legal Term Correction using Random Forest. Transactions of the Japanese Society for Artificial Intelligence : AI, Vol.35, No.1, pp.H-J53_1-14, 2020 (In Japanese).	Chapter 4
Takahiro Yamakoshi, Tomohiro Ohno, Yasuhiro Ogawa, Makoto Nakamura, and Katsuhiko Toyama. Hierarchical Coordinate Structure Analysis for Japanese Statutory Sentences Using Neural Language Models. Journal of Natural Language Processing，Vol.25, No.4, pp.393-420, 2018. (C) The Association for Natural Language Processing, (Licensed under CC BY 4.0）https://creativecommons.org/licenses/by/4.0/	Chapter 3

Peer-reviewed International Conference Papers

Takahiro Yamakoshi, Takahiro Komamizu, Yasuhiro Ogawa, and Katsuhiko Toyama. Japanese Mistakable Legal Term Correction using Infrequency-aware BERT Classifier. Proceedings of the 2019 IEEE International Conference on Big Data, pp.4342–4351, 2019.	Chapter 4

Takahiro Yamakoshi, Vee Satayamas, Hutchatai Chanlekha, Yasuhiro Ogawa, Takahiro Komamizu, Asanee Kawtrakul, and Katsuhiko Toyama. Thai Legal Term Correction using Random Forest with Outside-the-sentence Features. Proceedings of the 33rd Pacific Asia Conference on Language, Information and Computation, pp.279-287, 2019. Chapter 5

Takahiro Yamakoshi, Takahiro Komamizu, Yasuhiro Ogawa, and Katsuhiko Toyama. Japanese Legal Term Correction Using Random Forests. Legal Knowledge and Information Systems - JURIX 2018: The Thirty-First Annual Conference on Legal Knowledge and Information Systems, IOS Press, pp.161-170, 2018. Chapter 4

Takahiro Yamakoshi, Tomohiro Ohno, Yasuhiro Ogawa, Makoto Nakamura, and Katsuhiko Toyama. Coordination Analysis for Japanese Statutory Sentences Using Neural Language Models. Proceedings of the Eleventh International Workshop on Juris-informatics, pp.128-141, 2017. Chapter 3

Acknowledgement

First, I would like to show my most sincere appreciation to Professor Toyama Katsuhiko, Associate Professor Ogawa Yasuhiro, and Assistant Professor Komamizu Takahiro in Toyama Laboratory who have raised me with many aspects such as research supervisory, career consultation, and sometimes life consultation. I would like to same appreciation to Associate Professor Ohno Tomohiro who is currently at the Graduate School of Advanced Science and Technology, Tokyo Denki University and Associate Professor Nakamura Makoto who is currently at the Department of Engineering, Niigata Institute of Technology. They also kindly gave me insightful views on the coordination analysis study and future research plans in the doctoral program when they were in Toyama laboratory.

Next, I would like to express my appreciation to Professor Takeda Koichi, Professor Matsubara Shigeki, and Professor Sasano Ryohei, who gave me insightful comments in wrapping up this thesis.

Regarding the study of Japanese legal term correction, I would like to express my appreciation to Yamada Hisatake, Esq. and Mr. Matsushita Ken, who gave me an opportunity to participate in the LegalAI project. Without this opportunity, I will not have realized the achievements of legal term correction in this study. We would like to express another appreciation to Yamada Hisatake, Esq. and the staff in his Shobu law office, who supported my study and development of Legal AI system. Also, they enlightened me about my future life plans.

Besides, I would like to express my appreciation to Professor Asanee Kawtrakul, Associate Professor Hutchatai Chanlekha, and Mr. Vee Satayamas, who are at Department of Computer Engineering, Kasetsart University, Thailand, for allowing me to study Thai natural language processing in Asanee Laboratory. Without the supports and discussions in her laboratory, I could never develop the Thai legal term correction method. They also kindly supported my life in Bangkok.

Finally, I would like to appreciate my family, laboratory members, and friends for supporting my daily life and my student life.

Bibliography

[1] Tazin Afrin and Diane Litman. Annotation and classification of sentence-level revision improvement. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 240–246, 2018.

[2] Sarthak Anand, Debanjan Mahata, Haimin Zhang, Simra Shahid, Laiba Mehnaz, Yaman Kumar, and Rajiv Ratn Shah. MIDAS@SMM4H-2019: Identifying adverse drug reactions and personal health experience mentions from Twitter. In *Proceedings of the 4th Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*, pp. 127–132, 2019.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of International Conference on Learning Representations*, 15 pages, 2015.

[4] Sotiris Batsakis, George Baryannis, Guido Governatori, Ilias Tachmazidis, and Grigoris Antoniou. Legal representation and reasoning in practice: A critical comparison. In *Legal Knowledge and Information Systems - JURIX 2018: The Thirty-First Annual Conference on Legal Knowledge and Information Systems*, pp. 31–40. IOS Press, 2018.

[5] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, Vol. 3, pp. 1137–1155, 2003.

[6] Leo Breiman. Bagging predictors. *Machine Learning*, Vol. 24, pp. 123–140, 1996.

[7] Leo Breiman. Random forests. *Machine Learning*, Vol. 45, pp. 5–32, 2001.

[8] Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 52–75, 2019.

[9] Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 793–805, 2017.

[10] Young-Soog Chae. Improvement of Korean proofreading system using corpus and collocation rules. In *Proceedings of the 12th Pacific Asia conference on Language, Information and Computation*, pp. 328–333, 1998.

[11] Yuchang Cheng and Tomoki Nagase. Analysis of proofreading history intended for Japanese engineering document that foreign language speaker written. In *Proceedings of the 18rd Annual Meeting of the Association for Natural Language Processing*, pp. 34–37, 2012. In Japanese.

[12] Yuchang Cheng and Tomoki Nagase. An example-based Japanese proofreading system for offshore development. In *Proceedings of the 24th International Conference on Computational Linguistics*, pp. 67–76, 2012.

[13] Rishi Chhatwal, Peter Gronvall, Nathaniel Huber-Fliflet, Robert Keeling, Jianping Zhang, and Haozhen Zhao. Explainable text classification in legal document review a case study of explainable predictive coding. In *2018 IEEE International Conference on Big Data (Big Data)*, pp. 1905–1911, 2018.

[14] Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 53–62, 2016.

[15] Rachele De Felice and Stephen G. Pulman. A classifier-based approach to preposition and determiner error correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 169–176, 2008.

[16] Koenraad de Smedt and Victoria Rosén. Automatic proofreading for Norwegian: The challenges of lexical and grammatical variation. In *Proceedings of the 12th Nordic Conference of Computational Linguistics*, pp. 206–215, 2000.

[17] Stijn Debaene, Raf van Kuyck, and Bea Van Buggenhout. Legislative technique as basis of a legislative drafting system. *Legal Knowledge Based System: JURIX 1999: The Twelfth Conference*, pp. 23–34, 1999.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

[19] Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 72–78, 2017.

[20] Lester Faigley and Stephen Witte. Analyzing revision. *College Composition and Communication*, Vol. 32, No. 4, pp. 400–414, 1981.

[21] Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. Morphological inflection generation using character sequence to sequence learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 634–643, 2016.

[22] Jessica Ficler and Yoav Goldberg. Coordination annotation extension in the Penn Tree Bank. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 834–842, 2016.

[23] Jessica Ficler and Yoav Goldberg. A neural network for coordination boundary prediction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 23–32, 2016.

[24] Yoav Freund and Robert E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, Vol. 14, No. 5, pp. 771–780, 1999.

[25] Kazuya Fujioka, Takahiro Komamizu, Yasuhiro Ogawa, and Katsuhiko Toyama. Bubunkozo wo mochiita ruijireiki no kensaku. In *Proceedings of DEIM 2019*, 4 pages, 2019. In Japanese.

[26] Ruka Funaki, Yusuke Nagata, Kohei Suenaga, and Shinsuke Mori. A contract corpus for recognizing rights and obligations. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 2045–2053, 2020.

[27] Ingo Glaser, Elena Scepankova, and Florian Matthes. Classifying semantic types of legal sentences: Portability of machine learning models. In *Legal Knowledge and Information Systems - JURIX 2018: The Thirty-First Annual Conference on Legal Knowledge and Information Systems*, pp. 61–70. IOS Press, 2018.

[28] Carole D. Hafner and Marc Lauritsen. Extending the power of automated legal drafting technology. *Legal Knowledge and Information Systems - JURIX 2007: The Twentieth Annual Conference on Legal Knowledge and Information Systems*, pp. 59–68, 2007.

[29] Masato Hagiwara, Yasuhiro Ogawa, and Katsuhiko Toyama. Bootstrapping-based extraction of dictionary terms from unsegmented legal text. In *New Frontiers in Artificial Intelligence: JSAI 2008 Conference and Workshops, Revised Selected Papers, Lecture Notes in Computer Science*, Vol. 5447, pp. 213–227. Springer, 2009.

[30] Atsushi Hanamoto, Takuya Matsuzaki, and Jun'ichi Tsujii. Coordination structure analysis using dual decomposition. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 430–438, 2012.

[31] Kazuo Hara, Masashi Shimbo, Hidemaru Okuma, and Yuji Matsumoto. Coordinate structure analysis with global structural constraints and alignment-based local features. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*, pp. 967–975, 2009.

[32] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[33] Yuta Hitomi, Hideaki Tamori, Naoaki Okazaki, and Kentaro Inui. Proofread sentence generation as multi-task learning with editing operation prediction. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*, pp. 436–441, 2017.

[34] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

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

[36] Taishi Ikeda, Hiroyuki Shindo, and Yuji Matsumoto. Japanese text normalization with encoder-decoder model. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pp. 129–137, 2016.

[37] Masazumi Ishige. *Jichirippojitsumu no tame no hoseishitsumu shokai (2nd new edition)*. Gyosei, 2012. In Japanese.

[38] Emi Izumi, Kiyotaka Uchimoto, and Hitoshi Isahara. Error annotation for corpus of Japanese learner English. In *Proceedings of the Sixth International Workshop on Linguistically Interpreted Corpora*, pp. 71–80, 2005.

[39] Slava 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.

[40] Daisuke Kawahara and Sadao Kurohashi. A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 176–183, 2006.

[41] Daisuke Kawahara and Sadao Kurohashi. Coordination disambiguation without any similarities. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 425–432, 2008.

[42] Daisuke Kawahara, Sadao Kurohashi, and Makoto Nagao. Construction of a Japanese relevance-tagged corpus. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pp. 2008–2013, 2002.

[43] Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. GENIA corpus — semantically annotated corpus for bio-textmining. *Bioinformatics*, Vol. 19, No. suppl_1, pp. i180–i182, 2003.

[44] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, 2014.

[45] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*, 15 pages, 2015.

[46] Taku Kudo. MeCab : Yet another part-of-speech and morphological analyzer, 2005.

[47] Sadao Kurohashi, Yuiko Igura, and Shoko Sakaguchi. *Keitaiso · kobun tagu tsuki kopasu sakusei no sagyokijun*. Kyoto University, 2000. In Japanese.

[48] Sadao Kurohashi and Makoto Nagao. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, Vol. 20, No. 4, pp. 507–534, 1994.

[49] Sadao Kurohashi and Makoto Nagao. Building a Japanese parsed corpus while improving the parsing system. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, pp. 719–724, 1998.

[50] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, 2020.

[51] Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3291–3301, 2019.

[52] Zhuoran Liu and Yang Liu. Exploiting unlabeled data for neural grammatical error detection. *Journal of Computer Science and Technology*, Vol. 32, No. 4, pp. 758–767, 2017.

[53] Masamichi Maeda. *Workbook Houseishitsumu (completely revised)*. Gyosei, 1983. In Japanese.

[54] Hiroki Matsuyama, Kiyoaki Shirai, and Akira Shimazu. Horei bunsho wo taisho ni shita heiretsukozo kaiseki. In *Proceedings of the 18th Annual Meeting of the Association for Natural Language Processing*, pp. 975–978, 2012. In Japanese.

[55] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of International Conference on Learning Representations*, 12 pages, 2013.

[56] Yusuke Miyao and Jun'ichi Tsujii. Deep linguistic analysis for the accurate identification of predicate-argument relations. In *Proceedings of the 20th International Conference on Computational Linguistics*, pp. 1392–1398, 2004.

[57] Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*, Vol. 26, pp. 2265–2273, 2013.

[58] Koki Mori, Makoto Miwa, and Yutaka Sasaki. 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. In Japanese.

[59] Raphael Mudge. The design of a proofreading software service. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids*, pp. 24–32, 2010.

[60] Ryo Nagata, Edward Whittaker, and Vera Sheinman. Creating a manually error-tagged and shallow-parsed learner corpus. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 1210–1219, 2011.

[61] Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 229–234, 2017.

[62] Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pp. 1–14, 2014.

[63] Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the*

Seventeenth Conference on Computational Natural Language Learning: Shared Task, pp. 1–12, 2013.

[64] Ngan Nguyen and Yusuke Miyao. Alignment-based annotation of proofreading texts toward professional writing assistance. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 753–759, 2013.

[65] Truong-Son Nguyen, Le-Minh Nguyen, Akira Shimazu, and Kiyoaki Shirai. Structural paraphrasing in Japanese legal texts. In *Proceedings of the Eleventh International Workshop on Juris-informatics*, pp. 62–75, 2017.

[66] Diane Nicholls. The Cambridge Learner Corpus - error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics 2003 conference*, pp. 572–581, 2003.

[67] Office of the Council of State. *Legislative Drafting Manual*. 2008. In Japanese.

[68] Yasuhiro Ogawa, Masayuki Yamada, Ryuta Kato, and Katsuhiko Toyama. Design and compilation of syntactically tagged corpus of Japanese statutory sentences. In *New Frontiers in Artificial Intelligence: JSAI 2010 Conference and Workshops, Revised Selected Papers, Lecture Notes in Computer Science*, Vol. 6797, pp. 141–152. Springer, 2011.

[69] Toshihiko Oshima. *Hoseishitsumu no kiso chishiki (3rd edition)*. Dai-ichi Hoki, 2005. In Japanese.

[70] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2227–2237, 2018.

[71] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, 2016.

[72] Yoon-Hyung Roh, Ki-Young Lee, Sung-Kwon Choi, Oh-Woog Kwon, and Young-Gil Kim. Recognizing coordinate structures for machine translation of English patent documents. In *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation*, pp. 460–466, 2008.

[73] Michiaki Sato, Yasuhiro Ogawa, Takahiro Komamizu, and Katsuhiko Toyama. Automatic sentence extraction for Japanese statutes summarization using classifier systems. In *Proceedings of the 2018 Tokai-Section Joint Conference on Electrical, Electronics, Information, and Related Engineering*, 1 page, 2018. In Japanese.

[74] Yu Sawai, Mamoru Komachi, and Yuji Matsumoto. A learner corpus-based approach to verb suggestion for ESL. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 708–713, 2013.

[75] Julia Shaptala and Bohdan Didenko. Multi-headed architecture based on BERT for grammatical errors correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 246–251, 2019.

[76] Akira Shimazu. Structural paraphrase of law paragraphs. In *Proceedings of the Eleventh International Workshop on Juris-informatics*, pp. 62–75, 2017.

[77] Satoshi Shirai, Yoshifumi Ooyama, Satoru Ikehara, Masahiro Miyazaki, and Akio Yokoo. Introduction to "goi - taikei : a Japanese lexicon". Technical Report 106 (1998-IM-034), ATR Spoken Language Translation Research Laboratories and NTT Communication Science Laboratories, NTT Communication Science Laboratories, Faculty of Engineering Tottori University, Faculty of Engineering Niigata University, ATR Spoken Language Translation Research Laboratories, 1998. In Japanese.

[78] Keet Sugathadasa, Buddhi Ayesha, Nisansa de Silva, Amal Shehan Perera, Vindula Jayawardana, Dimuthu Lakmal, and Madhavi Perera. Legal document retrieval using document vector embeddings and deep learning. In *Intelligent Computing*, pp. 160–175, 2019.

[79] Kyoko Sugisaki. Towards data-driven style checking: An example for law texts. In *Legal Knowledge and Information Systems - JURIX 2016: The Twenty-ninth Annual Conference on Legal Knowledge and Information Systems*, pp. 93–100. IOS Press, 2016.

[80] Martin Sundermeyer, Ralf Schulüter, and Hermann Ney. LSTM neural networks for language modeling. In *Proceedings of the INTERSPEECH 2012*, pp. 194–197, 2012.

[81] Ben Swanson and Elif Yamangil. Correction detection and error type selection as an ESL educational aid. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 357–361, 2012.

[82] Koichi Takeda, Tetsunosuke Fujisaki, and Emiko Suzuki. CRITAC — a Japanese text proofreading system. In *Proceedings of the 11th International Conference on Computational Linguistics*, pp. 412–417, 1986.

[83] Michiaki Tanaka. "Deta no jidai" to "puraibashi no jidai" no ryoritsu. Technical report, NLI Research Institute, 2020.

[84] Hiroki Teranishi, Hiroyuki Shindo, and Yuji Matsumoto. Coordination boundary identification with similarity and replaceability. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*, pp. 264–272, 2017.

[85] Hiroki Teranishi, Hiroyuki Shindo, and Yuji Matsumoto. Decomposed local models for coordinate structure parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3394–3403, 2019.

[86] The Japanese Law Translation Council. Standard Legal Terms Dictionary (March 2018 Edition), 2018.

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

[88] Masato Uno and Akira Shimazu. Horeibun no kozoteki kakikae. In *Proceedings of the 18th Annual Meeting of the Association for Natural Language Processing*, pp. 427–430, 2012. In Japanese.

[89] Katsuhiko Utsubo. Automatic punctuation completion using LSTM. In *Proceedings of the 34th Annual Conference of the Japanese Society for Artificial Intelligence*, 4 pages, 2020. In Japanese.

[90] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30, pp. 5998–6008, 2017.

[91] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, 2019.

[92] Jaromír Šavelka and Kevin D. Ashley. Segmenting U.S. court decisions into functional and issue specific parts. In *Legal Knowledge and Information Systems - JURIX 2018: The Thirty-First Annual Conference on Legal Knowledge and Information Systems*, pp. 111–120. IOS Press, 2018.

[93] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference on the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1112–1122, 2018.

[94] Daisuke Yamada and Akira Shimazu. Horeibun no gengoteki tokucho wo riyoshita kadokusei kojo no tame no hyoji. In *Proceedings of the 12th Annual Meeting of the Association for Natural Language Processing*, pp. 196–199, 2006. In Japanese.

[95] Takahiro Yamakoshi, Tomohiro Ohno, Yasuhiro Ogawa, Makoto Nakamura, and Katsuhiko Toyama. Bunmyakujiyubunpo nimotozuku horeibun no heiretsukozo kaiseki. In *Proceedings of the 2016 Tokai-Section Joint Conference on Electrical, Electronics, Information, and Related Engineering*, 1 page, 2016. In Japanese.

[96] Tsuneyuki Yamamoto. *Jitsumu rippogijutsu*. Shojihomu, 2006.

[97] Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 180–189, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[98] Shoichi Yokoyama. Setsubiji ni chakumokushita tokyobun no heiretsukozo kaiseki. *Japio Year Book*, Vol. 2012, pp. 250–253, 2012. In Japanese.

[99] Katsumasa Yoshikawa. Identifying relevant civil law articles with judicial precedents. In *Proceedings of the Eleventh International Workshop on Juris-informatics*, pp. 142–155, 2017.

[100] Fan Zhang, Homa B. Hashemi, Rebecca Hwa, and Diane Litman. A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1568–1578, 2017.

[101] Fan Zhang and Diane Litman. Annotation and classification of argumentative writing revisions. In *Proceedings of the 10th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 133–143, 2015.

[102] Fan Zhang and Diane Litman. Using context to predict the purpose of argumentative writing revisions. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1424–1430, 2016.

[103] Geoffry Zweig and Christopher J.C. Burges. The Microsoft Research sentence completion challenge. Technical report, Microsoft Research, 2011.