

Semantic Analysis of Paragraphs Consisting of Multiple Sentences

Towards Development of a Logical Formulation System

Kenji TAKANO^a Makoto NAKAMURA^{b,1} Yoshiko OYAMA^a and
Akira SHIMAZU^b

^a *IR-ALT Inc., Japan*

^b *School of Information Science, JAIST, Japan*

Abstract. A logical formulation system functions to verify consistency of legal documents, and to eliminate inconsistent parts from a set of articles. We are studying legal document analysis methods, and in this paper we focus on how to deal with multiple sentences which constitute a paragraph of an article in a law. They need to be processed together because they are semantically dependent on each other. We analyzed National Pension Law of Japan and found that relations between sentences and their logical structures can be classified into four main types. We implemented a logical formulation system, which showed reasonable accuracy in dealing properly with paragraphs consisting of multiple sentences.

Keywords. Legal engineering, logical formulation, National Pension Law of Japan

Introduction

Our society is regulated by many laws, regulations, rules and customs, which relate to each other. This implies that laws are specifications of our society. From this view, we are studying Legal Engineering [1,2], which treats the systemic aspect of laws, in order to construct a trustworthy society. Particularly, this is meaningful for the coming e-Society.

In research from the above standpoint, we are developing legal document analysis methods, including translation from legal documents into logical forms, which functions to verify consistency of legal documents, and to eliminate inconsistent parts from a set of articles. In this paper we focus on how to analyze multiple sentences which constitute a paragraph of an article (provision) in a law. They need to be processed together because they are semantically dependent on each other.

Thus far, we have developed a system for logical formulation of law sentences [3]. This system, called WILDCATS², translates a law sentence into a logical formula, assigning content words in the sentence to a logical predicate. Even though WILDCATS shows high accuracy in terms of single sentences, it cannot deal with a number of sen-

¹Corresponding Author: Makoto Nakamura, School of Information Science, Japan Advanced Institute of Science and Technology, 1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan; E-mail: mnakamur@jaist.ac.jp

²WILDCATS is an acronym of “‘Wildcats’ Is a Legal Domain Controller As a Translation System.”

tences at once. A provision is expressed in a single sentence as a general rule, which makes the provision long and complicated. Many provisions, however, are written with a number of sentences in order to avoid ambiguous expressions. We focus on the processing of provisions consisting of multiple sentences, which are roughly categorized into the following items;

1. proviso – a sentence follows the preceding sentence as supplementation.
2. insertion with parentheses – one or more sentences are written within a pair of parentheses. In most cases it modifies the preceding word or sentence.
3. itemization – a number of phrases follow the main sentence.
4. (referential expression – some expressions refer to another provision, article, or law.)

Although the last item, referential expression, is not regarded as a provision with a number of sentences, other sentences related to the provision must be processed on the way to logical formulation. We have already reported about items 3 and 4 in [4]. In this paper, focusing on items 1 and 2, we propose a method to analyze relations among sentences in a provision, and show the effectiveness of our method, embedding the module into the current system. Our target is currently restricted to National Pension Law of Japan, which has characteristics mentioned in [5].

Although legislative style of Japan is prescribed in terms of constructing articles, paragraphs, items and so on in a law document, syntax of logical structure is confined to conjunctions for parallel noun phrases. In such a situation, we need to analyze legal sentences from the viewpoint of logical structures. There are studies about discourse structures, especially rhetorical structures. They mainly focus on semantic relations between sentences. Against such studies, this paper clarifies relations between multiple sentences in law paragraphs and their logical structures.

In this paper, we explain the characteristics of provisions in National Pension Law of Japan in Section 1. We show a semantic classification of provisions from the point of view of logical formulation of multiple sentences in Section 2, and we introduce implementation of our method into our current system in Section 3. We show experimental settings and results in Section 4, and conclude in Section 5.

1. Textual Characteristics of National Pension Law of Japan from the Standpoint of Logical Formulation

1.1. Legal Logical Structure

In most cases, a law sentence consists of a topic part, an antecedent part, and a consequence part. Structure of a sentence in terms of these parts is shown in Figure 1. These parts are composed from two parts of logical forms: a law requisite part and a law effectuation part [6]. In general, the law requisite part and the law effectuation part are related to logical implication (\rightarrow) or logical equivalence (\leftrightarrow). The problem is how to determine logical structure from the surface form. There are four cases in reflecting a surface pattern in the skeleton of a logical structure, two of which depend on which part in the logical structure a topic belongs to. The rest are cases in which the topic belongs to both parts, or in which there is no topic part.

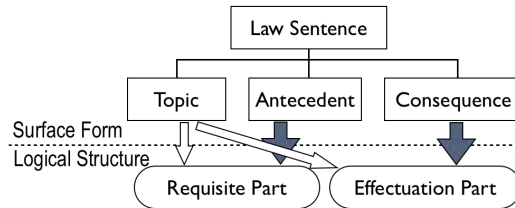


Figure 1. Structure of requisition and effectuation [6]

<p>Article 4 In terms of the premium for pensions established by this law,<i>Top</i> <u>when an extreme change in the living standard of the people or in other circumstances occurs,</u><i>Ant</i> <u>the revision should promptly be addressed to cope with the circumstances after the change.</u><i>Con</i></p> <p>第四条 この法律による年金の額は、<i>Top</i> <u>国民の生活水準その他の諸事情に著しい変動が生じた場合には、</u><i>Ant</i> <u>変動後の諸事情に因應するため、速やかに改定の措置が講ぜられなければならない。</u><i>Con</i></p>
--

Figure 2. Structure of requisite and effectuation (Article 4 in National Pension Law)

Figure 2 shows Article 4 as an example of the structure, where the indices *Top*, *Ant*, and *Con* denote topic, antecedent, and consequence, respectively. Linguistically speaking, topic words tend to move to the beginning of a sentence, regardless of syntactic case in a Japanese sentence. This is called topicalization.

The topic typically terminates with particular particles, and the antecedent with phrases corresponding to ‘if’ or ‘when.’ It is possible to make a pattern table of the cue phrases at which these parts are separated. If a sentence matches one of the patterns, clauses matching the pattern in the sentence can be assigned to the law requisite part, and the other clauses to the law effectuation part. Nakamura et al. [5] registered 33 patterns, which succeeded in covering 298 out of 335 sentences, in recognizing structure.

1.2. Article Consisting of Multiple Sentences

Many articles consist of a number of paragraphs. Each paragraph can independently be processed in logical formulation, because it generally has a logical structure of “law requisite” and “law effectuation.” The National Pension Law of Japan (as of August 20, 2007) consists of 216 Articles, and of 548 Paragraphs. Logical formulation should be taken into account by paragraph.

When a paragraph is expressed in two sentences, the latter sentence typically includes the phrase “In this case, ...” or “The same shall apply to ...”³ If the latter sentence starts from “Provided, however, that ...,” it is called a proviso. In most cases, provisos are used for an exception to the former condition. Some additional and exceptional conditions are inserted into the sentence, being enclosed in parentheses. In a paragraph consisting of multiple sentences, a supplementary sentence to the first sentence is regarded as follows;

- an additional case to the first sentence
- a proviso

³These expressions appear not only at the beginning of the sentence, but also at the end in Japanese.

- an inserted sentence enclosed in parentheses

Figure 3 shows that Article 20-2 in National Pension Law consists of the first sentence, a proviso, and an inserted sentence in parentheses. In this case, a number of logical formulas are necessary to represent its meaning. Note, however, that when a paragraph consists of a number of sentences as mentioned above, it is difficult to work out its correct logical formulas by translating sentences individually.

Article 20-2 (1)For pension benefits (3)(except the pension benefits for which payments of the full amount have been suspended according to another provision in this law or a provision in another law), when the recipients submit the petition, payments of the full amount shall be suspended; (2)provided, however, that when a portion of the amount of pension benefits is suspended according to another provision of this law or to a provision of another law, the rest of the amount of pension benefits shall be suspended.

Figure 3. Paragraph with a number of sentences (Article 20-2 in National Pension Law)

(1)1st sentence	For [pension benefits _{NP} .] _{T₁} when (the recipients submit the petition.) _{A₁} {payments of the full amount shall be suspended} _{C₁}
(2)2nd sentence	provided, however, that when (a portion of the amount of pension benefits is suspended according to another provision of this law or to a provision of another law.) _{A₂} {the rest of the amount of pension benefits shall be suspended} _{C₂}
(3)3rd (inserted) sentence	{except} _{C₃} (the pension benefits for which payments of the full amount have been suspended according to another provision in this law or a provision in another law) _{A₃}
logical structure 1	$\langle T_1 \rangle \wedge \neg \langle A_3 \rangle \wedge \langle A_1 \rangle \wedge \neg \langle A_2 \rangle \rightarrow \{C_1\}$
logical structure 2	$\langle T_1 \rangle \wedge \neg \langle A_3 \rangle \wedge \langle A_1 \rangle \wedge \langle A_2 \rangle \rightarrow \{C_2\}$

Figure 4. Logical structures of Article 20-2

1.3. Notation of Logical Structures

This paper focuses on logical structures such as requisite-effectuation structures and conjunction/disjunction structures in requisite and effectuation parts. So, we do not represent predicates in the logical representation of a sentence, and instead use character strings of the sentence. For example, for a sentence “when you enter a house, you take your shoes off,” we write

$\langle \text{you enter a house} \rangle \Rightarrow \{ \text{you take your shoes off} \}$

instead of writing

$\text{Enter}(x, y) \wedge \text{you}(x) \wedge \text{house}(y) \rightarrow \text{Take-off}(x, z) \wedge \text{shoes}(z).$

Figure 4 shows the logical structures of Article 20-2 corresponding to its surface form as an example, where T_i , A_i , and C_i denote topic, antecedent, and consequence parts of the i -th sentence of a paragraph.

[T] shows a topic of a sentence. A topic is related to an antecedent part of a sentence or a consequence part or both. When a topic is related to an antecedent, it is included in a requisite of the logical structure of the sentence. When a topic is related to a consequence, it is part of the effectuation part of the logical structure. X_{NP} shows a string of a noun phrase preceding an inserted sentence in parentheses. $\langle A \rangle$ shows that A is an antecedent

of a sentence, and will be a requisite main part of a logical structure for the sentence. {C} shows that C is a consequence of a sentence, and will be a main effectuation part of a logical structure for the sentence.

Article 20-2 is notated as follows, interpreting each sentence independently:

1. $\langle \text{pension benefits} \rangle \wedge \langle \text{the recipients submit the petition} \rangle \rightarrow \{ \text{payments of the full amount shall be suspended} \}$
 $(\langle T_1 \rangle \wedge \langle A_1 \rangle \rightarrow \{ C_1 \})$
2. $\langle \text{a portion of the amount of pension benefits is suspended according to another provision of this law or to a provision of another law} \rangle \rightarrow \{ \text{the rest of the amount of pension benefits shall be suspended} \}$
 $(\langle A_2 \rangle \rightarrow \{ C_2 \})$
3. $\langle \text{the pension benefits for which payments of the full amount have been suspended according to another provision in this law or a provision in another law} \rangle \rightarrow \{ \text{except} \}$
 $(\langle A_3 \rangle \rightarrow \{ C_3 \})$

However, the logical structure of this provision is not represented just by three independent logical structures. We must take into account context like the logical structures shown in Figure 4.

We made a pattern table which classifies the surface form of a paragraph into the skeletons of logical structures. Examples are shown in Figure 5. If phrases corresponding to the patterns are found in the sentences, the logical structures below are generated. The logical structures shown at the bottom in Figure 4 are derived from these patterns. The pattern table is explained in more detail in the next section.

Pattern of Type C1	
1st sentence	$\langle A_1 \rangle \{ C_1 \}$ $R_1 = \sim \text{ときは、} / \text{When } \sim \sim \text{においては、} / \text{in } \sim, \dots$ $E_1 = \dots (\text{verb})_e$
2nd sentence	ただし、 / provided, however, that $\langle A_2 \rangle \{ C_2 \}$ $C_2 = \sim \text{ときは、} / \text{when } \sim \sim \text{場合は、} / \text{in the case } \sim$
logical structure	$\langle A_1 \rangle \wedge \neg \langle A_2 \rangle \rightarrow \{ C_1 \}, \langle A_1 \rangle \wedge \langle A_2 \rangle \rightarrow \{ C_2 \}$
Pattern of Type E2	
sentence pattern	' X_{NP} '(A を除く。 / except A)
logical structure	$X_{NP} \wedge \neg A$

Figure 5. Patterns of Type C1 and Type E2

2. Classification of Mappings from a Paragraph to its Logical Structure

2.1. Logical Structure of the First Main and the Second Sentences of a Paragraph

A provision generally consists of the first and the second sentences in many cases. We call the first sentence the main sentence and the second sentence a subordinate sentence. The first or the second sentence generally contains inserted sentences in parentheses. In such a case, we also call the first or the second sentence the main sentence and the inserted sentence a subordinate sentence.

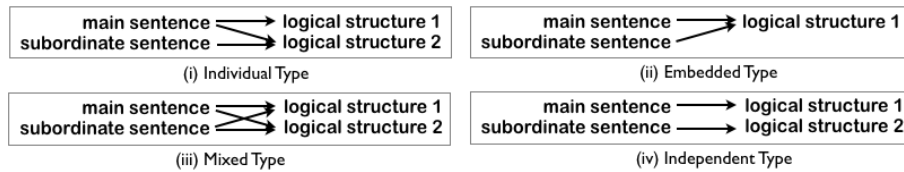


Figure 6. Relations between main and subordinate sentences and their logical structures

Table 1. Classification of correspondences of main and subordinate sentences by their logical structures

main – subordinate sentences	individual	embedded	mixed	independent
the first – the second sentences	A (4)	B (3)	C (3)	I
sentence – inserted sentences	D (5)	E (3)	F (1)	

We analyzed National Pension Law of Japan and found that relations between main and subordinate sentences, and their logical structures, can be classified into four types, as shown below.

Individual type Main and subordinate sentences are represented in two logical structures (Figure 6(i)). The first logical structure corresponds to the main sentence. The second logical structure corresponds to the subordinate sentence and some part of the main sentence.

Embedded type Main and subordinate sentences are represented in one logical structure (Figure 6(ii)).

Mixed type A main sentence and some part of a subordinate sentence are represented in the first logical structure, and the subordinate and some part of the main sentence are represented in the second logical structure (Figure 6(iii)).

Independent type A main sentence is represented in the first logical structure and a subordinate sentence is represented in the second logical structure (Figure 6(iv)).

2.2. Classification with a Main and a Subordinate Sentences

When a main sentence is the first sentence of a provision and a subordinate sentence is the second sentence, we label the sentences of “individual,” “embedded” and “mixed” types as Type A, B and C, respectively. When a main sentence is the first or the second sentence and a subordinate sentence is an embedded sentence in parentheses within the main sentence, we call these sentences of “individual,” “embedded” and “mixed” types Type D, E and F, respectively. Though we call “independent type” “Type I,” there is no Type I in the National Pension Law of Japan. As a result of our analysis, this classification is further divided based on surface linguistic expressions. We summarize the above in Table 1, where numerals in parentheses denote the number of subclasses in each type.

2.3. Correspondence between a Provision and a Logical Structure

This subsection describes how logical structures of a provision correspond to sentences of the provision. First, as a basic correspondence between sentences and logical struc-

tures, we show logical structures in which the first and the second sentences of a provision express the relation, and also logical structures of a single main sentence (the first or the second sentence) into which a second sentence is inserted in parentheses. Then, based on the basic correspondence, we show a correspondence between a provision and its logical structures.

Before explaining the correspondence, we introduce notations. We represent the first and the second sentence of a provision as S_1, S_2 , a sentence inserted in a parentheses of the first or the second sentence as S_j , and a substring of a noun phrase preceding a parentheses as X_{NP} . Then, if the parentheses is in S_1 , we describe S_1 as follows:

$$S_1 = \dots X_{NP}(S_j) \dots$$

We represent the first or the second sentence, removing its inserted sentences in parentheses as S'_i ($i = 1$ or 2). A logical structure of a sentence S_i is represented as $LS(S_i)$. We represent a logical structure based on the first and the second sentence of a provision, as $LS(S_1, S_2)$.

1. Based on the logical structure described in Section 2.1, we can represent logical structures for the first and the second sentences, S'_1, S'_2 , of a provision as follows:

$$LS_1(S'_1, S'_2) \text{ and } LS_2(S'_1, S'_2)$$

Precisely, logical structures of Type A are $LS_1(S'_1)$ and $LS_2(S'_1, S'_2)$. Type B are $LS_1(S'_1, S'_2)$. Type C are $LS_1(S'_1, S'_2)$ and $LS_2(S'_1, S'_2)$.

2. When logical structures of a sentence S_i and a sentence S_j inserted in parentheses within S_i are Type D, assuming that S_i is represented as follows:

$$S_i = \dots X_{NP}(S_j) \dots \quad (i = 1 \text{ or } 2)$$

logical structures of Type D are as follows:

$$LS_i(S'_i)(i = 1 \text{ or } 2) \text{ and } LS(S_j) \equiv A \leftrightarrow X_{NP}$$

where $S_i = \dots X_{NP}(S_j) \dots$, and S_j is “以下 A という。”(we say A in the following). Accordingly, we can represent logical structures of the first and the second sentences, and logical structures of a sentence (the first or the second) and an inserted sentence, independently.

3. When logical structures of a sentence S_i and a sentence S_j inserted in parentheses in S_i are Type E, assuming that S_i is represented as follows:

$$S_i = \dots X_{NP}(S_j) \dots \quad (i = 1 \text{ or } 2)$$

a logical structure of Type E is as follows:

$$LS_i(S_i) = LS_i(S'_i)|X_{NP} \wedge LS(S_j)/X_{NP}$$

This formula means that X_{NP} in $LS_i(S'_i)$ is replaced by $X_{NP} \wedge LS(S_j)$.

When there are a number of inserted sentences, we have the above logical structure for each inserted sentence, and each logical structure is constructed without affecting the other logical structures, assuming that X_{NP} does not appear in the inserted sentences. This assumption holds in the National Pension Law.

4. When logical structures of a sentence S_i and a sentence S_j inserted in parentheses within S_i are Type F, assuming that S_i is represented as follows:

$$S_i = \dots X_{NP}(S_j) \dots \quad (i = 1 \text{ or } 2)$$

and that $S_j = R X'_{NP}$ then logical structures of Type F1 are as follows:

$$LS(r(S'_i)) \wedge \neg LS(r(S_j)) \rightarrow LS(e(S'_i))$$

$$LS(r(S'_i)) \wedge LS(r(S'_j)) \rightarrow e(S'_j - X_{NP}) \wedge X'_{NP}$$

where $r(S'_i)$ and $e(S'_i)$ are the requisite and the effectuation parts of S'_i , and $S'_j - X_{NP}$ is a string of S'_j without X_{NP} . The above also holds in Type F3. The similarity holds in Type F2.

5. We showed how logical structures of main and subordinate sentences are constructed in the above, where we treated each case mentioned above independently. That is, logical structures of the first and the second sentences without inserted sentences do not depend on logical structures of the first or the second sentence and an inserted sentence. Accordingly, logical structures of a provision are constructed by first constructing logical structures of the first and the second sentences without inserted sentences, and then constructing for each inserted sentence logical structures of the first or the second sentence and the inserted sentence. When there are two-fold inserted sentences as “... (... (...) s_a ...) s_b ... s_c ,” we first construct logical structures of the deepest inserted sentence s_a and an inserted sentence s_b , and then construct logical structures of the inserted sentence s_b and the top level sentence s_c .

3. Implementation

Acquisition of logical formulas based on first order predicate logic by automatically reading natural language texts has widely been studied [7,8,9]. Our system, WILDCATS, derives logical forms from law sentences [3]. We modified this system to deal with multiple sentences. The following list is the procedure for multiple sentences in a paragraph.

1. Loading a target paragraph consisting of the main sentence, the second sentence and/or inserted sentences with parentheses.
2. Pulling the paragraph apart into independent sentences.
3. Analyzing morphology by JUMAN, the morphological analyzer of Japanese [10], and parsing target sentences by KNP, the Japanese dependency analyzer [11].
4. Searching patterns from the look-up table (see Figure 5) by cue phrases described in the paragraph, and assigning it to appropriate skeleton logical structures.
5. Analyzing clauses and noun phrases using a case frame dictionary.
6. Assigning variables and logical predicates. We generally assign verb phrases and *sahen*-nouns⁴ to both a logical predicate and an event variable, e_i . Then we assign other content words to both a case role predicate and a variable, x_j , representing an argument of a logical predicate.

⁴A *sahen*-noun is a noun which can become a verb with the suffix *-suru*.

7. Building logical formulas, and embedding fragments of logical connectives and predicates in the skeleton logical structures.

The procedure is roughly divided into two parts. One is to make the outside frame of the logical form (from Step 1 to Step 4, and Step 7) corresponding to the legal logical structure shown in Figure 1. The other (Steps 5 and 6) is for the inside frame. We assign noun phrases to bound variables and predicates using a case frame dictionary.

4. Experiments and Results

In this section, we show experimental results using our modified system. In particular, we focus only on testing whether our proposed method classifies a paragraph into a correct type of skeleton logical structure. In order to avoid inaccuracy by JUMAN and KNP, we manually modified the parsing output. Therefore, we prepared a set of paragraphs processed in advance from Step 1 to Step 3 in the procedure shown in Section 3. The other experiment on analyzing clauses and noun phrases using a case frame dictionary was mentioned in [5].

We examined the paragraphs in National Pension Law of Japan which have multiple sentences, except those of F type. The purpose of this experiment is to examine the accuracy of the pattern table in categorizing the surface structure of a paragraph consisting of multiple sentences into logical structures. Because the pattern table was manually made for National Pension Law of Japan, this experiment is a closed test. Thus far, we have currently defined 108 pattern rules, except for Type F and some subclasses of other types.

A successful example is shown in Figure 3. Article 20-2 has two elements to categorize; one is the combination between the first and the second sentences, and the other is between the main and the subordinate sentences. They correctly match Type C1 and E2, respectively. As a result, two logical skeletons are generated from Article 20-2. A paragraph could match more than one pattern, depending on the number of sentences.

Table 2 shows the experimental results for categorization of relations between sentences into Types A to F. The distribution of types is shown in the row ‘Distribution’ in the table. The row ‘Correct Answer’ denotes the number of patterns with which the system succeeded in guessing its type correctly. The row ‘Wrong type’ is counted if the system incorrectly guessed the relation as a wrong type. The row ‘No type’ denotes the number of relations which do not match any type.

Table 2. Number of paragraphs correctly categorized into Types A to F

Type	A	B	C	D	E	F
Distribution	8	24	1	40	34	8
Correct Answer	6 (75%)	23 (96%)	1 (100%)	38 (95%)	34 (100%)	0 (-)
Wrong Type	0 (0%)	0 (0%)	0 (0%)	2 (5%)	0 (0%)	0 (-)
No Type	2 (25%)	1 (4%)	0 (0%)	0 (0%)	0 (0%)	0 (-)

Article 18-2 and Article 70 failed to match Type A due to lack of patterns. In another case, two paragraphs, Paragraph 7, Article 12 and Paragraph 3, Article 128, which essentially belong to Type D, match not only its correct pattern but also an unexpected pattern in the same type. This is because the correct pattern is subsumed by another pattern. We recognized this case as failure, even though the system categorized the same type as its

correct answer. We need to avoid conflicts of the pattern table among types. For Type F and some subclasses of other types, we did not implement procedures necessary to construct logical structures and did not experiment. These problems will be solved in the next edition of the pattern list and the programs.

5. Conclusion

In this paper, we proposed how to deal with a paragraph consisting of multiple sentences in the processing of logical formulation of legal texts. Based on analyses of the National Pension Law of Japan, we defined four large types as possible relations between main and subordinate sentences in theory. We found cue phrases from the surface form of paragraphs to classify them into types, each corresponding to the skeleton of a logical formula. Experimental results showed that our system, WILDCATS, classified paragraphs into appropriate types with the pattern table for cue phrases.

Although this approach made it possible for our system to deal with multiple sentences in a paragraph, we have not yet evaluated our entire system from input of a paragraph to output of the logical formulas. That is one of the most important tasks in our future work.

References

- [1] Takuya Katayama. Legal Engineering – an Engineering Approach to Laws in e-Society Age –. *Proc. 1st Intl. WS on JURISIN*, 2007.
- [2] Takuya Katayama, Akira Shimazu, Satoshi Tojo, Kokichi Futatsugi, and Koichiro Ochimizu. e-Society and Legal Engineering (in Japanese). *J. the Japanese Society for Artificial Intelligence*, 23(4):529–536, 2008.
- [3] Makoto Nakamura, Shunsuke Nobuoka, and Akira Shimazu. Towards Translation of Legal Sentences into Logical Forms. *New Frontiers in Artificial Intelligence*, volume 4914 of *LNAI*, pp.349–362. Springer, 2008.
- [4] Yusuke Kimura, Makoto Nakamura, and Akira Shimazu. Treatment of Legal Sentences Including Itemized and Referential Expressions – Towards Translation into Logical Forms. *New Frontiers in Artificial Intelligence*, volume 5447 of *LNAI*, pp.242–253. Springer, 2009.
- [5] Makoto Nakamura and Akira Shimazu. Semantic Analysis of National Pension Law. *Proc. 3rd Intl. WS on JURISIN*, pp.83–94, 2009.
- [6] Kikuo Tanaka, Ichiro Kawazoe, and Hajime Narita. Standard Structure of Legal Provisions - for the Legal Knowledge Processing by Natural Language - (in Japanese). *IPSSJ Research Report on Natural Language Processing*, pp.79–86, 1993.
- [7] Jerry R. Hobbs, Mark Stickel, Paul Martin, and Douglas Edwards. Interpretation as Abduction. *Proc. 26th Annual Meeting on Association for Computational Linguistics*, pp.95–103, Morristown, NJ, USA, 1988. Association for Computational Linguistics.
- [8] Rutu Mulkar, Jerry R. Hobbs, and Eduard Hovy. Learning from Reading Syntactically Complex Biology Texts. *Proc. 8th Intl. Symposium on Logical Formalizations of Commonsense Reasoning, part of the AAAI Spring Symposium Series*, 2007.
- [9] Rutu Mulkar, Jerry R. Hobbs, Eduard Hovy, Hans Chalupsky, and Chin-Yew Lin. Learning by Reading: Two Experiments. *Proc. IJCAI 2007 WS on Knowledge and Reasoning for Answering Questions*, 2007.
- [10] Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. Improvements of Japanese Morphological Analyzer JUMAN. *Proc. WS on Sharable Natural Language Resources*, pp.22–28, 1994.
- [11] Sadao Kurohashi and Makoto Nagao. KN parser : Japanese Dependency/Case Structure Analyzer. *Proc. WS on Sharable Natural Language Resources*, pp.48–55, 1994.