Semantic Analysis of National Pension Law

Makoto Nakamura and Akira Shimazu

School of Information Science
Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan
{mnakamur, shimazu}@jaist.ac.jp

Abstract. This paper reports recent progress on the development of the system for translating legal documents into logical forms. As a study for clarifying themes which Legal Engineering pursues, we are treating National Pension Law of Japan, which provides administrative procedures of the national pension system. We investigate it from the linguistic viewpoint and find linguistic structural regularity of National Pension Law. We implement the translation system on the basis of the analysis. Our experimental result shows that it is promising to capture logical structures of National Pension Law according to its linguistic analyses.

1 Introduction

A new research field called *Legal Engineering* was proposed in the 21st Century COE Program, Verifiable and Evolvable e-Society [1, 2]. Legal Engineering serves for computer-aided examination and verification of whether a law has been established appropriately according to its purpose, whether there are logical contradictions or problems in the document per se, whether the law is consistent with related laws, and whether its revisions have been modified, added, and deleted consistently. One of the problems with which we deal in the field of Legal Engineering is to verify consistency of a legal document, otherwise, to eliminate inconsistent parts from a set of articles. One approach to verifying law sentences is to convert law sentences into logical or formal expressions and to verify them based on inference [3].

In order to accomplish the task, we translate legal documents into logical expressions, using some techniques in natural language processing. Acquisition of knowledge bases by automatically reading natural language texts has widely been studied so far, and is one of the main themes of the field of natural language processing [4]. Though the definition of semantic representation differs depending on what the language processing systems deal with, many systems try to generate logical forms based on first order predicate logic [5]. While researchers pursue the establishment of the fundamental technique, the usual conventional methods have so far focused on only translating an isolated sentence [6, 7]. However, a sentence must be interpreted with its related sentences. Especially, this is obligatory for procedural laws such as pension laws as shown in this paper. To properly treat a single sentence, we first need to analyze linguistic characteristics of sentences in a procedural law document.

The system which can fulfill our requirement for Legal Engineering needs to run on a variety of expressions appearing in the legal documents. In addition, it needs to deal with a batch of sentences in a document all together, taking into account the relations among them, because a law is expressed to make a sense with a complete set of sentences in itself. For example, some sentences enumerate conditions in the following items, refer to other articles concerning its conditions, include conditional sentences with parentheses, and add provisos about additional conditions followed by themselves.

Another problem to be solved in Legal Engineering is to develop a support system for an electronic society. In particular, we need to help the developers, linguistically analyzing procedural laws enacting the electronic society system. For example, if there is a system which derives the calculating formula of pension from the analysis of National Pension Law, it would reduce the developers' burden. To do that, we also need a system to interpret the legal documents logically.

From the point of the above mentioned view, we deal with National Pension Law, studying linguistic analysis. Thus far, we have developed a system for translation of legal documents into logical forms [8], and proposed a solution for itemized sentences [9]. In this paper, we report our ongoing research effort on the current system on the basis of the result of analysis of the law.

In Section 2, we describe the structure of National Pension Law, and in Section 3, we show our system for semantic analysis currently in development. In Section 4, we linguistically analyze the law. Section 5 reports experiments with our system, and we conclude in Section 6.

2 Characteristics of National Pension Law

As a study for clarifying themes which Legal Engineering pursues, we are treating National Pension Law of Japan. The themes we take up here are a methodology to assist design and modification of a law implementation information system and a methodology to verify the consistency of laws which are the base of the law implementation information system. The National Pension Law is really a law suitable for studying the methodologies.

National Pension Law of Japan provides administrative procedures of the national pension system. The Social Insurance Agency has a big information processing system, which implements and operates the National Pension Law. In the sense that the information processing system is implemented according to the law, the law is regarded as the specification of the information system. Indeed, the National Pension Law provides various administrative procedures for operating the national pension system, and the information processing system for the national pension really operates the administrative procedures.

The National Pension Law provides mainly (1) conditions of the insured and qualifications for receiving a pension, (2) provisions for premium, payment and its term, (3) types of pension benefits, (4) fashion for calculating an amount of benefits, (5) National Pension Fund. The following are some examples and explanation.

(1) Conditions of the insured and qualifications for receiving a pension The National Pension Law provides conditions concerning the insured and qualifications for receiving a pension such as necessary conditions for the insured, time for acquiring qualifications for receiving, time for losing his/her right, necessary term as the insured and time for receiving.

Though the articles of the National Pension Law are written in a style specific to Japanese law writing, as shown in the following examples, the logical content of most articles is described in a requisite part and an effectuation part.

For example, Article 7 provides what kind of persons are the insured and its logical content is as follows:

> Person A satisfies conditions (1), (2), ..., (n) $\Rightarrow$ A is the insured.

Specifically, Article 7 shown in Fig. 1 provides types of the insured, and Article 8 provides when the insured has qualifications for each type of the insured.

Article 7 A person who falls under any of the following items is the insured of National Pension:

Subscriber group No. 1 Persons who do neither receive a Welfare Pension nor are members of a mutual aid association and their unsupported spouses

Subscriber group No. 2 Recipients of a Welfare Pension or a mutual aid association

Subscriber group No. 3 Spouses supported by subscriber group No.2

Fig. 1. Article 7 in National Pension Law

Article 26 provides requirements for pension payment, that is, how much and how long the insured needs to pay a premium for receiving his/her pension. Its logical content is as follows:

> Person A pays a premium for a prescribed period $\Rightarrow$ A receives his pension.

(2) Regulations for premium, payment period, provision of pension and so on The National Pension Law provides concretely an amount of a premium, when a premium is collected, how long the insured must pay a premium, conditions for exempting a premium and so on. For example, Item 2 of Article 87 clarifies the time when a premium is collected and Item 3 describes the amount for each fiscal year.

(3) Types of a Pension The national pension system has several types of pension such as pension for an old-aged person, pension for a handicapped person and pension for a bereaved person. Article 15 defines the types of national pension and the section corresponding to each type of pension concretely provides requirements for payment, amount of pension, suspension of payment and so on for each type of pension. That is, each type of pension system has its own procedure. For example, the article for pension for a bereaved person provides who has a right to be paid.

(4) Calculation for an amount of pension The National Pension Law provides concretely how to calculate an amount of each type of pension. For example, the section of pension for an old-aged person provides how to calculate an amount of pension, taking into consideration periods of exempted premiums. However, though they are concrete arithmetic calculations, they are written in natural language and are not easy to read.

Many things described in the articles of the National Pension Law are procedures to treat the pension system and could be represented in some formal language concretely and clearly. However, they are written in the traditional writing style for laws and it causes unreadability. Furthermore, there have been a lot of modifications of the National Pension Law from its enactment. It seems that such modifications also causes many small conditions and supplementation.

An example of unreadability is the provisions about calculation of a pension. Another example is that there is a case that related things are written apart. The premium of a pension for an old-aged person is provided in the article of the section for expense, not for the pension for an old-aged person. The article next to the article for the premium of a pension for an old-aged person provides a premium for some pension but it is not clear what pension the article regulates. What the article provides is a premium for an additional pension and it is written referentially in the section for an additional pension.

From the above phenomena, it seems hard for officers in the government to enact laws and to modify them, preserving logical and systemic consistency. Seeing such things, we expect Legal Engineering, which supports works for enacting and modifying laws and rules in organizations, using information science and software technology.

3 Semantic Analysis using WILDCATS

In this section, we briefly explain our ongoing research to develop a translation system called WILDCATS [8].[1] WILDCATS translates legal sentences into logical forms, and can run under the Common LISP environment.

The following list is the procedure for one sentence. We repeat it when we process a set of sentences.

[1] WILDCATS is an acronym of " 'Wildcats' Is a Legal Domain Controller As a Translation System."

1. Analyzing morphology by JUMAN and parsing a target sentence by KNP.
2. Splitting the sentence based on the characteristic structure of a law sentence.
3. Assignment of modal operators with the cue of auxiliary verbs.
4. Making one paraphrase of multiple similar expressions for unified expression.
5. Analyzing clauses and noun phrases using a case frame dictionary.
6. Assigning variables and logical predicates. We generally assign verb phrases and *sahen*-nouns to a logical predicate and an event variable, e_i, and other content words to a case role predicate and x_j, which represents an argument of a logical predicate.
7. Building a logical formula based on fragments of logical connectives, modal operators, and predicates.

The procedure is roughly divided into two parts. One is to make the outside frame of the logical form (Step 1 to 3 and 7), which corresponds to the legal logical structure shown in Fig. 4. The other (Step 4 to 6) is for the inside frame. We assign noun phrases to bound variables and predicates using a case frame dictionary.

We show an example of the process for Article 4 shown in Figure 2, where the underlines and indices are explained in the later section. Figure 3 shows the output for Article 4, where small capital symbols such as LOC, OBJ, and GOL denote deep cases for *location*, *object*, and *goal*, respectively. The symbol DOU denotes a grammatical relation, which should be deleted. The logical expression basically depicts a relation between a verb and a noun with a deep case, and between two nouns. Some predicates remain isolated, because the system is still under development.

Article 4 When an extreme change about a living standard of nations or other circumstances occurs,$_{Cond}$ the revision$_{Obj}$ for the premium for pensions established by this law$_{Subj}$ should promptly be addressed$_{Prov}$ to cope with the circumstances after the change.$_{Cont}$

第四条 この法律による年金の額は、$_{Subj}$ 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、$_{Cond}$ 変動後の諸事情に応ずるため、$_{Cont}$ 速やかに改定の措置が$_{Obj}$ 講ぜられなければならない。$_{Prov}$

Fig. 2. Article 4 in National Pension Law

4 Characteristics of Linguistic Structure of National Pension Law

In Section 3, we showed outline of our current system, which has been repeatedly improved with our linguistic analyses of legal documents. To deal with National Pension Law, we especially reinforce the process for the structure of law sentences (Section 4.1), noun phrases with ‘A *no* B’ (Section 4.2), and case patterns of verb occur(e_9) $\wedge$ LOC(e_9,x_6) $\wedge$ nation(x_4) $\wedge$ living_standard(x_5,x_4) $\wedge$ circumstance(x_6) $\wedge$ OBJ(e_9,e_8) $\wedge$ extreme(x_7) $\wedge$ change(e_8) $\Rightarrow$ address(e_{18}) $\wedge$ GOL(e_{18},x_3) $\wedge$ pension(x_2) $\wedge$ amount(x_3,x_2) $\wedge$ OBJ(e_{18},e_{17}) $\wedge$ measure(e_{17}) $\wedge$ OBJ(e_{17},e_{16}) $\wedge$ revision(e_{16}) $\wedge$ to(x_{14}) $\wedge$ DOU(e_{13},x_{14}) $\wedge$ cope_with(e_{13}) $\wedge$ GOL(e_{13},x_{12}) $\wedge$ after_change(x_{11}) $\wedge$ circumstance(x_{12},x_{11}) $\wedge$ this(x_0) $\wedge$ law(x_1) $\wedge$ promptly(x_{15})

Fig. 3. Example of output for Article 4

phrases (Section 4.3). In addition, taking into consideration the relationships among statutory sentences, we analyze referential expressions (Section 4.4) and inserted statement (Section 4.5).

In this section, we describe the characteristics of National Pension Law on the basis of our linguistic analyses. We analyzed all the sentences in National Pension Law selected by the following procedure:

1. Inserted statements are removed from the main sentence and are regarded as independent sentences. Then, the identical sentences which repeatedly appear in parentheses are effective for once.
2. A sentence including items is pre-processed to deal with a number of declarative sentences, embedding the itemized expressions properly, according to the previous study [9].
3. Expressions which are not recognized as a sentence like titles of articles are removed.
4. Sentences which include expressions referring to other articles in National Pension Law are removed with some exceptions.

The total number of sentences are 1268 at the second item, and finally 299 sentences are mainly surveyed in this section. We call the 299 sentences "test set" in the rest of the paper. We also analyze referring expressions in sentences other than the test set.

4.1 Structure of Law Sentences

In most cases, a law sentence consists of a law requisite part and a law effectuation part, which designate its legal logical structure [10]. Structure of a sentence in terms of these parts is shown in Fig. 4. The law requisite part is further divided into a subject part and a condition part, and the law effectuation part is into an object, content, and provision part.

Figure 2 shows Article 4 as an example of the structure, where the indices of *Subj*, *Cond*, *Obj*, *Cont*, and *Prov* denote subject, condition, object, content, and provision, respectively. Since the order of words in Japanese is basically flexible, we often find sentences in which the order between object and content is switched like Article 4.

The subject and object parts typically terminate with particular particles, and the condition part with phrases corresponding to 'if' or 'when.' If a sentence matches one of the patterns, each clause in the sentence can be assigned to the

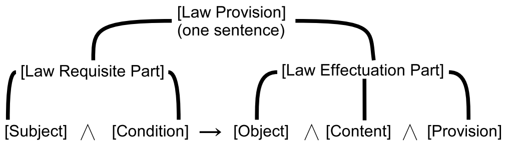

Fig. 4. Structure of requisition and effectuation [10]

subject part or the condition part in the law requisite part, and the rest to the law effectuation part. In general, the law requisite part and the law effectuation part are related to the logical implication (→) or the logical equivalence (↔).

Table 1. Registered patterns of the legal logical structure

Category	#Patterns
1: Subj + Cond + Cond + Prov	9
1-2: Cond + Cond + Prov	4
2: Subj + Cond + Prov	5
3: Cond + Subj + Prov	6
4: Cond + Prov	4
Others	5
Total	33

On the basis of our investigation of the structure for National Pension Law, we modified the set of patterns for the structure from the previous study [8]. Table 1 shows the latest registered patterns for extracting the structure from a sentence. In order to simplify the task, we focused only on the detailed parts of subject, condition, and provision, which can be found with case particles. We totally registered 33 patterns, each of which has a tuple of a logical structure and a set of case particles corresponding to the detailed parts in the structure. In the table, the label 'others' denotes special patterns for treating sentences which are different from the structure.

4.2 Semantic Structure of Noun Phrases with 'A no B'

Japanese has many noun phrase patterns of the type '*A no B*' consisting of two nouns *A* and *B* with an adnominal particle '*no*,' which is interpreted as some relation between *A* and *B*. This type of noun phrase has been widely studied by many researchers. Shimazu et al. [11] classified it into many semantic relations, according to the properties or functions of *A* and *B*. For example, if the noun *B* expresses an action or an event, *A* is its case element such as agent, object, and so on. In this case, *B* is typically a *sahen*-noun, which can become a verb with the suffix *-suru*. For example, '*tekiyou-suru*' (apply) is a verb while '*tekiyou*' (application) is a noun. We classified 'A *no* B' expressions into 5 cases. Table 2 shows a tuple of the typical logical form, the number of tokens, the number of types, and an example in each case.

From the viewpoint of representing the semantics of ‘*A* no *B*’ in logical forms, most of the expressions of ‘*A* no *B*’ consist of predicates corresponding to the words *A* and *B*, and to a relation between them as follows:

1. A logical form of typical expressions consists of predicates corresponding to *A* and *B*, and to a relation between them. *B* is a *sahen*-noun.
2. In a case that *B* functions as a case role such as location, and is restricted relatively by *A*.
3. In a case that *B* is an attribute of *A*, a logical form consists of two predicates corresponding to *A* and *B*.
4. A logical form of typical expressions consists of predicates corresponding to *A* and *B*, and to a relation between them. *A* is a *sahen*-noun.
5. *A* specifies *B*: (5-1) situational restriction (5-2) quantificational restriction (5-3) classificational restriction (5-4) relational restriction

Table 2. Frequency of A *no* B expressions in National Pension Law

Case	#Tokens	#Types	Logical Form
1	956	330	$A(x) \wedge B(e) \wedge rel(e, x)$
	kitei$_{(A)}$ *no* *tekiyou*$_{(B)}$ (application$_{(B)}$ of provision$_{(A)}$)		
2	272	71	$A(x) \wedge B(y, x)$
	(2)*shougai*$_{(A)}$ *no* *joutai*$_{(B)}$ (condition$_{(B)}$ of disability$_{(A)}$)		
3	190	57	$A(x) \wedge B(y, x)$
	(3)*ko*$_{(A)}$ *no* *kazu*$_{(B)}$ (number$_{(B)}$ of children$_{(A)}$)		
4	27	20	$A(e) \wedge B(y) \wedge rel(e, y)$
	shikyuu$_{(A)}$ *no* *youken*$_{(B)}$ (requirement$_{(B)}$ for payment$_{(A)}$)		
5-1	1075	450	$A(x) \wedge B(y, x)$
	tsuki$_{(A)}$ *no* *yokugetsu*$_{(B)}$ (next month$_{(B)}$ of the month$_{(A)}$)		
5-2	168	73	$A(y) \wedge B(y)$
	teido$_{(A)}$ *no* *shougai*$_{(B)}$ (degree$_{(A)}$ of disability$_{(B)}$)		
5-3	85	45	$A(y, x) \wedge B(y)$
	ta$_{(A)}$ *no* *nenkin kyuufu*$_{(B)}$ (other$_{(A)}$ pensions$_{(B)}$)		
5-4	3	3	$A(x) \wedge B(y, x)$
	shobun$_{(A)}$ *no* *fufuku*$_{(B)}$ (discontent$_{(B)}$ with punishment$_{(A)}$)		

4.3 Case Patterns of Verb Phrases

In the test set we found 1029 predicates, which consist of 191 kinds of verbs including *sahen*-nouns, causative forms and passive forms. There are 394 case patterns, each of which consists of a tuple of a predicate and case particles with deep cases. Furthermore, we investigated the expressions of ‘A *no* B,’ some of which represent a relationship between a case pattern and a predicate. We extracted case patterns from the expressions. As a result, we found 156 predicates, which consist of 54 kinds of *sahen*-nouns, and there are 54 case patterns. Totally, the number of predicates in the test set was 1185, the number of verbs was 215, and the number of case patterns was 439.

We show the list of the most frequent patterns in descending order in Table 3. The particle *attr* denotes an attributive form of a verb, which corresponds to a relative clause in English. An example is shown in the following noun phrase:

kakugou-ni$_1$ *sadameru*$_2$ *ritsu*$_3$ (The rate$_3$ which is fixed$_2$ at each item$_1$)

The noun '*ritsu*' has a relation to the predicate '*sadameru*' despite no case particle. We give a temporary particle '*attr*' to the noun modified by an attributive form of a verb, recognizing that it has the same deep case as the sentence "*kakugou-ni*$_1$ *ritsu-wo*$_2$ *sadameru*$_3$ (The rate$_2$ is fixed$_3$ at each item.$_1$)."

Table 3. List of the most frequent case patterns

Predicate	Case Particle	#
kakaru (係る)	*-ni, attr*	35
sadameru (定める)	*-no, attr*	26
kitei-suru (規定する)	*-de*	22
sadameru (定める)	*-de, attr*	17

4.4 Analysis of Referential Expressions

There are reference phrases in law sentences. For example, the phrase "X *-ni-gaitou-suru* Y (Y corresponding to that in X)" represents that the term Y appears in the article, provision, or some place described by X. These phrases are divided into two categories. One is the phrase from which the sentence referred is necessary for the analysis, called Category-A, and the other is, called Category-B, the phrase from which the sentence referred is unnecessary because the term necessary for the analysis appears in the referring sentence. We analyzed typical reference phrases from the test set before pre-processing for itemization. The result is shown in Table 4.

4.5 Analysis of Inserted Statement

In general, articles, paragraphs, and items consist of a number of sentences or phrases, some of which are inserted in a sentence with parentheses. There are some reasons for the use of inserted statements such as definition of terms. We investigated the inserted statements in the test set before pre-processing for itemization.

Regarding a set of sentences described in an article, paragraph, or an item as a unit, we counted the number of inserted statements appearing in a unit.

Table 4. Typical reference phrases in National Pension Law

Category-A		
X *-ni-oite*	prescribed in X	26
X *-wo-nozoku*	except X	4
Category-B		
X *-no-kitei-ni-yoru* Y_{noun}	Y_{noun} which is prescribed in X	140
X *-no-kitei-ni-yori* Y_{verb}	Y_{verb} as prescribed in X	114
X *-no* Y_{noun}	Y_{noun} prescribed in X	96
X *-ni-gaitou-suru* Y_{noun}	Y_{noun} corresponds to that prescribed in X	52

As a result, there are 753 units, and we found 269 inserted statements. Note that there are a number of sentences in some units. We classify the inserted statements into four categories, shown in Table 5.

Table 5. Classification of inserted statements

Class	#$^{(*)}$	note
Definition	55	Define a term appearing just prior to the statement
Numbering	24	Designate an article number in a case for referring another law
Effectuation	60	Specify, enlarge, or shrink the extent where the statement is effective
Condition	139	Add a condition to the statement

$^{(*)}$ There are 9 cases applying to two classes.

5 Experiments and Results

In order to exclude inaccuracy of the morphological and parsing processes from our experiments, we have manually modified the result from parsing sentences in advance. Therefore, the input to the system is the set of parsed trees which are accurate.

5.1 Experiment for the Structure for Requisite and Effectuation

For closed test, we tested 335 sentences which are produced from the target 299 sentences by separating compound sentences. The set of patterns covers 298 out of 335 sentences. The remaining 37 sentences are failed because of lack of the pattern, a number of the structures in a sentence, and so on. For open test, the system covers 39 out of 58 sentences, which are produced from 50 sentences.

5.2 Experiment for the Case-Frame Dictionary

On the basis of the analysis, we build a case frame dictionary, in which we can derive a set of deep cases from the tuple of a verb and case particles. It is a modification of the previous study [8].

Table 6. Coverage of case patterns

Test	#Pred	#Succ	#Fail	Coverage
Closed	1047	1027	20	98.0 %
Open	388	296	92	76.3 %

We examined how much the case frames extracted from the test set (299 sentences) cover the whole of the law. The higher the coverage is, the more accurate the assignment of the deep cases in output such as shown in Figure 3 is. We put a set of sentences for open test as all the sentence from Article 1 to Article 52-5 except those that were used for the case patterns. The result is shown in Table 6. The labels 'Test,' '#Pred,' '#Succ,' '#Fail,' and 'Coverage' denote a test type (Open/Close), the number of predicates to check, the number of Succeeded patterns, the number of failed patterns, and its coverage, respectively.

In the closed test, the reason that the coverage is less than 100 % may come from some mistakes on the way for the case frame dictionary. We are still investigating the causes of the low coverage in the open test.

6 Conclusion

In this paper, we reported recent progress on the development of the system for translating legal documents into logical forms. Different from the previous work [8], we have shifted to focusing on National Pension Law.

Because we have not yet finished implementation with respect to our linguistic analyses, we cannot examine the whole integrated system. The following list shows the remaining problems, some of which have already finished implementation and can run as stand alone.

1. In some cases a sentence includes a number of structures of requisition and effectuation. (within a sentence)
2. There are a lot of supplementary explanation such as insertion sentences or provisos. (within a provision)
3. In some cases a provision applies another one, e.g. "The provision of this article with respect to marriage shall apply in that case, その場合には婚姻に関する本条の規定を適用する．" (among provisions)
4. There are a lot of expressions by multiple sentences such as referring to detailed items, other articles or laws. (among articles)

In addition to the above list, we need to take into account the process of unification of predicate variables among multiple sentences. Implementing the solutions for the problems in the order of the list, the system could cover the relationship among statutory sentences over a wide range in National Pension Law.

Acknowledgment This research was partly supported by the 21st Century COE Program 'Verifiable and Evolvable e-Society' and Grant-in-Aid for Scientific Research (20300057 and 19650028). We would like to give special thanks to Yoshiko Oyama, Daiki Nomura and Kenji Takano, who helped us to analyze logical representations of National Pension Law and implement the system translating articles into logical representations. We also thank the anonymous three referees for their valuable comments that greatly improved the quality of this paper.

References

1. Katayama, T.: Legal engineering – an engineering approach to laws in e-society age –. In: Proc of the 1st Intl. Workshop on JURISIN. (2007)
2. Katayama, T., Shimazu, A., Tojo, S., Futatsugi, K., Ochimizu, K.: e-Society and legal engineering (in Japanese). Journal of the Japanese Society for Artificial Intelligence **23** (2008) 529–536
3. Hagiwara, S., Tojo, S.: Stable legal knowledge with regard to contradictory arguments. In: AIA'06: Proc of the 24th IASTED international conference on Artificial intelligence and applications, Anaheim, CA, USA, ACTA Press (2006) 323–328
4. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall PTR, Upper Saddle River, NJ, USA (2000)
5. Hobbs, J.R., Stickel, M., Martin, P., Edwards, D.: Interpretation as abduction. In: Proc of the 26th annual meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1988) 95–103
6. Mulkar, R., Hobbs, J.R., Hovy, E.: Learning from reading syntactically complex biology texts. In: Proc of the 8th International Symposium on Logical Formalizations of Commonsense Reasoning, part of the AAAI Spring Symposium Series. (2007)
7. Mulkar, R., Hobbs, J.R., Hovy, E., Chalupsky, H., Lin, C.Y.: Learning by reading: Two experiments. In: Proc of IJCAI 2007 workshop on Knowledge and Reasoning for Answering Questions. (2007)
8. Nakamura, M., Nobuoka, S., Shimazu, A.: Towards translation of legal sentences into logical forms. In Satoh, K. et al. eds.: New Frontiers in Artificial Intelligence: JSAI 2007 Conference and Workshops, Miyazaki, Japan, June 18-22, 2007, Revised Selected Papers. LNAI 4914, Springer (2008) 349–362
9. Kimura, Y., Nakamura, M., Shimazu, A.: Treatment of legal sentences including itemized and referential expressions In Hattori, H. et al. eds.: New Frontiers in Artificial Intelligence: JSAI 2008 Conference and Workshops, Asahikawa, Japan, June 11-13, 2008, Revised Selected Papers. LNAI 5447, Springer (2009) 242–253
10. Tanaka, K., Kawazoe, I., Narita, H.: Standard structure of legal provisions - for the legal knowledge processing by natural language - (in Japanese). In: IPSJ Research Report on Natural Language Processing. (1993) 79–86
11. Shimazu, A., Naito, S., Nomura, H.: Semantic structure analysis of Japanese noun phrases with adnominal particles. In: Proceedings of the 25th annual meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1987) 123–130