Towards Translation of Legal Sentences into Logical Forms

Makoto Nakamura, Shunsuke Nobuoka, 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 our ongoing research effort to develop a system which translates legal texts into logical forms in which we can check for inconsistency. Our logical formalization conforms to Davidsonian Style, which is suitable for languages allowing expressions with zero-pronouns such as Japanese. We examine our system with actual data of legal documents. As a result, the system was 78% of accurate in terms of deriving predicates with bound variables. We discuss our plan for further development of the system from the viewpoint of the following two aspects: (1) improvement of accuracy (2) formalization of output necessary for logical processing.

1 Introduction

In recent years, a new research field called *Legal Engineering* was proposed in order to achieve a trustworthy electronic society [1, 2]. Legal Engineering serves to examine and verify whether the following issues are satisfied:

- A law is established appropriately according to its purpose,
- There are no logical contradictions or no problems as a document *per se*,
- The law is consistent with related laws, and
- It is modified, added, and deleted consistently for its revision.

Legal Engineering also serves to design an information system which works based on laws. Towards the achievement of the goal of Legal Engineering, we need to develop a system with advanced technology which deals with electronically processed legal texts in computers. The core of the system roughly consists of two procedures:

1. Translating legal texts into logical forms
2. Proving consistency in terms of the given logical forms [3]

This paper reports our ongoing research effort to develop a system for the translation into forms. Hence, our purpose in this paper is to develop a system which translates legal texts into logical forms in which we can check for inconsistency.

Acquisition of knowledge bases by automatically reading natural language texts has widely been studied, and is one of the main themes of the field of natural language processing [4]. Because the definition of semantic representation differs depending on what the language processing systems deal with, a few systems try to generate logical formulae based on first order predicate logic [5]. A study of knowledge acquisition by Mulkar et al. [6, 7] is one of those systems. They extracted well-defined logical formulae from textbooks of biology and chemistry. Although the final stage of their work was to apply high school AP exam questions to the system in order to measure its ability relative to high school students, it is not yet as robust as our target which checks for inconsistency of a set of logical formulae.

Let us change the viewpoint to AI in law. Logical processing in the legal domain has widely been studied by AI researchers for a long time [8, 9]. They have, however, aimed at finding what kind of law can apply to a particular incident, not at proving law *per se*. Furthermore, most of the systems require manual transcription of legal documents and authoritative examples into logical formulae. Therefore, our system would help them as a pre-processor, which automatically processes law texts.

Because law sentences[1] have to describe the details of their intentions precisely, they are usually very long and thus complicated. These long and complicated sentences potentially have ambiguities in syntax, although ambiguous description must never be permitted. This is the primary reason that reading legal texts is more difficult for people than reading other daily-use documents. However, we consider that it is easier to process such characteristic expressions in legal texts with an appropriate method than that of daily-use documents. In this paper, we pay attention to linguistic characteristics in the surface form of legal texts, such as:

If-then Statement A law sentence can roughly be separated into two parts. The former part is called the law requisite part, and the latter is the law effectuation part. Thus, a sentence is basically described as an *'if-then'* statement [10].

Coordinate Rule There are some kinds of expressions for conjunction and disjunction, which are used in a different layer.

Conventional Expression Making a specific case frame dictionary is effective.

Our logical formalization conforms to Davidsonian Style [11, 12], in which a relation between events can be represented by a predicate which has more than one event variables without defining a higher order predicate form in order to express predicates referring to predicates. This style is suitable for languages allowing expressions with zero-pronouns, as seen in Japanese. We expect output of this style to be easily converted into other first order predicate logic forms [3].

In this research, we linguistically investigated legal documents such as sentences of the Income Tax Law (100 articles, 255 clauses, 247 items, among 244

[1] Throughout this paper, the word 'sentence' never indicates judicial decision nor logical formula, but linguistic meaning.

articles), sentences of the National Pension Law (100 articles among 148 articles) in addition to sentences in municipal law of Toyama Pref. and Chiyoda Ward, Tokyo, Japan (38 articles, 90 clauses). After that, taking into account the result of the investigation, we realized a system which generates a logical formula corresponding to an input sentence from law documents.

In Section 2 we first describe how to deal with law documents with the methodology of natural language processing, based on linguistic analyses. We describe the process of our system in Section 3. We examine our system with actual data of legal documents, and report its results in Section 4. Finally, we conclude and describe our future work in Section 5.

2 Linguistic Analysis of Legal Documents

In this section, we show some linguistic characteristics of legal texts, and then consider some grammatical constraints.

2.1 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, 13]. Structure of a sentence in terms of these parts is shown in Fig. 1.

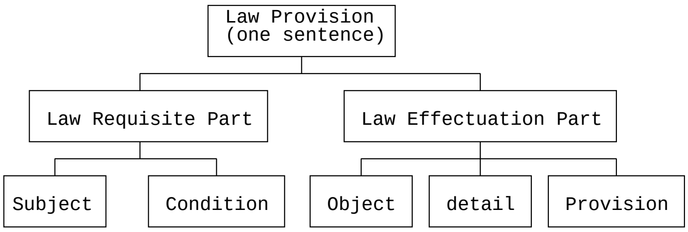

Fig. 1. Structure of requisition and effectuation [10]

Dividing a sentence into these two parts in the pre-processing stage makes the main procedure more efficient and accurate. We analyzed plain texts consisting of a total of 501 sentences from 138 provisions of the National Pension Law, and the municipal laws of Toyama Pref. and Chiyoda Ward in Tokyo Pref., and found 84 patterns of cue phrases which represent a combination of the subject part and the condition part as shown in Fig. 1. Example patterns are shown in Table 1, in which the subject part terminates 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 subject part or the condition part in the law requisite part, and the rest to the law effectuation part.

Table 1. Patterns for Subject Part and Condition Part

Subject	Condition
	‘...*suru-toki-wa,*’ (～するときは、)
‘...*wa,*’ (～は、)[Theme]	‘...*ni-tsuite-wa,*’ (～については、)
‘...*ga,*’ (～が、)[Nominative Case]	‘...*ni-oite-wa,*’ (～においては、)
‘...*mo,*’ (～も、)[too/also]	‘...*suru-baai-niwa,*’ (～する場合には、)
	‘...*niyori,*’ (～により、)

2.2 Analysis of Noun Phrases – Coordinate Structure

There are strict constraints in terms of coordinate structures which appear very frequently in law sentences [13]. For example, ‘*matawa*’ (又は) and ‘*moshikuwa*’ (若しくは), both of which are equivalent to an English word ‘or,’ have different precedence in embedding order. ‘*moshikuwa*’ (若しくは) is used in deeper embedding level than ‘*matawa*’ (又は). Therefore, a phrase ‘A *moshikuwa* B *matawa* C’ should be interpreted as:

$$((A \quad moshikuwa \quad B) \quad matawa \quad C) \qquad (1)$$

There are similar constraints for the other coordinate structure markers, too.

For a parallel phrase consisting of three or more coordinate noun phrases, the last one that follows a conjunction or disjunction, *e.g.* ‘*sonota-no*’ (その他の) corresponding to “or other,” tends to be a hypernym of the precedent noun phrases. An example is shown in the following expression:

機関に係る	申請、	届出	その他の	手続き等
(*kikan-ni-kakaru*	*shinsei,*	*todoke-de*	*sonota-no*	*tetsuzuki-tou*)
concerned with the organization	applications,	notifications	or other	procedures

The precedent words ‘applications’ and ‘notifications’ imply the last phrase ‘procedures.’ The first phrase ‘concerned with the organization’ should be considered to modify each of the following phrases. We examined the number of distinct expressions of conjunctive phrases from 38 provisions. As a result, we found 5 kinds of conjunctions or disjunctions used in parallel phrases.

Taking the characteristics of expressions into account, we cope with the problem of complexity in the hierarchical coordinate structure.

2.3 Analysis of Noun Phrases – Adnominal Particles ‘no’

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. [14] 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, '*teishutsu-suru*' (submit) is a verb while '*teishutsu*' (submission) is a noun.

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. A and/or B is a *sahen*-noun.

   shinseisho$_{(A)}$ *no* *teishutsu*$_{(B)}$ (申請書の提出)
   "submission$_{(B)}$ of an application form$_{(A)}$"

   申請書 $(x) \wedge$ 提出 $(e) \wedge \mathrm{obj}(e, x)$
   $\mathrm{application\_form}(x) \wedge \mathrm{submit}(e) \wedge \mathrm{obj}(e, x)$

2. In a case as B is an attribute of A, a logical form consists of two predicates corresponding to A and B.

   hi-hokensha$_{(A)}$ *no* *shimei*$_{(B)}$ (被保険者の氏名)
   "the name$_{(B)}$ of the person insured$_{(A)}$"

   被保険者 $(x) \wedge = ($氏名 $(x), n)$
   $\mathrm{person\_insured}(x) \wedge = (\mathrm{name}(x), n)$

3. In a case that A or B is a compound noun, a logical form of A or B generally consists of more than one predicate.

   hatachi-miman$_{(A)}$ *no* *mono*$_{(B)}$ (二十歳未満の者)
   "a person$_{(B)}$ below the age of twenty$_{(A)}$"

   者 $(x) \wedge = ($年齢 $(x), t) \wedge < (t, 20)$
   $\mathrm{person}(x) \wedge = (\mathrm{age}(x), t) \wedge < (t, 20)$

2.4 Davidsonian Formalization

Our logical formalization conforms to Davidsonian Style [11, 12], which is a logical formalism focusing on verb meanings, which are interpreted as properties of events or relations between individuals and events. In this style, an event in a proposition is expressed by predicates with an event variable, some of which are of thematic roles of the event. A relation between events can be represented by a predicate which has more than one event variables without defining a higher order predicate form in order to express predicates referring to predicates. Because words or phrases modifying a verb in a target sentence can directly be assigned to a predicate, this style is suitable for languages allowing expressions with zero-pronouns, as seen in Japanese.

Davidson's original motivation for his proposal comes from the phenomenon of adverbial modification [12]. The Davidsonian approach has been extended and modified by Parsons [11]. According to them, events should not be taken to be structured entities at all, but should be taken instead to be primitive entities. This approach includes an assumption that events are not a subclass of propositions. For example, in Parsons's theory, both action predicates like *run* and predicates that do not describe actions, like *is drunk*, express properties of eventualities: the difference between them is that, while the former expresses a property of events, the latter expresses a property of states. Sentences (2) and (4) are translated respectively as (3) and (5) in Parsons's theory (Tense is ignored here):

$$\text{Jones ran.} \tag{2}$$

$$\exists e(\text{running}(e) \wedge agent(e, j)) \tag{3}$$

$$\text{Jones was drunk.} \tag{4}$$

$$\exists e(\text{being-drunk}(e) \wedge patient(e, j)) \tag{5}$$

3 Language Processing

Here, we explain an outline of our system in the first subsection. The details of each process are described in the following subsections. Because the system is still being developed, some functions have not yet been implemented. For example, we do not deal with quantifiers.

3.1 Outline of Processing

The following list is the procedure for one sentence. We repeat it during processing a set of sentences.

1. Analyzing morphology and parsing a target sentence.
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 a paraphrase of some similar expressions to a unified expression.
5. Analyzing clauses and noun phrases using a case frame dictionary.
6. Assigning variables and predicates. We assign verb phrases and *sahen*-nouns to an event variable, e_i, and other content words to x_j.
7. Building a logical formula from the fragments of logical connectives, modal operators, and predicates.

In our system, we use JUMAN [15] and KNP [16], which are a Japanese morphological analyzer and a Japanese dependency analyzer, respectively. Both are representative tools for Japanese language processing.

Each sentence is divided depending on the structure consisting of a law requisite part and a law effectuation part. We assign a modal operator to the law effectuation part, based on the expression at the end of the sentence, which corresponds to auxiliary verbs in English. There are three types of modal operators, O, M, and P, which correspond to Obligation, Possibility (may), and Permission, respectively.

In order to assimilate a variety of similar representations into a unified logical form, we make paraphrases of particular expressions. We consider that legal texts are easier to analyze than daily-use documents because of unfamiliar but typical expressions, nevertheless this process is still necessary for stable output.

We describe details about case frame and noun phrase analyses in the next subsections.

3.2 Case Frame Analysis

Using a case frame dictionary, we can search for semantic relations between a verb and modifier nouns in a sentence. We assign semantic relations to predicates which connect to other predicates sharing a common event variable.

We built a case frame dictionary by extracting relations between verbs and their modifier nouns from 818 sentences of 366 provisions in 13 prefectures. As a result, a total of 517 verbs were registered into the dictionary. In the dictionary, each verb is an index and has a number of case frames, each of which stores semantic relations between nouns and the index verb. A case frame consists of a number of case slots, each of which is composed of a deep case, a case particle, a semantic category of nouns, and a set of example nouns and their frequency in use, as:

$$\begin{array}{rl} V: & \{CF_1, CF_2, \ldots, CF_n\} \\ CF_i: & \{CS_1, CS_2, \ldots, CS_m\} \\ CS_j: & (\text{[deep case]}, \text{[case particle]}, \text{[semantic category of nouns]}, \\ & \{(noun_1, freq_1), (noun_2, freq_2), \ldots\}) \quad , \end{array} \tag{6}$$

where V, CF_i, and CS_j denote an index of verb phrases, a case frame, and a case slot, respectively. An example of the dictionary is shown in Fig. 2. We manually annotated one deep case for each slot.

verb idx : *kakeru* [掛ける]
CF_1 [*meaning "hanging"*]:
((agent '-*wa* [は]' 'person [人]' (('he [彼]' 6)))
(object '-*wo* [を]' 'clothing [衣類]' (('jacket [上着]' 2)('coat [コート]' 4))))
CF_2 [*meaning "calling"*]:
((agent '-*wa* [は]' 'person [人]' (('I [私]' 2)))
(destination '-*ni* [に]' 'person [人]' (('friend [友達]' 1) ('mother [母]' 1)))
(object '-*wo* [を]' 'things [物]' (('telephone [電話]' 2))))

Fig. 2. Case Frame and Case Slot

We search for a case frame candidate corresponding to a verb modified by a number of sets of a noun and a case particle. If there are multiple candidates, we choose the one with the highest score by the following calculation method.

1. If a case particle of a noun matches one of the case slots, the current frame scores 5 points, otherwise, 2.5 points either for a sub particle, or for a hidden case used in '*A no B*' and a relative clause.
2. If one of the target nouns matches one of the examples in a case slot, the current frame obtains 5 more points.
3. Using a thesaurus of Japanese [17], we calculate a value of similarity between the head noun of the case element and an example noun stored in the case frame dictionary, and add the value to the score.

Here, we explain the calculation method of 3. Let w_1 be the head noun of the case element, and w_2 be an example noun stored in the case slot. The similarity between the two nouns is calculated as:

$$Sim_w(w_1, w_2) = \frac{2L}{l_1 + l_2} \quad , \tag{7}$$

where l_1 denotes the depth of w_1 from the root node in the thesaurus, l_2 is for w_2, and L denotes the depth of the least upper bound of the category node between w_1 and w_2. If we assume that the case frame dictionary holds n words for a particular case belonging to a predicate, then the similarity between a word, w_1, and the set of n examples, $w_{2,1}, \ldots w_{2,n} \in S$, is calculated in the following equation:

$$Sim_c(w_1, S) = \frac{\sum_{i=1}^{n} Sim_w(w_1, w_{2,i}) \times c_i}{\sum_{i=1}^{n} c_i} \quad , \tag{8}$$

where c_i denotes appearance frequency of $w_{2,i}$.

3.3 Noun Phrase Analysis

We especially put our efforts into analysis of noun phrases concerned with relative clauses and '*A no B*' relations. For a relative clause, because a predicate variable of the modificand noun phrase is shared by the events both of the relative clause and of the main verb, we took care in assignment of predicate variables. For example, for a sentence of "He hit the man who sold the book," the man is the agent of an event 'sell the book' as well as the object of the other event 'hit the man.'

For an '*A no B*' relation, because we regard a noun phrase with a *sahen*-noun as a verb, it is transformed to an event. Even though there is only one semantic relation to the event, it is easy to transform the noun phrase into a logical form with an event variable. If we introduced anaphora analysis for the event, we could generate better output, adding predicates in terms of obligatory cases. However, we can make a temporary result without obligatory cases in the current system. This is the reason that Davidsonian formalism is suitable for some languages allowing zero-pronouns.

4 Experiments and Results

We examined our system with actual data, which is an ordinance of Hiroshima city on a ban on dumping cans and cigarette butts, which consists of 61 predicate verbs in 20 provisions. Because of the difficulty of testing correctness of logical connectives or the logical formula itself, we focus on testing only whether the system correctly derives predicates, which correspond to words and semantic relations between nouns and verbs. Because we could not find any other models in terms of translating Japanese law sentences into logical forms, we do not compare experimental results with others.

We assume a baseline model which derives predicates of a semantic relation chosen by the surface form of a case particle instead of by the case frame dictionary. For example, a noun phrase with a case particle of '-*ga*' (が) is likely to become an agent. With some verbs, however, this particle has a different meaning, and the case frame dictionary may refuse to assign it to agent. As a result, our system realized 78.6% accuracy, while the baseline model was 61.2% accurate.

Here, we show an example of results, as follows. The following text is a provision of Hiroshima city, concerned with obligations of the mayor.

Hiroshima city provision 13-2 When the mayor designates a district for promoting beautification, s/he must in advance listen to opinions from the organizations and the administrative agencies which are recognized to be concerned with the district[2].

Our system worked out the following logical formula, in which the implication (→) forms the boundary between the law requisite part and the law effectuation part, and this formula includes a modal operator for obligation in the law effectuation part:

$$\begin{aligned}
&\mathrm{designate}(e_2) \wedge agt(e_2, x_0) \wedge \mathrm{mayor}(x_0) \wedge obj(e_2, e_1)\\
&\quad \wedge \mathrm{district\_for\_promoting\_beautification}(e_1)\\
&\rightarrow \mathrm{O}(\mathrm{listen}(e_{12}) \wedge agt(e_{12}, x_0) \wedge obj(e_{12}, e_{11})\\
&\quad \wedge \mathrm{opinion}(e_{11}) \wedge obj(e_{11}, x_{10}) \wedge \mathrm{administrative\_agency}(x_{10})\\
&\quad \wedge \mathrm{organization}(x_9)\\
&\quad \wedge \mathrm{recognize}(e_8) \wedge obj(e_8, x_{10}) \wedge obj(e_8, e_7)\\
&\quad \wedge \mathrm{concern}(e_7) \wedge res(e_7, x_6) \wedge \mathrm{district}(x_6)) \quad ,
\end{aligned} \tag{9}$$

where *agt*, *obj*, and *res* denote thematic roles of verbs in terms of agent, object and result, respectively. Because the noun phrase 'a district for promoting beautification' is a compound noun in Japanese, it is represented as one predicate. We point out incorrect parts as follows:

- The variable attached to 'district_for_promoting_beautification' should not be an event variable but an object as x_1.
- The predicate of '$obj(e_{11}, x_{10})$' should be '$agt(e_{11}, x_{10})$.'
- The predicate of 'organization(x_9)' is neglected. It should be '(administrative_agency(x_{10}) ∨ organization(x_{10})).'

[2] The original sentence is as follows: (広島市条例第 13 条 2 項)　市長は、美化推進区域を指定しようとするときは、あらかじめ、当該区域に関係すると認められる団体等および行政機関の意見を聴くものとする。

- There are two object terms modifying the predicate 'recognize(e_8).' One of the predicates '$obj(e_8, e_7)$' should be '$res(e_8, e_7)$.'
- The variable of the predicate 'district(x_6)' should be unified with 'e_1' (or 'x_1').

Owing to Davidsonian-style, some predicates can refer to events corresponding to event variables. An example is shown in a part of the output; 'listen(e_{12}) $\wedge$ $obj(e_{12}, e_{11})$ $\wedge$ opinion(e_{11}),' in which the object of the event 'listen' is the event 'opinion,' which is recognized as an event due to a *sahen*-noun in Japanese. Using this style, we can represent such a simple formula. Otherwise, we would have to define a higher order predicate logic form in order to express predicates referring to predicates.

As long as we do not realize anaphora analysis, the system cannot find out from the sentence the agent of the event 'recognize,' which is an obligatory case of the verb. Davidsonian style allows us to make a logical formula for the verb, regardless of the number of arities for which the predicate is necessary.

5 Concluding Remarks

Our purpose in this paper is to develop a system which translates legal texts into logical forms. We took into account linguistic characteristics of legal texts, regarding them as suitable for language processing.

We showed experimental results, in which our system provided high accuracy in terms of predicates corresponding to words and their semantic relations. Because the accuracy of semantic relations is mostly affected by that of the case frame dictionary, some errors in the example shown in Section 4 come from the lack of examples of the dictionary.

We reported our ongoing study in this paper. The rest of this section is spent on our plan for further development of the system. The remaining problems are concerned with (1) improvement of accuracy, and (2) formalization of output necessary for logical processing. Firstly, we expect to improve the analytical accuracy of our system by attaching syntactic rules to the parser. Particularly, the hierarchical rules for conjunction and disjunction would be effective for legal texts, as was mentioned in Section 2.2. In addition, making a case frame dictionary adequate for stochastic processing is important. The more texts provided to the language analyzer, the greater the size of case frame dictionary, eliminating the problem of data sparseness. In fact, our current database lacks word use frequency. Although we manually annotated each item of the database for deep case makers, in the next version we aim for the realization of automatic annotation.

Secondly, forming logical representation is very important for the next step in logical processing. Assignment of quantifiers to logical forms is necessary for logical processing. Because the Japanese language tends not to describe quantifiers explicitly, it is difficult to do this. We expect to solve the problem also by linguistic characteristics of legal texts. For example, we can interpret a quantifier of '$\forall x$ Citizen(x)' from the sentence "All the citizens have the right," while Japanese tends to lack the expression 'all.' We can, however, recognize that subjective nouns denoting a person or an organization in a law sentence tend to be applied to a universal quantifier, while objective nouns are applied to an existential quantifier. Therefore, we could make a logical form by attaching quantifiers from the sentence "The mayor can dismiss a deputy mayor." to "$\forall x \exists y \exists e P(\text{mayor}(x) \land \text{dismiss}(e) \land \text{deputy\_mayor}(y) \land \text{agent}(e, x) \land \text{object}(e, y))$."

We have two strategies about modal operators. On the one hand, we consider adding other kinds of modal operators in order to allow flexible expressions. Tense operator is one of the major candidates. On the other hand, we consider removing modal operators from the formalization of logic representation, in order to realize smooth logical processing. This problem is a trade-off between language processing and logical processing.

Acknowledgment This research was partly supported by the 21st Century COE Program 'Verifiable and Evolvable e-Society.' We would like to give special thanks to Akio Kitada and Akira Ejiri, who developed a pilot version of the system as members of our laboratory.

References

1. Katayama, T.: The current status of the art of the 21st COE programs in the information sciences field (2): Verifiable and evolvable e-society - realization of trustworthy e-society by computer science - (in Japanese). IPS Japan (Information Processing Society of Japan) Journal **46**(5) (2005) 515–521
2. Katayama, T.: Legal engineering – an engineering approach to laws in e-society age –. In: Proc. of the 1st Intl. Workshop on JURISIN. (2007)
3. Hagiwara, S., Tojo, S.: Stable legal knowledge with regard to contradictory arguments. In: AIA'06: Proceedings 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: Proceedings 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: Proceedings 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: Proceedings of IJCAI 2007 workshop on Knowledge and Reasoning for Answering Questions. (2007)
8. Nitta, K., Shibasaki, M., Sakata, T., Yamaji, T., Xianchang, W., Ohsaki, H., Tojo, S., Kokubo, I., Suzuki, T.A.: New HELIC-II: a software tool for legal reasoning. In: ICAIL '95: Proceedings of the 5th international conference on Artificial intelligence and law, New York, NY, USA, ACM Press (1995) 287–296

9. Ashley, K.D.: Modeling Legal Argument: Reasoning with Cases and Hypotheticals. Artificial Intelligence and Legal Reasoning Series. MIT Press (Bradford), Cambridge, Massachusetts (1990)
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. Parsons, T.: Events in the Semantics of English: A Study in Subatomic Semantics. MIT Press, Cambridge, MA (1990)
12. Zucchi, A.: The Language of Propositions and Events. Studies in Linguistics and Philosophy. Kluwer, Dordrecht (1993)
13. Nagai, H., Nakamura, T., Nomura, H.: Skeleton structure acquisition of Japanese law sentences based on linguistic characteristics. In: Proc. of NLPRS'95, Vol 1. (1995) 143–148
14. 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
15. Kurohashi, S., Nakamura, T., Matsumoto, Y., Nagao, M.: Improvements of Japanese morphological analyzer JUMAN. In: Proceedings of the Workshop on Sharable Natural Language Resources. (1994) 22–28
16. Kurohashi, S., Nagao, M.: KN parser : Japanese dependency/case structure analyzer. In: Proceedings of the Workshop on Sharable Natural Language Resources. (1994) 48–55
17. Ikehara, S., Miyazaki, M., Yokoo, A., Shirai, S., Nakaiwa, H., Ogura, K., Ooyama, Y., Hayashi, Y.: Nihongo Goi Taikei – A Japanese Lexicon (In Japanese). Iwanami Shoten, Tokyo (1997)