Title	Towards Translation of Legal Sentences into Logical Forms
Author(s)	Nakamura, Makoto; Nobuoka, Shunsuke; Shimazu, Akira
Citation	Lecture Notes in Computer Science, 4914: 349-362
Issue Date	2008
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/7879
Rights	This is the author-created version of Springer, Makoto Nakamura, Shunsuke Nobuoka, and Akira Shimazu, Lecture Notes in Computer Science, 4914, 2008, 349-362. The original publication is available at www.springerlink.com, http://dx.doi.org/10.1007/978-3-540-78197-4_33
Description	

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 proposes a framework for translating legal sentences into logical forms in which we can check for inconsistency, and describes the implementation and experiment of the first experimental system. 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. Otherwise, it is necessary to define a higher order predicate form in order to express predicates referring to predicates. This was pointed out in the previous researches. For example, Yoshino [13] proposed his own representation. On this aspect, we use the well-known representation formalism. We expect output of this style to be easily converted into other first order predicate logic forms [3].

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

In this research, we linguistically investigated legal documents such as sentences of the Income Tax Law (100 articles, 255 clauses, 247 items, among 244 articles), sentences of the National Pension Law (100 articles among 148 articles) in addition to sentences in municipal laws of Toyama Pref. and Chiyoda Ward, Tokyo, Japan (38 articles, 90 clauses). Taking into account the investigation, we propose a framework for generating a logical formula corresponding to an input legal sentence. Based on the framework and investigation, we implement the first version which realize the framework.

In Section 2 we first describe how to deal with law documents with the methodology of natural language processing, based on linguistic analyses. We propose a framework of our system in Section 3. We show the implementation of the first version and its experimental 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 aim at finding an appropriate method of replacing a law sentence to a logical formula, investigating some linguistic characteristics of legal texts. In Section 2.1, we explain our logical formalization, which is suitable for Japanese legal texts, and then consider some grammatical constraints in the following subsections.

2.1 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 in (1) and (2) are translated respectively as the following formulae (Tense is ignored here):

$$\text{Jones ran:} \quad \exists e(\text{running}(e) \wedge agent(e, j)) \tag{1}$$

$$\text{Jones was drunk:} \quad \exists e(\text{being-drunk}(e) \wedge patient(e, j)) \tag{2}$$

2.2 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, 14]. Structure of a sentence in terms of these parts is shown in Fig. 1. The law requisite part is further divided into a subject part and a condition part, and the law effectuation part is into an object, detail, and provision part.

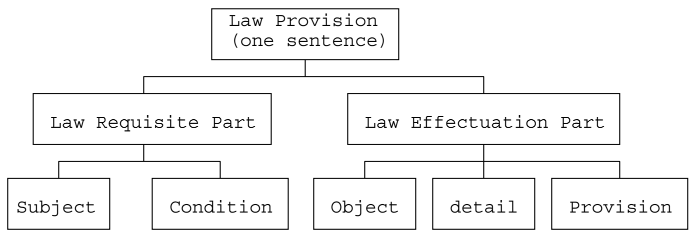

Fig. 1. Structure of requisition and effectuation [10]

Dividing a sentence into these two parts in the pre-processing stage makes the main procedure more efficient and accurate. Nagai et al. [14] proposed an acquisition model of this structure from Japanese law sentences. Dealing with strict linguistic constraints of law sentences, their model succeeded to acquire the structures at fairly high accuracy for a simple method, which specifies surface forms of law sentences. Our approach is different from theirs in that we consider some semantic analyses in order to represent logical formula.

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 and object parts[2] 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 subject part or the condition part in the law requisite part, and the rest to the law effectuation part. The object part includes phrases with a dative case particle in addition to ones for the subject part. For the law effectuation part, the provision part denotes the predicate in the main clause of a sentence, which is the last phrases including the main verb in the sentence, and the remaining phrase(s) between the object part and the provision part is assigned to the detail part.

[2] For the object part, we analyzed 38 provisions of the municipal laws of Toyama Pref. and Chiyoda Ward in Tokyo Pref.

Table 1. Patterns for Subject, Object and Condition Parts

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

Let us consider how to associate the structure with the logical framework. In general, the law requisite part and the law effectuation part are related to the logical implication ($\rightarrow$) or the logical equivalence ($\leftrightarrow$)[3]. In order to classify phrases of a sentence into the detailed five parts in the Fig. 1, we separate a sentence into clauses based on the cue phrases. After that, we assign the sentence to a logical frame according to its structure, shown in the following items:

- The law requisite part consists of a subject part and a condition part:

$$(\text{Subject}) \wedge (\text{Condition}) \rightarrow (\text{Provision})$$

- The law effectuation part consists of an object part and a provision part:

$$(\text{Condition}) \rightarrow (\text{Object}) \wedge (\text{Provision})$$

- There is neither a subject part or an object part:

$$(\text{Condition}) \rightarrow (\text{Provision})$$

The condition part may appear twice, or may not, depending on the clauses.

According to our investigation about the first 60 sentences described in the National Pension Law, we found that 32 of the sentences included the implication and 20 sentences for the equivalence. Therefore, most of the sentences have the structure consisting of both the law requisite and effectuation parts. We found cue phrases from 30 out of 52 sentences including the implication or equivalence.

Modal Operator In Japanese law sentences, there exist many kinds of expressions, where the difference of the meaning among them is sometimes delicate. Particularly, such delicate expressions appear at the end of the sentence, which corresponds to auxiliary verbs in English. Therefore, we consider to assign a modal operator to the law effectuation part. The modal operators specify the meaning of the sentence such as obligation, permission, possibility, and inhibition. We generally classify modal linguistic expressions and assign each expression to its corresponding modal operator. Strictly speaking, there are expressions

[3] We are still examining the cue phrases for distinguishing the logical implication ($\rightarrow$) and the logical equivalence ($\leftrightarrow$), which tends to be used for the definitions of words. In this paper, we focus on the logical implication.

which does not correspond to such a modal operator in some cases. For example, "*monotosuru*" generally corresponds to obligation but does not represent obligation in some cases. In order to have accurate logical representation, we must analyze such expressions more and clarify how many kinds of such operators we need to represent Japanese legal sentences and what kind of information we can use to transform expressions into such modal operators.

2.3 Analysis of Noun Phrases – Coordinate Structure

There are strict constraints in terms of coordinate structures which appear very frequently in law sentences [14]. 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:

$$((A \;\; moshikuwa \;\; B) \;\; matawa \;\; C) \tag{3}$$

We replaced the disjunction markers into logical connectives, that is, '∨,' regardless of the embedding order. Hence, the above expression is translated into the following logical formula:

$$((A \;\vee\; B) \;\vee\; C) \tag{4}$$

There are similar constraints for the other coordinate structure markers, too. The conjunctions, '*oyobi*' (及び) and '*narabini*' (並びに), correspond to an English word 'and,' which is replaced by '∧.'

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.

2.4 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. [15] 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. In the accordance with the previous works, we manually made transformation rules from 430 sentences in 110 provisions. As a result, we classified 'A *no* B' expressions into 10 patterns. We show three examples in the next paragraph.

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

3 Proposal of Framework

Here, we explain an outline of our framework, which reflects our linguistic investigation mentioned in Section 2. 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 a predicate and 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.

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. 1. 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.

4 Implementation and Experiments

4.1 Implementation

In the current system, we use JUMAN [16] and KNP [17], which are a Japanese morphological analyzer and a Japanese dependency analyzer, respectively. Both are representative tools for language processing of Japanese. A Japanese thesaurus [18] is also used for the calculation of similarity among words.

Each sentence is divided depending on the structure consisting of a law requisite part and a law effectuation part. In the current system, there are three types of modal operators, O, M, and P, which correspond to Obligation, Possibility (may), and Permission, respectively. Especially, a sentence expressing a ban on something is represented by the use of the modal operator for Permission with negation ($\neg P$).

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.

4.2 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,

verb idx : *kakeru* [掛ける]
CF$_1$ [*meaning "hanging"*]:
((**DC:** agent **CP:** '-*ga* [が]' **SEM_CAT:** 'person [人]'
examples: (('he [彼]' 6)))
(**DC:** object **CP:** '-*wo* [を]' **SEM_CAT:** 'clothing [衣類]'
examples: (('jacket [上着]' 2)('coat [コート]' 4))))
CF$_2$ [*meaning "calling"*]:
((**DC:** agent **CP:** '-*ga* [が]' **SEM_CAT:** 'person [人]'
examples: (('father [父]' 2)))
(**DC:** destination **CP:** '-*ni* [に]' **SEM_CAT:** 'person [人]'
examples: (('friend [友達]' 1) ('mother [母]' 1)))
(**DC:** object **CP:** '-*wo* [を]' **SEM_CAT:** 'things [物品]'
examples: (('telephone [電話]' 2))))
CF$_3$ [*meaning "sitting"*]:
((**DC:** agent **CP:** '-*ga* [が]' **SEM_CAT:** 'person [人]'
examples: (('Bob' 1)('mother [母]' 1)))
(**DC:** instrument **CP:** '-ni [に]' **SEM_CAT:** 'things [物品]'
examples: (('chair [椅子]' 2)))
(**DC:** object **CP:** '-wo [を]' **SEM_CAT:** 'body [身体]'
examples: (('waist [腰]' 2))))

DC, **CP**, and **SEM_CAT** denote deep case, case particle, and semantic category, respectively.

Fig. 2. Case Frame and Case Slot

a semantic category of nouns, and a set of example nouns and their frequency in use, as:

$$\begin{aligned} V &: \{CF_1, CF_2, \ldots, CF_n\} \\ CF_i &: \{CS_1, CS_2, \ldots, CS_m\} \\ CS_j &: ([\text{deep case}], [\text{case particle}], [\text{semantic category of nouns}], \\ &\quad \{(noun_1, freq_1), (noun_2, freq_2), \ldots\}) \quad , \end{aligned} \tag{5}$$

where V, CF_i, and CS_j denote an index of verb phrases, a case frame, and a case slot, respectively. We manually annotated one deep case for each slot. We show in Fig. 2 an example of the dictionary, in which a verb '*kakeru*' has three case frames.

We search for a case frame candidate corresponding to a verb modified by a number of sets of a noun and a case particle. Let us consider which case frame in Fig. 2 the following word '*kakeru*' belongs to: "*Kare-ga watashi-ni denwa-wo kakeru* [彼が私に電話を掛ける]," meaning "He calls me." First of all, we screen out the case frames which do not match the number of case frames. Therefore, CF_1

leaves out of the selection, and CF_2 and CF_3 become the candidates because of the same case particles.

When 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 [18], 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:

$$Sim_w(w_1, w_2) = \frac{2L}{l_1 + l_2} \quad , \tag{6}$$

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:

$$Sim_c(w_1, S) = \frac{\sum_{i=1}^{n} Sim_w(w_1, w_{2,i}) \times c_i}{\sum_{i=1}^{n} c_i} \quad , \tag{7}$$

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

For the example sentence, the case frame CF_2 gets 45.5 points, while CF_3 gets 28.4. Therefore, our system consequently chooses CF_2 as the appropriate case frame of the sentence. Because the score greatly depends on the example words of the case frame dictionary, it is important to extract the case frames from the large number of actual law texts.

4.3 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.

4.4 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%.

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[4].

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:

$$\begin{aligned}
&\text{designate}(e_2) \wedge agt(e_2, x_0) \wedge \text{mayor}(x_0) \wedge obj(e_2, e_1)\\
&\qquad \wedge \text{district_for_promoting_beautification}(e_1)\\
&\rightarrow \text{O}(\text{listen}(e_{12}) \wedge agt(e_{12}, x_0) \wedge obj(e_{12}, e_{11})\\
&\qquad \wedge \text{opinion}(e_{11}) \wedge obj(e_{11}, x_{10}) \wedge \text{administrative_agency}(x_{10})\\
&\qquad \wedge \text{organization}(x_9)\\
&\qquad \wedge \text{recognize}(e_8) \wedge obj(e_8, x_{10}) \wedge obj(e_8, e_7)\\
&\qquad \wedge \text{concern}(e_7) \wedge res(e_7, x_6) \wedge \text{district}(x_6)) \quad ,
\end{aligned} \tag{8}$$

[4] 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。

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:

1. The variable attached to 'district_for_promoting_beautification' should not be an event variable but an object as x_1[5]. Hence, '$obj(e_2, e_1)$' should be '$obj(e_2, x_1)$.'
2. The predicate of '$obj(e_{11}, x_{10})$' should be '$agt(e_{11}, x_{10})$.'
3. The predicate of 'organization(x_9)' is neglected. It should be '(administrative_agency(x_{10}) $\vee$ organization(x_{10})).'
4. There are two object terms modifying the predicate 'recognize(e_8).' One of the predicates '$obj(e_8, e_7)$' should be '$res(e_8, e_7)$.'
5. The variable of the predicate 'district(x_6)' should be unified with 'x_1.'

The problems of 2 and 4 come from the lack of examples in the case frame dictionary. The problems of 3 and 5 took place, because we have not yet dealt with an appropriate semantic analysis to unify these variables.

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 research purpose 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, proposed a processing framework and showed the implementation of the present version.

Our present 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

[5] This problem has already been solved.

accuracy of our system by attaching syntactic rules to the parser. Particularly, complex sentences with paragraphs would be analyzed by structural analysis of law article sentences, and the hierarchical rules for conjunction and disjunction would be effective for legal texts, as was mentioned in Section 2.3. 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 "$\forall x \exists y \exists e P(\mathrm{mayor}(x) \wedge \mathrm{dismiss}(e) \wedge \mathrm{deputy\_mayor}(y) \wedge \mathrm{agent}(e, x) \wedge \mathrm{object}(e, y))$."

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' and Grant-in-Aid for Scientific Research. 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, and to Yoko Shibata, Kaori Kohno and Yoshiko Oyama, who analyzed law sentences and their logical representations.

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. Yoshino, H.: Representation of legal knowledge by compound predicate formula. In: Proc. of Workshop on Legal Application of Logic Programming, ICLP' 94. (1994) 128–137
14. 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
15. 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
16. 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
17. Kurohashi, S., Nagao, M.: KN parser : Japanese dependency/case structure analyzer. In: Proceedings of the Workshop on Sharable Natural Language Resources. (1994) 48–55
18. 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)