Treatment of Legal Sentences Including Itemization Written in Japanese, English and Vietnamese – Towards Translation into Logical Forms –

Makoto Nakamura[†], Yusuke Kimura[†], Minh Quang Nhat Pham[†], Le Minh Nguyen[†] and Akira Shimazu[†]

This paper reports how to treat legal sentences including itemized expressions in three languages. Thus far, we have developed a system for translating legal sentences into logical formulae. Although our system basically converts words and phrases in a target sentence into predicates in a logical formula, it generates some useless predicates for itemized and referential expressions. In the previous study, focusing on Japanese Law, we have made a front end system which substitutes corresponding referent phrases for these expressions. In this paper, we examine our approach to the Vietnamese Law and the United States Code. Our linguistic analysis shows the difference in notation among languages or nations, and we extracted conventional expressions denoting itemization for each language. The experimental result shows high accuracy in terms of generating independent, plain sentences from the law articles including itemization. The proposed system generates a meaningful text with high readability, which can be input into our translation system.

Key Words: Legal Engineering, Itemization, Vietnamese Law

1 Introduction

A new research field called *Legal Engineering* was proposed in the 21st Century COE Program, Verifiable and Evolvable e-Society (Katayama 2005, 2007; Katayama, Shimazu, Tojo, Futatsugi, and Ochimizu 2008). 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 approach to verifying law sentences is to convert law sentences into logical or formal expressions (Nakamura, Nobuoka, and Shimazu 2008) and to verify them based on inference (Hagiwara and Tojo 2006a, 2006b).

Thus far, in order to take charge of text processing, we have developed a system for automatically converting Japanese legal documents into logical forms (Nakamura et al. 2008). The system

[†]School of Information Science, Japan Advanced Institute of Science and Technology

analyzes law sentences, determines logical structures, and then generates logical expressions. We have shown our system provides high accuracy in terms of generating logical predicates corresponding to words and their semantic relations. However, some predicates generated concerned with itemization and reference were meaningless, because predicates converted from words and phrases, such as "the items below," "Article 5," and so on are not intrinsic to a logical representation of the sentence. These words should be replaced with appropriate phrases before the process of translation. Since Japanese legal documents have strict rules concerning its description and modification, we succeeded to extract conventional expressions in the documents by some regular expressions (Kimura, Nakamura, and Shimazu 2009). In order to investigate whether the proper method depends on the language or the nation establishing laws, we try to apply our approach to the English and Vietnamese versions of Vietnamese Law and the United States Code. Therefore, our purpose in this paper is to show a difference of the method to generate independent, plain sentences from legal texts including itemization. This study is regarded as a derivation of the series of our main study to translate legal documents into logical forms (Nakamura et al. 2008). We expect that these fruitful results are able to be applied not only to the English and Vietnamese versions of the translation system into logical forms, but also to a support system for reading legal documents and a text-to-speech system.

In this paper, we introduce our current system and its problems in Section 2. In Section 3 we show analysis of law sentences including itemization or reference, and we propose a method to rewrite the law sentences into plain sentences in Section 4. We also examine our new method and report its results in Section 5. Finally, we conclude and describe our future work in Section 6.

2 The Current System and Problems

In this section, we describe our current system for translating legal documents into logical forms, and its problems. We call our system WILDCATS[1].

2.1 Wildcats

Acquisition of knowledge bases by automatically reading natural language texts has widely been studied. While the definition of semantic representation differs depending on what the language processing systems deal with, some systems try to generate logical formulae based on first order predicate logic (Hobbs, Stickel, Martin, and Edwards 1988; Mulkar, Hobbs, and Hovy 2007a; Mulkar, Hobbs, Hovy, Chalupsky, and Lin 2007b). Legal documents are suited for translating

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

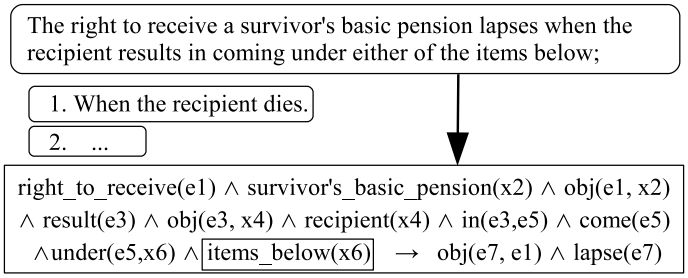

Fig. 1 Converting a law sentence including a reference phrase

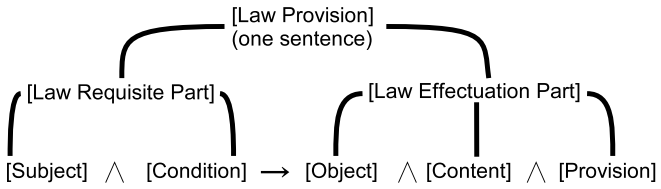

Fig. 2 Structure of requisition and effectuation

into logical representation, since they are different from daily-use texts in that they are described with characteristic expressions in order to avoid ambiguous description. Taking into account the linguistic analysis of the expressions, we can extract the logical structure of legal documents.

Our system, Wildcats, derives logical forms forms from law sentences (Nakamura et al. 2008). We explain an outline of our current system, showing an example of input and output in Fig. 1.

In most cases, a law sentence in Japanese Law consists of a law requisite part and a law effectuation part, which designate its legal logical structure (Tanaka, Kawazoe, and Narita 1993; Nagai, Nakamura, and Nomura 1995). The structure of a sentence in terms of these parts is shown in Fig. 2. The law requisite part is further divided into a subject part and a condition part, and the law effectuation part is divided into an object, content, and provision part.

Dividing a sentence into these two parts in the pre-processing stage makes the main procedure more efficient and accurate. Nagai et al. (Nagai et al. 1995) proposed an acquisition model for this structure from Japanese law sentences. Dealing with strict linguistic constraints of law sentences, their model succeeded in acquiring the structures at fairly high accuracy using a simple method, which specifies the surface forms of law sentences. Our approach is different from theirs in that we consider some semantic analyses in order to represent logical formulae.

The following list is the procedure for one sentence (See Fig. 3). We repeat it when we process

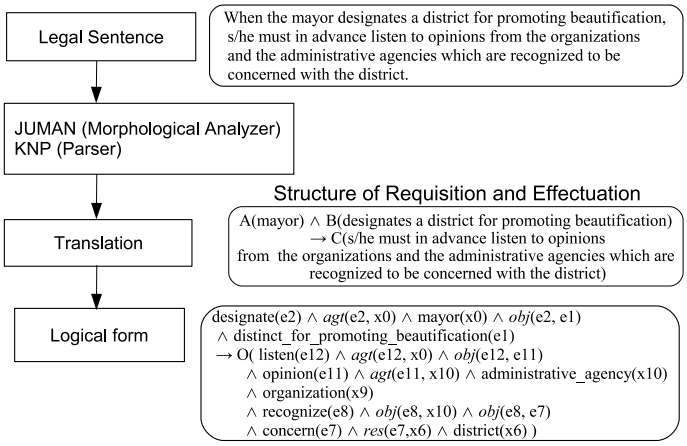

Fig. 3 Flow chart of Wildcats

a set of sentences.

(1) Analyzing morphology by JUMAN[2] and parsing a target sentence by KNP[3].

(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[4] 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. 2. 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.

[2]JUMAN is a morphological analyzer of Japanese developed by Kyoto University (Kurohashi, Nakamura, Matsumoto, and Nagao 1994). It segments sentences into morpheme sequences with many additional pieces of information such as a semantic category of each word, a part of speech, and so on, based on a hidden Markov model.

[3]KNP is a rule-based Japanese dependency analyzer developed by Kyoto University (Kurohashi and Nagao 1994).

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

2.2 Problems of Wildcats

When our system converts a law sentence including referential phrases, it is not interpreted correctly. For example, in Fig. 1, the enclosed predicate "items_below(x6)" is useless. Even if the post-processing system performs logical operations with the generated logical expressions, it does not result in line with our expectations. This is because the generated predicates lack information which must be referred. In case of Fig. 1, there is no connection between the predicate "items_below(x6)" and the ones in the following items.

One solution is to replace these phrases with appropriate ones in the items before the process of translation into logical forms. In other words, the front end system of Wildcats rewrites a sentence including itemization into a plain sentence. Therefore, substituting corresponding referent phrases for these expressions appropriately, our proposed system in this paper generates a meaningful text with high readability, and then the generated text can be input to the translation system. For example, the system should process the following instead of the input sentences in Fig. 1; "The right to receive a survivor's basic pension lapses when the recipient dies." As long as treating meaningful sentences like this, the system does not generate any more redundant predicate such as "items_below(x6)." In this paper, we propose a pre-processing system which modifies input sentences including itemization.

2.3 Scope of Our Study

The scope of the study in this paper is restricted to sentences including itemized expressions written in Japanese, Vietnamese, and English. In the preceding study (Kimura et al. 2009), focusing on Japanese legal sentences, we showed that the system worked well using some simple regular expressions. In this paper, we apply it to the Vietnamese Law on Enterprises and the United States Code: Title 39-Postal Service.

3 Analysis of Law Sentences with Itemization

In general, a notation of law sentences is strictly affected not only by the written language, but also by the nation establishing the law. In other words, it depends on the legislative process whether or not our simple approach is useful for other countries' law. In this section, we analyze law sentences including itemization from the following two aspects; One is linguistic characteristics, and the other is comparison of legislative proceedings among nations.

3.1 Characteristics of Law Documents among Nations

The legislation system of Japanese Law is rational to keep the notation of expressions of law[5]. A bill is basically proposed by the proper authority of the law. Once the authority has made a draft of the bill, it negotiates with other authorities. After that, the cabinet strictly examines the draft in terms of inconsistency with other laws, expressions, formats and so on, using the database of legislation. As a result, this system keeps even the usage of comma and period.

Not all other countries have the system similar to Japan. In the United Kingdom, the description check by the legislature is not as strict as Japan, since in most cases the draft of a bill is prepared by an outsider of the ministry. In the United States of America, there is no organization or system for consolidating expressions of laws. In Asian countries except Japan and Korea, each ministry independently makes out a draft of a bill without coordinating various opinions from other ministries. As a result, the notation of bills becomes different among ministries. Moreover, in some countries bills are often modified during deliberation in the national assembly, while bills mostly pass the National Diet in Japan as drafted. This political process causes inconsistencies in notation.

Vietnamese laws are also strictly examined by a number of organizations concerning the laws before passing the National Diet (Endo 2007). Particularly, there is a rigorous inspection about interpretation of laws by the standing committee in Parliament, though it is unknown whether the notation of expressions is surveyed by some authorities as strict as Japan. In fact, in order to reform the legislative structure, the Vietnamese government enacted the Law on Promulgation of the Legal Documents[6] in 2008 (Endo 2008). This law is considered as an evidence that laws have carelessly been proposed in a number of independent organizations so far. Thus, it seems difficult to keep a writing notation in administrative documents in Vietnam. This is a difference between Japan and Vietnam from a point of view of keeping the notation.

3.2 Definition of Itemization

In general, a law consists of a number of articles, each of which is further subdivided into a number of paragraphs or items. Both articles, paragraphs, and items have sequential numbers with a typeface different from each other. For example, in the English version of Vietnamese Law, articles start with "Article 1," "Article 2" and so on, paragraphs with "$1., 2., \ldots$," and items with "$(a), (b), \ldots$." Although there are a few differences in notation between English and Vietnamese,

[5]This section is written based on the discussion with Prof. Matsuura in Graduate School of Law, Nagoya University. For more detail about the administrative structure of legislation of Japanese Law, see Nagano (Nagano 2005).

[6]http://vietlaw.gov.vn/LAWNET/docView.do?docid=22443&type=html (in Vietnamese)

Article 21 Contents of requests for business registration

(1) Name of the enterprise.

(2) Address of the head office of the enterprise; telephone number, facsimile number, email transaction address (if any).

(3) ....

Fig. 4 Article 21 in the Vietnamese Law on Enterprises

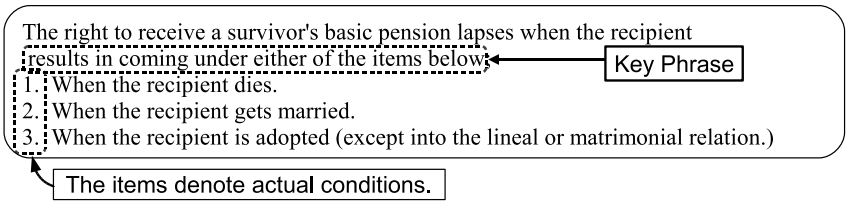

Fig. 5 Itemization of conditions in the law requisite part

itemization structure can be dealt with easily by pre-processing.

In the case of Japanese Law, we basically recognize an itemized expression as a noun phrase or a subordinate clause following the upper paragraph or article, which consists of sentences. An example of itemized expression is shown in Fig. 1. On the contrary, in the case of Vietnamese Law, even some articles are expressed as a phrase which lacks the subject or the main verb. We show an example in Fig. 4. Taking it into consideration, we define itemization, with which we deal in this study, as a phrase or a sentence following a sentence in an article or a paragraph. The article shown in Fig. 4 is not recognized as itemization, because it does not have a complete sentence.

3.3 Analysis of Itemization in Japanese Law Sentences

Some law sentences include itemization of conditions in the law requisite part, an example of which is shown in Fig. 5. The enclosed phrase should be replaced with one of the items denoting actual conditions. When one or more conditions are satisfied, the description in the law effectuation part becomes effective. We found 34 sentences of such a style in National Pension Law. Therefore, we considered a method to embed itemized conditions instead of cue phrases of itemization.

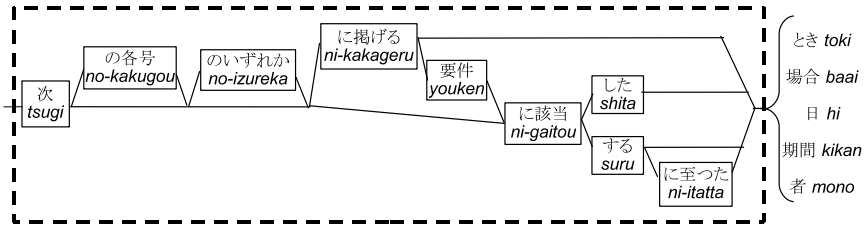

Fig. 6 Key phrases for itemization

Table 1 Frequency of Key Phrases

(**KP**: Key Phrase)

Format of KPs / Frequency			
KP + *toki*	(とき)	when	9
KP + *baai*	(場合)	case	9
KP + *mono*	(者)	person	6
KP + *hi*	(日)	day	3
KP + *kikan*	(期間)	period	1
KP + *youken*	(要件)	requirement	1
KP + a noun			5
Total			34

Table 2 Frequency of Condition Items

CI: Condition Items

Format of CIs / Frequency			
CI + *toki*	(とき)	when	106
CI + *koto*	(こと)	matter	4
CI + *mono*	(もの)	thing	3
CI + *mono*	(者)	person	2
CI + a noun			9
Total			124

We defined *Key Phrases*, which always appear in sentences before an itemization[7]. As we analyzed sentences from all 215 articles of the National Pension Law, the set of Key Phrases can be expressed as a regular expression, the diagram of which is shown in Fig. 6. For example, the phrase *"Tsugi no kaku gou ni gaitou suru ni itatta,"* meaning "to result in coming under either of the items below[8]," which is derived from the generative rule in Fig. 6, is regarded as a Key Phrase.

Itemized condition sentences appear next to sentences which contain Key Phrases. The last words of these sentences are "*Toki* (time)," "*Mono* (person)," and so on. In this paper, we call these sentences excluding the last words *Condition Items*. Key Phrases and Condition Items appearing in National Pension Law are shown in Table 1 and Table 2, respectively.

We will describe a method to remove itemization using Key Phrases and Condition Items in

[7]There may be a proviso between the sentence and itemization.

[8]If we do not care about word-to-word translation for the Japanese law sentence, the following phrase is more appropriate; "to be included in one of the following cases."

Section 4.

3.4 Analyses of Itemization in Vietnamese Law Sentences Written in Vietnamese and English

In order to find *Key Phrases* and *Condition Items*, we analyzed 100 out of 172 articles in the Vietnamese Law on Enterprises. Although all Key Phrases identify one regular expression in Japanese Law, we defined 15 and 14 rules of regular expression for Vietnamese and English, respectively. This means there are a variety of expressions denoting itemization in Vietnamese Law. We show the set of rules for the English and Vietnamese versions of Vietnamese Law in Fig. 7 and Fig. 8, respectively. Since we manually made the sets of rules in English and Vietnamese separately, each rule in the English version does not correspond to that of Vietnamese with the same label, and vice versa.

In the English version of the law, we need to deal with inflection of words. The number of rules would be reduced, if we did not consider an irregular conjugation for some particular nouns. For example, Rule 9 accepts the phrase "following rights," "following obligations," or "following undertakings" and generates an appropriate phrase with the condition item, omitting the word "following" and the suffix '-s.' Because some irregular conjugations such as 'duties' are not accepted by Rule 10, an additional rule is added to the rule-set. Rule 10 works the same as Rule 9 except replacing the suffix '-ies' to '-y.' Moreover, each regular expression accepts a number of Key Phrases corresponding to a Condition Item. In other words, some rules which could be merged with other rules are separated due to the different Condition Items.

Similar to Japanese, Vietnamese does not distinguish singular or plural nouns by inflection. Vietnamese distinguish singular and plural nouns by quantifiers which precede corresponding nouns, such as 'các (all),' 'những (some),' 'tất cả (every),' 'một (one),' 'hai (two),' and so on. The rules of Vietnamese are simpler than that of English, being not necessary for the process of inflection. Some rules similar to each other are distinguished depending on the Condition Items corresponding to the rule.

3.5 Analysis of Itemization in the United States Code

We analyzed Postal Service of the United States Code. As a result, we defined 4 rules of regular expression, shown in Fig. 9.

Rule 1 in Fig. 9 covers most of the items, since the main clause becomes a complete sentence, replacing a hyphen ('-') at the last of the clause with each item. Figure 10 shows an example of itemization. Therefore, we rarely took care of inflection for the rule extraction. This simple

```
1  ^(.*)\s+(the following terms shall be construed as follows:)$
2  as follows:
3  (at least the two following elements|in one of the following cases):
4  (enterprise|except) in the following cases:
5  ((in) (one|any) of|(with)|(in)) the following (manner|case|provision)s[:;]
6  (all of )?(t|T)he following (condition|case)s(.*[^:;\.])?([:;\.])$
7  following rights and duties:
8  following criteria and conditions:
9  following (right|obligation|undertaking)s:
10 following duties:
11 (one of |either of )?the (two )?following (act|manner)s(.*):
12 following attached (.+):
13 following ((main [a-zA-Z]+)|([a-zA-Z]+particulars)|
             ([a-zA-Z]+((,|\s+or|\s+and)\s+[a-zA-Z]+)*)(\s+.+)?):$
14 (in which|by way of|the right)s?:$
```

Fig. 7 Key Phrases for the English version of Vietnamese Law

```
1  (khi|nếu) (có|thuộc) (một trong|đủ) (những|các) (trường hợp|điều kiện) sau đây
2  trong (những|các) (trường hợp) sau đây
3  (phải) (.+) các tiêu chuẩn và điều kiện sau đây:
4  các (báo cáo|báo cáo và tài liệu|hoạt động|tài liệu) sau đây:
5  các (nội dung|vấn đề) sau đây
6  các nội dung.*:
7  có các (.+) sau đây:
8  (gồm)(.*):
9  Tổ chức, cá nhân sau đây
10 (có ít nhất) (.+) thành tố sau đây:
11 (theo|bằng) (các|một trong các|một trong hai) (.+) sau đây:
12 theo (quy định|nguyên tắc) sau đây:
13 (trong đó):
14 để thực hiện (một trong các|các) hành vi sau đây:
15 (Các|các|Những|những) (.+) sau đây
```

Fig. 8 Key Phrases for Vietnamese Law

```
1  ^((.*)[^\.]) -\s*$
2  ^(.*) following (.*)s(, among others):
3  ^(.*) following (.*)s:
4  ^\s*[Tt]he following provisions (.*):
```

Fig. 9 Key Phrases for the United States Code

expression seems to be a result of study for high readability, although it differs from the English version of the Vietnamese Law.

4 Method for Removing Itemization

In Section 3.3, we defined Key Phrases as cue phrases that always appear with itemization, like "*tsugi-no kaku gou no izureka ni gaitou-suru* ((something) to which either of the following items

Sec. 202. Board of Governors in Title 39 - Postal Service
(e)(2) The Inspector General shall be appointed -
(A) for a term of 7 years;
(B) without regard to political affiliation; and
(C) solely on the basis of integrity and demonstrated ability in accounting, auditing, financial analysis, law, management analysis, public administration, or investigations.

Fig. 10 Example of itemization in the United States Code

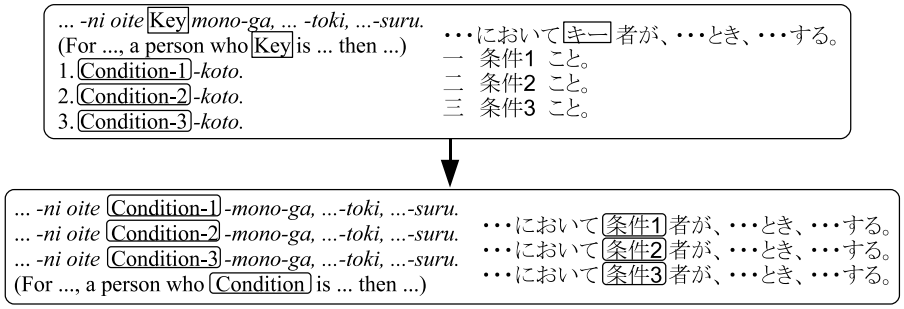

Fig. 11 Removing itemization

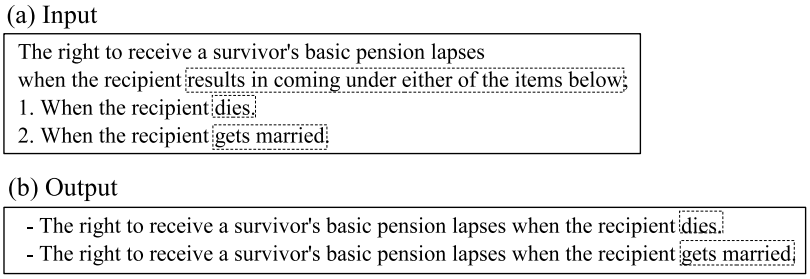

Fig. 12 An example of removing itemization

is applicable)," and we search for itemization with it. If a Key Phrase is found, we regard the following items as Condition Items, and replace the Key Phrase with one of the Condition Items for each. Then we have sentences which are understandable separately, as shown in Fig. 11. We show an example of the pair of input and output in Fig. 12.

The process of Vietnamese Law is different from that of Japanese in that there are a number of rules of regular expression. Since some rules conflict with other rules, priority is established in the order of the rule number. Each rule has a corresponding Condition Item, which is defined as regular expression. We show an example of rewriting itemized expressions in Fig. 13, in which

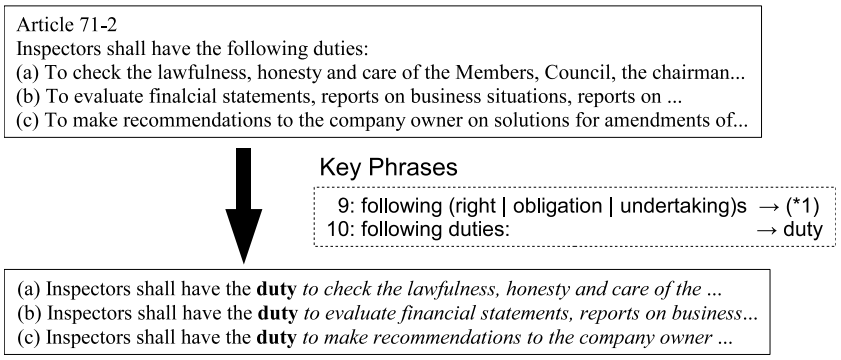

Fig. 13 An example of removing itemization (Article 71-2, the Vietnamese Law on Enterprises)

the dashed box labeled 'Key Phrases' denotes a part of rewriting rules. Rule 9 rewrites a phrase matching the regular expression in the left hand side to one of the words in the bracket. In this case, Rule 10 matches the phrase 'following duties' in the article, which is replaced to the word 'duty' described in the right hand side in Rule 10.

5 Experiments and Results

We tested our system on itemization. The test set for each language is shown in Table 3. Since we extract Key Phrases of Japanese from National Pension Law, we used it for a closed test. For an open test we used Income Tax Law as the test set.

In these experiments, it is difficult to establish a baseline due to the distinctiveness of our model and its target. Some studies which extract web contents from HTML or XML documents (Liu, Grossman, and Zhai 2003) may be able to deal with itemization in HTML or XML documents. However, our method is different from them in that it includes process to find itemized phrases without a tag, and to make plain sentences. We examine whether or not our model works well to the law documents in some languages, regardless of the linguistic characteristics, or of its nation which established the law.

We extract Key Phrases of both Vietnamese and English from 100 out of 172 articles in the Vietnamese Law on Enterprises. Therefore, we use sentences from Article 1 to 100 for a closed test and from Article 101 to 172 for an open test. Hereafter, we call the open test *vopen1*. In addition, we examine the Law on Bankruptcy for another open test, called *vopen2*. We have two kinds of experiments; one is to examine whether the system successfully identify articles including itemization, and the other is to measure the accuracy of removing itemized expressions that the

Table 3 Input texts for open and closed tests

	Test	**#item**	**Test Set**
Japanese	closed	124	National Pension Law
	open	548	Income Tax Law
Vietnamese	closed	354	Article 1-100 in Law on Enterprises
	open 1	275	Article 101-172 in Law on Enterprises
	open 2	165	Article 1-98 in Law on Bankruptcy
Vietnamese (English)	closed	354	Article 1-100 in Law on Enterprises
	open 1	275	Article 101-172 in Law on Enterprises
	open 2	165	Article 1-98 in Law on Bankruptcy
the US Code	closed	141	Part I of Postal Service
	open	154	Part I and II of Public Contracts

Table 4 Experimental results for identifying itemization

	Test	**#Art**	**Find**	**Over**	**Err**	**P**	**R**
Japanese	closed	33	33	1	1	97.1%	97.1%
	open	147	133	15	1	99.2%	89.8%
Vietnamese	closed	70	72	0	2	97.2%	**100%**
	open 1	57	46	11	0	**100%**	80.7%
	open 2	36	34	4	2	94.1%	88.9%
Vietnamese (English)	closed	70	73	0	3	95.9%	**100%**
	open 1	57	52	5	0	**100%**	91.2%
	open 2	36	9	28	1	88.9%	22.2%
the US Code	closed	43	43	0	0	**100%**	**100%**
	open	42	26	16	0	**100%**	62.0%

#Art: the number of articles including itemization, **Find**: Found,
Over: Oversight, **Err**: Error, **P**: Precision, **R**: Recall

system successfully found.

For the United States Code, Key Phrases are extracted from Postal Service. For an open test we used the US Code-Title41:Public Contracts as the test set.

Firstly, we show the experimental result for identifying itemization by key phrases in Table 4. The labels '#Art,' 'Find,' 'Over,' 'Err,' 'P,' and 'R' denote 'the number of articles including item-

Table 5 Experimental results for removing itemization

	Test	**#item**	**Succ**	**Err**	**Acc**
Japanese	closed	119	87	32	73.1%
	open	426	219	207	51.4%
Vietnamese	closed	333	309	24	92.8%
	open 1	238	218	20	**91.6%**
	open 2	131	94	37	71.7%
Vietnamese (English)	closed	333	315	18	94.6%
	open 1	261	191	70	73.2%
	open 2	22	15	7	68.2%
the US Code	closed	146	146	0	**100.0%**
	open	95	84	11	88.4%

#item: the number of items to be processed,

Succ: Succeeded, **Err**: Error, **Acc**: Accuracy

Article 5-1, the Law on Bankruptcy The bankruptcy procedures applicable to enterprises and cooperatives which fall into the state of bankruptcy shall *include:*
(a) The submission of applications for, and opening of bankruptcy procedures;
(b) The restoration of business operation;
(c) The liquidation of properties, debts;
(d) The declaration of bankruptcy of enterprises, cooperatives.

Fig. 14 Example of failure in the English version of the Law on Bankruptcy

ization,' 'the number of articles that the system identified as itemized expressions,' 'the number of oversights,' 'the number of errors,' 'precision' and 'recall,' respectively. The result shows that the system sufficiently found articles including itemization except *vopen2* in the English test. In the Law on Bankruptcy, expressions are quite different from those in the Law on Enterprises. We show an example of the Law on Bankruptcy in Fig. 14. In this case, the complete sentences can be generated by adding each item to the last of the main clause with some minor modifications to remove a colon (:) and replace a semi-colon (;) with a period (.). Although we did not extract any rule processing this expression from the Law on Enterprises, it is often used in the United States Code. The difference between the Law on Enterprises and the Law on Bankruptcy in notation may be caused by different interpreters.

Secondly, the experimental results for removing itemization are shown in Table 5. The labels '#item,' 'Succ,' 'Err' and 'Acc' denote 'the number of items to be processed,' 'the number of

Paragraph 2, Article 27-3, National Pension Law *In the case of the following items,* the revision of the rate after the base year is fixed on the basis of the rate on the item, regardless of the provisions stipulated in the preceding paragraph.

(1) The price rate exceeds the nominal net wage rate, and the nominal net wage rate exceeds 1 the nominal net wage rate

(2) The price rate exceeds 1, and the nominal net wage rate falls below 1 $\frac{1}{}$

Fig. 15 Example of failure in Japanese National Pension Law

items that the system successfully processed,' 'the number of errors,' and 'accuracy,' respectively. In the closed test of Japanese, we found that 11 of the whole errors were items which denote a combination of a Condition Item and an object part in the law effectuation part. In other words, the objects of these sentences change depending on the Condition Items. An example is shown in Fig. 15. This article determines the revision of the rate after the base year about the national pension. An important thing here is that each item consists of a condition part and its result, separated with a space[9]. That is, the first Key Phrase denoting "In the case of the following items," which is emphasized corresponds to the first phrases of each item, while the second Key Phrase denoting "on the basis of the rate on the item" which is underlined corresponds to the second phrases of each item. Our system did not deal with this type of itemization.

For the result of open test with Income Tax Law, a little more than half of the sentences were processed well. There seems to be some difference in notation between National Pension Law and Income Tax Law. Particularly, we found the increase of itemization consisting of a combination of a Condition Item and an object part to 84. Results will be improved after an analysis of the mistakes.

The results of both English and Vietnamese show higher accuracy than that of Japanese in terms of removing itemized expressions. This is because the number of rules of regular expression is increased to 14 and 15, while there is only one rule for Japanese. In the English test, we found that some Key Phrases were followed by a number of types of Condition Items different from each other, so that the set of rules did not cover all the Key Phrases even in the closed test. Because this decision becomes much more difficult in the open tests, the accuracies in the open tests come down to 73.2% for *vopen1* and 68.2% for *vopen2*. We show an example of failure which occurred in *vopen1* in Fig. 16. There are two rules which deal with the phrase "in the following cases" in Rule 4 and 5 in Fig. 7, which rewrite it to the corresponding phrases "in the case of" and "in the case that," respectively. Figure 16 shows that the key phrase was replaced to the phrase "in the case that" although the following item is a noun phrase.

[9]In Japanese writing, no spaces are left between words. Since there is a space only between the condition part and the result in the item, they are absolutely identified.

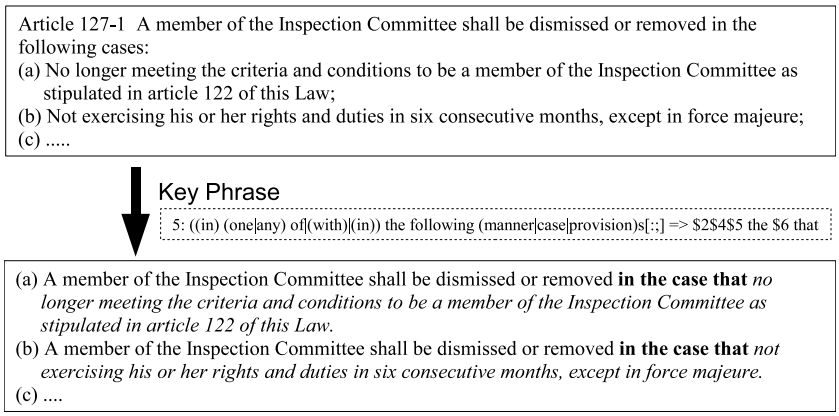

Fig. 16 An example of failure (Article 127-1, the Vietnamese Law on Enterprises, English)

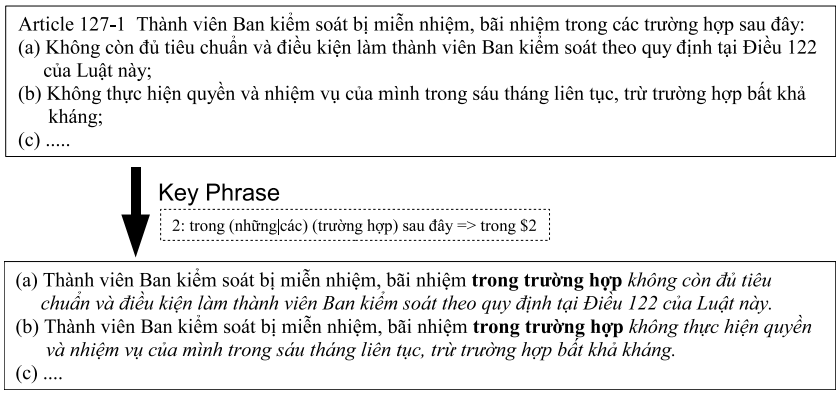

Fig. 17 An example of success (Article 127-1, the Vietnamese Law on Enterprises, Vietnamese)

In the case of Vietnamese law text, we also found some errors that the meaning of sentences generated are different from original meaning, and ungrammatical sentences may be generated. However, the accuracy keeps high even in the open test. This is because the Vietnamese grammar is not as strict as English in terms of distinction between a phrase and a clause. Figure 17 shows that the key phrase is successfully replaced in the itemized expression where it failed in the English version.

In the case of the US Code, the accuracy decreased in the open test. The difference in notation of itemization between the titles affects not only the result on identifying itemization, but also the one on removing itemization. This problem can be overcome by adding extra rules to the Key Phrase.

Overall accuracy would be improved depending on the rule set of regular expression. Therefore, we can conclude that our method is quite suitable not only for Japanese legal texts but also for other languages with some modification.

6 Conclusion

In this paper we proposed a method to rewrite legal sentences including itemization into independent, plain sentences, focusing on laws written in three languages. From the linguistic analyses, we showed the difference of the number of regular expressions for extracting Key Phrases between Japanese, Vietnamese and English. It implies that fixed expressions are often used in Japanese Law. In Vietnamese law documents, there are some common words and phrases which appear with high frequency at the Key Phrases. In the United States Code, most of itemized forms are identified with a hyphen at the last of the main clause. Further investigation of Vietnamese words and phrases is required for making accurate regular expression rules.

In the experiments, we showed that the system successfully extracted itemized expressions with some exceptions. We consider that the system is useful not only for the front end of our main system, Wildcats, but also for assistance in reading legal documents. We can improve this system by introducing a method for enhancing readability of the output sentences. Concerning the Vietnamese version of translation system (Wildcats), we need to wait for the development of a dependency parser for Vietnamese.

Acknowledgment

We would like to give special thanks to Prof. Yoshiharu Matsuura in Nagoya University for discussion about the differences among nations in notation of law documents. This research was partly supported by the 21st Century COE Program 'Verifiable and Evolvable e-Society' and Grant-in-Aid for Scientific Research (19650028 and 20300057).

Reference

Endo, S. (2007). "The national assembly and legislative process in Vietnam (in Japanese)." *Gaikoku no rippou (Foreign legislation)*, **231**, pp. 110–151.

Endo, S. (2008). "Legal structure reform in Vietnam: Law on Promulgation of the Legal Documents 2008 (in Japanese)." *Gaikoku no rippou (Foreign legislation)*, **238**, pp. 177–190.

Hagiwara, S. and Tojo, S. (2006a). "Discordance Detection in Regional Ordinance: Ontology-based Validation." In *Legal Knowledge and Information Systems: JURIX 2006: The Nineteenth Annual Conference (Frontiers in Artificial Intelligence and Applications)*, pp. 111–120. IOS Press.

Hagiwara, S. and Tojo, S. (2006b). "Stable legal knowledge with regard to contradictory arguments." In *AIA'06: Proceedings of the 24th IASTED international conference on Artificial intelligence and applications*, pp. 323–328 Anaheim, CA, USA. ACTA Press.

Hobbs, J. R., Stickel, M., Martin, P., and Edwards, D. (1988). "Interpretation as abduction." In *Proceedings of the 26th annual meeting on Association for Computational Linguistics*, pp. 95–103 Morristown, NJ, USA. Association for Computational Linguistics.

Katayama, T. (2005). "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)." *IPSJ (Information Processing Society of Japan) Journal*, **46** (5), pp. 515–521.

Katayama, T. (2007). "Legal Engineering – An Engineering Approach to Laws in e-Society Age –." In *Proc. of the 1st Intl. Workshop on JURISIN*.

Katayama, T., Shimazu, A., Tojo, S., Futatsugi, K., and Ochimizu, K. (2008). "e-Society and Legal Engineering (in Japanese)." *Journal of the Japanese Society for Artificial Intelligence*, **23** (4), pp. 529–536.

Kimura, Y., Nakamura, M., and Shimazu, A. (2009). "Treatment of Legal Sentences Including Itemized and Referential Expressions –Towards Translation into Logical Forms–." In Hattori, H., Kawamura, T., Ide, T., Yokoo, M., and Murakami, Y. (Eds.), *New Frontiers in Artificial Intelligence: JSAI2008 Conference and Workshops, Asahikawa, Japan, June 11-13, 2008, Revised Selected Papers*, Vol. 5447 of *Lecture Notes in Artificial Intelligence*, pp. 242–253. Springer.

Kurohashi, S. and Nagao, M. (1994). "KN Parser : Japanese Dependency/Case Structure Analyzer." In *Proceedings of the Workshop on Sharable Natural Language Resources*, pp. 48–55.

Kurohashi, S., Nakamura, T., Matsumoto, Y., and Nagao, M. (1994). "Improvements of Japanese Morphological Analyzer JUMAN." In *Proceedings of the Workshop on Sharable Natural Language Resources*, pp. 22–28.

Liu, B., Grossman, R., and Zhai, Y. (2003). "Mining Data Records in Web Pages." In *Proc. of the ninth ACM SIGKDD*, pp. 601–606.

Mulkar, R., Hobbs, J. R., and Hovy, E. (2007a). "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.

Mulkar, R., Hobbs, J. R., Hovy, E., Chalupsky, H., and Lin, C.-Y. (2007b). "Learning by Reading: Two Experiments." In *Proceedings of IJCAI 2007 workshop on Knowledge and Reasoning for Answering Questions.*

Nagai, H., Nakamura, T., and Nomura, H. (1995). "Skeleton Structure Acquisition of Japanese Law Sentences based on Linguistic Characteristics." In *Proc. of NLPRS'95, Vol 1*, pp. 143–148.

Nagano, H. (2005). "Foundation and Common sence for legislation (in Japanese)." *Jichitai Houmu Kenkyuu*, **winter**.

Nakamura, M., Nobuoka, S., and Shimazu, A. (2008). "Towards Translation of Legal Sentences into Logical Forms." In Satoh, K., Inokuchi, A., Nagao, K., and Kawamura, T. (Eds.), *New Frontiers in Artificial Intelligence: JSAI 2007 Conference and Workshops, Miyazaki, Japan, June 18-22, 2007, Revised Selected Papers*, Vol. 4914 of *Lecture Notes in Artificial Intelligence*, pp. 349–362. Springer.

Tanaka, K., Kawazoe, I., and Narita, H. (1993). "Standard Structure of Legal Provisions - For The Legal Knowledge Processing by Natural Language - (In Japanese)." In *IPSJ Research Report on Natural Language Processing*, pp. 79–86.

Makoto Nakamura: received the Bachelor degree in Information Engineering from Kyushu Institute of Technology in 1995. He received the Master and Doctoral degrees in Information Science from JAIST in 1997 and 2004, respectively. He is now an assistant professor at School of Information Science, JAIST. His research interests include Legal Text Processing, and Simulation of Language Evolution.

Yusuke Kimura: received the Bachelor degree in Engineering from Osaka University in 2005. He received the Master degree in Information Science from JAIST in 2008. He now works at Pixela Corporation as a software engineer.

Minh Quang Nhat Pham: is now a Ph.D student at Natural Language Processing Laboratory, School of Information Science, Japan Advanced Institute of Science and Technology (JAIST). He received the B.S degree of Information Technology from Vietnam National University of Hanoi (VNUH) in 2006, and received M.S degree of Information Science from JAIST in 2010. His research interests include Machine Learning, Text Generation, and Legal Text

Processing.

Le Minh Nguyen: received the BS degree in information technology from Hanoi University of Science, and the MS degree in information technology from Vietnam National University, Hanoi in 1998 and 2001, respectively. He received the Ph.D in information science from Graduate School of Information Science, Japan Advanced Institute of Science and Technology (JAIST) in 2004. He is now an assistant professor at Graduate School of Information Science, JAIST. His research interests include text summarization, machine translation, natural language processing, machine learning, and information retrieval.

Akira Shimazu: received the Bachelor and Master degrees in mathematics from Kyushu University in 1971 and 1973, respectively, and the Doctoral degree in Natural Language Processing from Kyushu University in 1991. From 1973 to 1997, he worked at Musashino Electrical Communication Laboratories of Nippon Telegram and Telephone Public Corporation, and at Basic Research Laboratories of Nippon Telegraph and Telephone Corporation. From 2002 to 2004, he was the president of the Association for Natural Language Processing. He has been a professor in the Graduate school of Information Science, JAIST since 1997.

(Received May 15, 2009)
(Revised Dec. 9, 2009)
(Accepted Dec. 16, 2009)