Study on extracting conceptual structures from legal texts

Pham Anh Cam, Minh Le Nguyen, Akira Shimazu
School of Information Science,
Japan Advanced Institute of Science and Technology

Abstract: To understand and to check the correctness and the consistence of a law is very important but normally very difficult because of long and difficult texts. Our research aims to take a hand in this work by modeling Japanese laws. Such laws can be seen as specifications of social systems. Our purpose is to clarify a method for extracting a model (conceptual model) of social system which such laws represent. We first create the typed dependency representations of Japanese law texts and then transform these dependencies into UML classes. Our model will be used to develop information systems based on laws.
Keywords: Japanese pension law, conceptual model, UML class

1. Introduction

The purpose of our research is to clarify a method for extracting a conceptual model of a social system which Japanese laws represent.

Our research is conducted as the study of legal engineering, which aims to exam and verify whether a law has been established appropriately according to its purpose, whether the law is consistent with related laws, and whether the law has been modified, added, and deleted consistently. There are two important goals of legal engineering which are to help experts make complete and consistent laws; and to design an information system which works based on laws [1][2].

To model the social system represented by Japanese laws, we use a conceptual model. The models have giant power in helping us understand many phenomena easily and clearly. People process information every time. This processing turns out to be a conceptual model of how things in our surrounding environment work. In the most general sense, a model is anything used in any way to represent anything else. The term conceptual model may be used to refer to models which are represented by concepts or related concepts which are formed after a conceptualization process in the mind.

In order to express the result of conceptual modeling, we need a modeling language. A modeling language is any artificial language that can be used to express information or knowledge or systems in a structure that is defined by a consistent set of rules. A modeling language can be graphical or textual. There are several elaborate graphical notations that could be used for conceptual modeling, ranging from those used in the past for Entity-Relationship Model [ERM 1976], conceptual graphs [Sowa 1982] to those presently used for object modeling [UML 1997].

Our model is built by the following way: the common nouns are extracted to become the classes. These classes have attributes, methods and relations extracted correspondingly from the dependencies of legal terms like in table 1. These classes then are represented textually.

POS Tagging	UML class component
Noun	Class
Verb	Method of Class
Modifier Noun, Verb Adjective	Attribute of Class
Relation between nouns	Relation between classes

Table 1: Mapping from POS tagging to UML class components.

Our research contribution is that our model can be used as a tool to understand the law easily. Instead of reading a difficult law, people can take a look to the model to find out the related terms, actions as well as the relations between the objectives of the law – legal terms. Moreover, the relations between legal objectives are modeled; the logic of the law is also clearly expressed so that we can check the correctness and consistence of a law easily. We can also design the information systems based on laws using our conceptual model.

The remaining paper is structured into the following sections: Section 2 describes background and the related work. Section 3 illustrates the proposed method. Section 4 presents the evaluation of our model. Section 5 is the conclusion. Final is the acknowledgements.

2. Background

To represent relations between natural language texts, some approaches were proposed. To transform natural language to UML model, there are some proposed approaches.

The first one is the transformation of text to SBVR (Semantic of Business Vocabulary and Rules) and then from SBVR to UML. Mathias Kleiner et al [3] and Imran Sarwar Bajwa et al [4] are the typical researchers of this approach. Their models can be summarized in table 2. The second one was proposed by Narayan Debnath [5]. The author proposed five transformation rules for defining the UML class diagram from ATL model:

Natural language	SBVR metamodel	UML metamodel
Common nouns	Object Types	Class
Proper nouns	Individual Concepts	Object
Auxiliary + Action verbs	Verb Concepts	Class Method
Noun + Verb / Noun + Verb + Noun / Associative, Pragmatic Relations	Unary / Binary / Associative Fact Types	Association
"Is-property-of", Possessed nouns	Characteristic	Class Attribute
Indefinite articles, plural nouns, cardinal numbers	Quantifications	Cardinalities
"is-part-of", "included-in", "belong-to"	Partitive Fact Type	Generalization
"is-category-of", "is-typeof","is-kind-of"	Categorization Fact Types	Aggregation

Table 2: The mapping from Natural language to SBVR and UML metamodels [3][4]

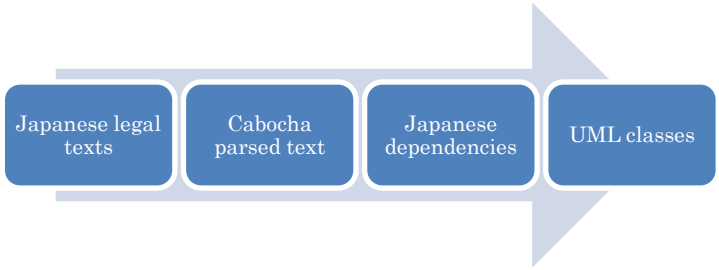

Figure 1: Conceptual model extraction system

Transformation Subject to Class; Object to Class; Subject Behavioral Response to Method; Subject Information to Method Parameter; and Language Extended Lexicon Relationships to Class Relationships.

Different from these above methods, our method is used for Japanese language and instead of using a semi-formal language as a medium; we directly extract UML classes from dependencies of Japanese legal terms which are extracted based on Cabocha parse output. Because of a huge part of extracted classes, we don't represent classes graphically but textually. Our system is presented in Figure 1.

3. Method

3.1. Cabocha parsing

We started the experiment from parsing legal text using Cabocha [7]. CaboCha is a Japanese dependency parser based on Support Vector Machines. To our best knowledge, Cabocha is the most accurate statistical Japanese dependency parser system (it reaches to 89.29% accuracy). In addition, it is a Cascaded Chunking Model using a non-backtracking definitive analysis algorithm so that the parse can be performed relatively efficient.

We will use token information: token base, pos, ctype in the next phase to extract the dependencies between tokens.

3.2. Dependency extraction

According to the representation of Stanford parser [8], we extracted five types of dependencies: dep(noun1_pos, noun2_pos, verb), dep(noun_pos, verb), no(noun1_pos, noun2), no(noun_pos, verb) and no(noun_pos, adjective). Nouns, verbs and adjectives in the sentences are extracted in phrases. For example, "労働保険", "開催された", "必要な" phrases are extracted. The POS tagging of the first token in the noun phrase is extracted as the POS tagging of this phrase. The POS tagging of the verb/adjective token in the verb/adjective phrase is extracted as the POS tagging of this phrase.

The dep dependency shows the relation between nouns through a verb. In the sentence which has two or three noun components: subject, direct object or indirect object, we extract the dependency dep(noun1, noun2, verb) in which the nouns are the corresponding components. In the sentence which only has only one noun component (either subject or direct object or indirect object), we extract the dependency dep(noun, verb).

The no dependency is the relation between a noun and its characteristic. The characteristic may be a modifier noun, verb or adjective.

i) dep(noun1__pos, noun2__pos, verb)

This dependency is extracted by the following way: the nouns are either subject, direct object or indirect

Dependency	Class	Attribute	Method	Relation
dep(noun1, noun2, verb)	noun1, noun2		verb	noun1 verb noun2
dep(noun, verb)	noun		verb	noun verb noun
no(noun1, noun2)	noun1, noun2	noun2		
no(noun, verb)	noun	verb		
no(noun, adjective)	noun	adjective		

Table 3: UML transformation

object and thus in general, we have 3 dependencies: dep(subj__pos, dobj__pos, verb), dep(subj__pos, iobj__pos, verb), dep(dobj__pos, iobj__pos, verb). For example, from the sentence "事業者は労働者に給料を払う。" we extract three dependencies: dep(事業者__名詞-一般, 労働者__名詞-サ変接続*, 払う), dep(事業者__名詞-一般, 給料__名詞-一般, 払う) and dep(労働者__名詞-サ変接続, 給料__名詞-一般, 払う). We used pos of the first word for compound nouns, considering the use of the word meaning.

ii) dep(noun__pos, verb)

Depend on the type of the noun, we extract one of three dependencies: dep(subj__pos, verb), dep(dobj__pos, verb) and dep(iobj__pos, verb). For example, from the sentence "労働者は働く。", we have the dependency: dep(労働者__名詞-サ変接続, 働く)

iii) no(noun1__pos, noun2)

This dependency is extracted from the following structures: "noun1 は… noun2 です/である", "noun1 <space>…noun2 をいう" . For example, from the sentence: "事業者　事業を行う者で、労働者を使用するものをいう。", we extract the dependency no(事業者__名詞-一般, もの).

This dependency also can be extracted from "noun1 の noun2" structure. If before and after の、there are many continuous nouns separated by a comma or conjunction words, all the pairs are extracted. For example, from the structure: 工場又は事務所の輸送、建築物、機械器具, we can extract these following dependencies: no(工場__名詞-一般, 輸送), no(工場__名詞-一般, 建築物), no(工場__名詞-一般, 機械器具), no(事務所__名詞-一般, 輸送), no(事務所__名詞-一般, 建築物), no(事務所__名詞-一般, 機械器具).

iv) no(noun__pos, verb)

This dependency is extracted from the structure "verb noun". The verb is extracted in full form of core form along with its tense. For example, from 働いた労働者 phrase, we extract the dependency no (労働者__名詞-サ変接続, 働いた)

v) no(noun_pos, adjective)

This dependency is extracted from the structure "adjective noun" or "noun は/が adjective です/である". For example, from 必要な事項 phrase, we extract no(事項__名詞-一般, 必要な) dependency.

3.3. UML transformation

In this step, the extracted dependencies in above phases are transformed into UML classes. All noun phrases go with the pos is "名詞-一般" or "名詞-サ変接続" are transformed to classes. The dep dependencies are transformed to methods and relations of the class. The no dependencies are transformed to attributes of classes. The detail transformations are presented in Table 3.

3.4. Transformation result

Our model is represented in classes textually. Below is a sample of a class of 国民年金法 5. Class Name: 積立金

Attribute:
- 運用
- 管理
- 積立て
- 額
- 規定する
- 係る

Method:
- なる
- 留意し
- 行う
- 寄託する
- 預託する
- 積み立てなければならない

Relation:
- なる with class 一部
- 行う with class 効率的
- 寄託する with class 運用
- 預託する with class 財政融資資金
- 積み立てなければならない with class 基金

4. Evaluation

We evaluated our model by comparing the conceptual model built by human with our result. The most important advantage of our model is that our model can capture a huge part of common nouns with their characteristics and transform them to classes go along with attributes with the high accuracy. While human

work frequently fails to extract some classes or attributes, our model can capture a huge part of relations of phrases correctly because it is based on the basic structures of Japanese grammar. For example, with 国民年金法, human extracted legal terms in 7 levels in which legal terms in next level are the characteristics of legal terms in the prior level. While the human work gets 23 legal terms in level 1, 119 legal terms in level 2, 69 legal terms in level 3, 89 legal terms in level 4, 181 legal terms in level 5, 127 legal terms in level 6 and 1 legal term in level 7, in total, 247 legal terms are captured, our model capture 2069 nouns as classes with 3548 attributes, 3575 methods and 8648 relations. Though these terms and their characteristics and relations include useless words (for example ため, もの, こと…), the accuracy is relatively high because we only extracted these legal terms and relations between them from basic Japanese dependencies. We even can extract more legal terms with their characteristics and relations if we relax the dependency extraction by adding some new extraction rules. We applied our method to 108 Japanese laws [6] and get the good result as well.

5. Conclusion

In this study, we proposed the method to extract conceptual model of Japanese laws. Our model use Cabocha parsed information to extract the dependencies of legal terms and then transform them to UML classes with attributes, methods and relations. We presented the classes textually. The advantage of our research is the ability to capture all legal terms go along with their characteristics and other related terms. This conceptual model not only be able to be used to understand the law clearly but also to check the correctness and consistence of the law.

6. Acknowledgements

This research was partially supported by the Ministry of Education, Science, Sports and Culture, COE Research and Grant-in-Aid for Scientific Research (B).

References

[1] T. Katayama, 2005. *The current status of the art of the 21st COE programs in the information sciences field. Verifiable and evolvable e-society – realization of trustworthy e-society by computer science – (in Japanese)*. In IPSJ (Information Processing Society of Japan) Journal, 46(5), pp.515-521.

[2] T.Katayama, 2007. *Legal engineering – an engineering approach to laws in e-society age*. In Proceedings of the 1st International Workshop on JURISIN.

[3] Mathias Kleiner, Patrick Albert, and Jean Bézivin, 2009. *Parsing SBVR-Based Controlled Languages*, In Proceedings of the 12th International Conference on Model Driven Engineering Languages and Systems.

[4] Imran Sarwar Bajwa, Ali Samad and Shahzad Mumtaz. 2009. *Object Oriented Software Modeling Using NLP Based Knowledge Extraction*, European Journal of Scientific Research, Vol.35 No.1, ISSN 1450-216X, pp. 22-33.

[5] Narayan Debnath et al, 2011. *An ATL Transformation from Natural Language Requirements Models to Business Models of a MDA Project*, Security Workshop, 2011 11th International Conference on ITS Telecommunications.

[6] http://www.japaneselawtranslation.go.jp/

[7] http://www.code.google.com/p/cabocha/

[8] http://nlp.stanford.edu/software/lex-parser.shtml