Automatic Identification of Candidate Analysis Classes in Korean Language by Semantic Information Extraction Technique

Hyoungil Jeong[l], Jungyun Seo[l,2], Sooyong Park' and Soojin Park[a]

{[1]Department of Computer Science and Engineering, [2]lnterdisciplinary Program of Integrated Biotechnology, [3]Sogang Institute of Advanced Technology},
Sogang University, 121-742, Korea
{hijeong, seojy, sypark, psjdream} @sogang.ac.kr

Abstract. With the advancements in the tools that support software design, the task of developing design models and analysis models is being automated gradually. However, several tasks still need human instinct. A typical task is the identification of analysis classes. Parsing Korean is very difficult because of its flexible word order. Therefore, the rule-based approaches used for the analysis of a sentence structure cannot be adopted for the elicitation of semantics in Korean sentences. In this study, we adopted a statistical information extraction technique for the identification of candidate classes. To validate the feasibility of this technique, we applied it to the precision and recall of the candidate classes in a real environment. To the best of our knowledge, our work is the first work to report the numerical performances for the automatic identification of analysis classes, and also the first work to use the statistical methods, in a real environment.

Keywords: Automatic Analysis Class Identification, CRFs Classifier, Phrase Chunking, Natural Language Processing

1 Introduction

The object-oriented technology (OOT) is the most popular method used for the design and development of software. Identification of analysis classes is the most salient and difficult task, because it is not fully automated and offers the least possibility of automation of the tasks of OOT [1]. The analysis classes are semantics that should be managed by the system in the requirement documents based on natural language. And the task of identification of analysis classes is a process of abstraction. This task needs human intuition, and it is difficult to automate using predefined rules or patterns. This attribute is similar to many problems in natural language processing (NLP) that must be addressed to differentiate between specific semantics in a natural language.

Researchers have adopted various methods for NLP in the analysis of software requirements. In most of these methods, rules are defined by analyzing the structure of the requirement sentences, and candidates for analysis classes or use cases are extracted by applying the defined rules [2-5]. However, to parse automatically Korean

is very difficult because it has the flexible word order, and omissions and abbreviations occur frequently. To solve this problem, we applied statistical information extraction techniques to the identification of analysis classes. This technique has advantages as needless to parsing and portableness to other domains

We propose a statistical method for processing semantics by extracting classes consists of three tasks. In the first, the raw corpus is annotated with parts-of-speech *(POS)* tags by automatic morphological analyzer and begin-inside-or-outside *(BIO)* tags [6] manually. In the second, a model for the identification of analysis classes is learned by the annotated corpus and the conditional random fields (CRFs) classifier [7]. In the third, the proposed system extracts the analysis classes and measures itself.

The purpose of this paper is to propose a method for the identification of candidate analysis classes which can work on requirement documents even in the languages with a flexible word order, such as the Korean language. To the best of our knowledge, our work is the first to report the results of statistical method for the automatic identification of candidate analysis classes. We hope that it can contribute to lessen human effort in the process of software development.

2 Annotating Corpus

The raw corpus is constructed by sentences that are present basic flow about normal activity for system on the description of a use case expressed in natural language. And they are not dependent in specific domain of application. In this paper, we collected 554 sentences with 472 classes with 29 descriptions of use cases in a banking operation. To raise the performance of a statistical classifier, one needs a large corpus. Unfortunately, the raw corpus as mentioned above is rather small. However, we cannot describe additional requirements to extend the corpus for description of a use case. Therefore, we suggest an alternative of collect additional sentences from the Web. To expand our corpus, we used *Google* as a Web searcher, classes as queries, and sentences of snippets searched as additional sentences. We collected an additional 9,763 sentences with 18,300 classes. The sentences in the original raw corpus have a frequency of classes of 0.85, and that is very small. The extended raw corpus has a frequency of 1.87, and that is noticeably high. This new raw corpus has low quality, but it can be used as an open system because the recall of the statistical is enhanced by the increase in the number of objects that can be extracted as patterns about classes.

We then needed to tag our raw corpus. Unlike the parsing process, automatic morphological analyzers and *POS* taggers have accuracies of more than 90%, even in Korean. In English, a word is a spacing unit, but in Korean, an *eojeol* that consists of one or more morphemes comprises a spacing unit. We adopted SMASH[8] as automatic morphological analyzer and *POS* tagger. And, we added manually *BIO* tags[6]; *B* is "Beginning of class," *I* is "Inside of class," and *0* is "Outside of class".

3 Learning the CRFs model

In this paper, the proposed method uses only morphemes and POS tags as features that are not dependent on a specific domain. This characteristic can be portable itself how the system needs only a statistical classifier and a morphological analyzer. The features for the CRFs classifier are extracted in a window that has a size of w around a target morpheme. In this window, morphemes of n-grams and POS tags of m-grams are extracted. The n-gram model is very famous and has been used often in field of the text categorization, etc [9]. In a corpus with a small volume, if a target morpheme is used to features, then the statistical model can work like a dictionary and its evaluation can be unfair. We did not use a target morpheme as a feature.

We constructed many combinations of features, and evaluated each combination, and adapted it to the system. To evaluate performance, we chose *Recall*, *Precision*, and F_2*-Score*. *Recall* is a rate of how many classes are identified compared to the classes that should be identified. *Precision* is a rate of how many correct classes are identified among the identified classes by the system. F_{β}*-score* is a harmonic means between *Recall* and *Precision*. In general, F_1*-score* is most used that has a rate of 1:1 for *Recall* and *Precision*. But, we thought that the most important factor is the identification of maximum number of classes because our goal is the reduction of human intervention. So we used F_2*-score* that has a rate of 2:1 for *Recall* and *Precision*.

We evaluated each feature that was set to a different size w of window, n-grams of morphemes, and m-grams of POS tags. In Table 1, Feature3 (w=5, n=2 and m=2) was chosen because it has the highest value F_2*-Score*.

Table 1. The features and their performances

Feature **ID**	Lexicon	POS	*Recall*	*Precision*	F_2 *score*
Featurel	n=1	m=1	0.7380	0.8455	0.7573
Feature2	n=1,2	m=1	0.8076	0.8934	0.8234
Feature3	**n=1,2**	**m=1,2**	**0.8381**	**0.8955**	**0.8490**
Feature4	n=1,2	m=1,2,3	0.8286	0.8887	0.8399

We learned the statistical model that can classify *BIO* tags by using CRF++ [10] as a CRFs classifier. The goal is to extract as many classes as possible. So, although the probability of the *0* tag is bigger than the probability of the *B* or *I* tag, we choose these tags when these are bigger than threshold *0*. If both of their probabilities are bigger than *0*, the larger one is chosen in both. If the system adopts the concept about threshold *0* for *B* and *I* tags, the size of the elicited classes can be maximized There is an issue as to how to decide the proper value of threshold *0*. We observed the performance of the system when threshold *9* is changed. And we set threshold *0* to 0.4 with the highest F_2*-score* as the result of this experiment, as shown in Table 2.

Table 2. The performance by change in threshold 0

0	*Recall*	*Precision*	F_2 *score*
no_threshold	0.8381	0.8955	0.8490
0.5	0.8381	0.8942	0.8487

0.4	**0.8862**	**0.8842**	**0.8858**
0.3	0.9016	0.8269	0.8456
0.2	0.9278	0.7338	0.8812
0.1	0.9574	0.6753	0.8836

4 Extracting the Classes and Testing

For evaluating the performance of the extraction of *BIO* tags and classes, we proceeded with 5-fold cross validation wherein we separate the raw corpus into 5 equal parts and repeat learning and testing with a rate of 4:1. We used *Precision, Recall,* and F_2*_score.* We used the Feature3 set the threshold *0* to 0.4. As a result, in Table 3, the system has an F_2 *score* of 0.8857 for *B* and *I* tags.

Table 3. The performance for the extraction of BIO tags

BIO tag	*Recall*	*Precision*	F_2*_score*
B	0.8163	0.8379	0.8206
1	0.9560	0.9306	0.9508
Average	0.8862	0.8842	0.8857

We want to extract class, not *BIO* tags, so we evaluated the performance of the extraction of classes that has combined morphemes by using *BIO* tags for each morpheme. A way of combining is to reverse the definition of *BIO* tags. By this definition, the first morpheme of the class should have *B* tag, and the next morphemes should have *I* tag. This extraction rule can be represented by a regular expression like $(^5)$.

$$Class = B + I * \qquad (^5)$$

The extraction of classes corresponded to (5) in sequences of *BIO* tags. For example, in case of *0-0-0,* there are no elicited classes. In *O-B-I-I-0*, *B-I-I* can be elicited as the class. And in *0-I-I-1-0,* no classes are elicited by (5) although - in reality - *I-I-I* is possible as a class. So, we try to increase recall by the addition of another definition that is a transformed (5). This extraction rule is (6) that reflects an unconventional approach for a sequence of *BIO* tags.

$$Class = (B\ 11\ )+ \qquad (6)$$

Table 4 shows the performances when (5) and (6) are adopted respectively. F_2_score was 0.8010 and 0.8588, respectively. In (5), the performance for the elicitation of classes was lower than the performance of the extraction of *BIO* tags because the system can pick up the wrong boundary of classes. (6) increases the F_2 *score* more than 5% absolutely and more than 29% relatively.

Table 4. The performance for identification of classes

Extract Rule	*Recall*	*Precision*	*F2 score*
(5)	0.7944	0.8283	0.8010
(6)	0.8771	0.7925	0.8588

5 Discussion and Conclusion

In this paper, we propose a system that can automatically elicit analysis classes from requirement sentences in a natural language, without using the rule-based methods of previous systems. We used a statistical method to elicit classes automatically. We focused on the distinctiveness of Korean language in this study and on the fact that the construction of a complete rule set that can substitute for human intuition is very difficult. We constructed a statistical model for information extraction that has been used in natural language processing, and we tried to validate the effectiveness of our proposed system in the identification of classes.

Previously, authors have tried to validate their approach in their own way. However, there has been little research quantitative performance. Therefore, we can say that our system can elicit classes more than 80% of the time by any measure of *Precision, Recall,* or *F2 score*.

Surely, there are more issues. One is evaluation to maximize recall according to the threshold for probability of *BIO* tags and the formula for identification of classes. This may be evaluated by porting to the various software applications. Then, the construction of learning data needs the human intuition. This data should adapt to the type of semantic web that can be reused in a similar environment. Our future works will be studies on the reuse of semantics for minimizing human efforts in preprocessing and the validation or adjustment of feature, threshold, etc. in learning.

Acknowledgment This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)(NIPA-20 **12-** (H0301-12-3004)).

Reference

1. Booch, G., Maksimchuk, R., Engle, M., Young, **B.,** Conallen, J., Houston, K.: Object-Oriented Analysis and Design with applications, third edition. Dan Joraanstad, Addison-Wesley Professional, Boston (2007)
2. Vinay, S., Aithal, S., Desai, **P.:** An Approach towards Automation of Requirements Analysis. In: International Multi-Conference of Engineers and computer Scientists 2009, vol. 1, pp. 1080--1085. Hong Kong (2009)

3. Kumar, D.D., Sanyal, R.: Static UML Model Generator from Analysis of Requirements. In: Advanced Software Engineering and Its Applications 2008, pp. 77--84. IEEE Computer Society, Washington (2008)
4. Giganto, R., Smith, T.: Derivation of Classes from Use Cases Automatically Generated by a Three-Level Sentence Processing Algorithm. In: 3rd International Conference on Systems 2008, pp. 75--80. IEEE Computer Society, Cancun (2008)
5. Liu, D., Subramaniam, K., Eberlein, A., Far, B.H.: Natural Language Requirements Analysis and Class Model Generation Using UCDA. In: Orchard, R., Yang, C., Ali, M. (eds.) IEA/AIE 2004, LNCS, vol. 3029, pp. 295--304. Springer, Heidelberg (2004)
6. Ramshaw, L.A., Marcus, M.P.: Text Chunking using Transformation-Based Learning. In: 3rd Workshop on Very Large Corpora, pp. 82--94. Yarowsky, D., Church, K., Cambridge (1995)
7. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: 18th International conference on Machine Learning, pp. 282--289. Williamstown (2001)
8. Yang, J.: Korean Morphological Analysis for Mobile Devices with Limited Hardware Resources. Master Thesis, Sogang University, Seoul (2009)
9. Jo, T.: Representation of Texts into String Vectors for Text Categorization. Journal of Computing Science and Engineering, 4, 2, 110-127 (2010)
10. CRF++: Yet Another CRF Toolkit, http://crfpp.sourceforge.net