SVM-based Clause-dependency Determination in Syntactic Analysis

Mi-Young Kim
School of Computer Science and Engineering
Sungshin Women's University
Dongseon-Dong 3-Ga, Seongbuk-Gu, Seoul
Republic of Korea

Abstract: - In syntactic analysis of a long sentence, it is difficult to determine the dependency among clauses. To handle such syntactic ambiguity among clauses, this paper proposes a machine learning-based determination method of clause-dependency. We extract various features from clauses, and experiment using SVM machine learning algorithm. Our experimental results showed that the proposed SVM-based method outperformed the previous methods.

Key-Words: - syntactic analysis, dependency, clause, SVM

1 Introduction

Syntactic analysis based on dependency structure is popular in relatively free word-order languages – such as Korean and Japanese. To construct a dependency structure is to find the governor-dependent relation for each syntactic unit. The longer the input sentences, the worse the parsing results. So, a long sentence is usually segmented into several clauses, and syntactic analysis of each clause is performed. Then, we must merge all the analysis results into one dependency structure. Many ambiguities exist in determining the dependency relation among clauses. Therefore, this paper proposes an SVM-based clause-dependency determination. This paper extracts some clause-specific features, and analyzes the effect of each feature on the performance. In addition, we compare the performance of our proposed method with those of previous rule-based methods.

This paper is organized as follows. Section 2 presents previous work on determining the dependency among clauses. In section 3, SVM-based clause-dependency determination method and features will be explained. In section 4, experimental results will show that the proposed features are effective in clause-dependency determination, and we also analyze the effect of each feature on the clause-dependency determination. Finally, a conclusion will be given.

2 Previous Work

Several studies have focused on the dependency problem among clauses [1], [2], [3], [4], [5], [6]. In the syntactic analysis, the dependency ambiguity among clauses is a serious problem. So, many researchers have tried to solve the problem by detecting the scope embedding preference of clauses. If the scope of a clause 'A' is embedded within the scope of a clause 'B', then 'B' is determined the governor of the clause 'A'.

Shirai[1] and Minami[3] classified the surface forms of the endings of the final words in clauses into several classes, and constructed the embedding preference rules among classes. Roh[5] also constructed heuristic rules to determine the dependency among clauses according to the comma and conjunction information. These rules have limits that they cannot cope with many exceptional cases.

To complement rule-based methods, Kawahara[2] and Utsuro[6] applied Decision List machine learning algorithm. Utsuro[6] used four features from a clause– punctuation, grammatical tag, conjugation form of the final conjugative word, and lexicalized form. Kawahara[2] used only two features – surface form of the ending of the final word in a clause, and comma information. Kawahara[2] used the original surface forms, not classifying them into some classes. He insisted that the classifications were too coarse to handle the scope length among classes precisely. These machine learning-based methods did not show how much each feature contributed to the performance. Also, the performances of these machine learning-based methods were not compared with the previous rule-based methods. So, they did not show

these machine learning-based methods worked better than rule-based methods.

To determine the dependency among clauses, we propose an SVM-based machine learning method. We also describe some features from clauses.

3 How to Determine Clause-dependency

3.1 SVM machine learning

To determine the dependency among clauses, we apply SVM method. SVM algorithms have been successfully applied to NLP problems[8, 9, 10]. SVMs have good characteristics to cope with the data sparseness problem and achieve high generalization even with training data of a very high dimension. Since SVMs are binary classifiers, we must extend SVMs to work as multi-class classifiers in order to classify three or more classes. As a machine learning program, we employed LIBSVM[11] for multi-class.

3.2 Features

Each clause has 4 features, as shown in Table 1. The 1st feature takes the value of the surface form of the last ending of a predicate. Korean is an agglutinative language and the ending of a predicate indicates the connective function with the next clause. As a 2nd feature, a semantic concept of a predicate is used. A semantic concept expressed by the Kadokawa thesaurus is divided into 1110 semantic classes. The 3rd feature is a clause-specific one. It indicates whether a clause shares the same subject with a 'dependent clause' or not. When clauses require the same subject, the subject appears only once in a sentence and clauses share the subject. In many cases, the real governor of a 'dependent clause' is the nearest clause. However, if the nearest clause does not share the same subject with the 'dependent clause', then the real governor can be a far clause that shares the same subject. So, the subject-sharing information is important. The 4th feature deals with information on whether a predicate is followed by a comma or not. The use of a comma to insert a pause in a sentence is an important key in determining the embedding scope of clauses. So, it influences the dependency among clauses. We use the information of 7 clauses – one is for a 'dependent clause' and the other 6 clauses are for the near clauses in the right side of the 'dependent clause'. The class set consists of 6 values (1~6) to indicate the position of the real governor clause.

4 Experimental Evaluation

The experiment to determine the dependency among clauses was evaluated under the KIBS[1] data set of 19,000 Korean sentences, average 13.57 words/sentence, using 10-cross validation.

In the experiments, the precision of our method is 81.97 percent (see Table 2).

To compare our performance with those of the previous methods, we performed two other experiments. One is the experiment based on the rule 'nearest modifiee principle'. This principle determines that the governor clause is the right near clause of a 'dependent clause'. The other experiment is based on Noma[4]'s rule. As described in Table 2, our method showed better performance.

As shown in Table 3, the clause-dependency precision became better when we used the 3 features without semantic information. It means the semantic concept information gives bad effect on determining the dependency among clauses. We can conclude that the meaning of a predicate in each clause is not important in determining the dependency among clauses.

In addition, the most effective feature is the surface form of the ending of a predicate. As mentioned before, the surface form information indicates the connective function with the next clause (e.g. '*umulo*(because)' indicates it functions as a reason for the next clause). So, it contributes most significantly to the dependency among clauses.

5 Conclusion

This paper described a clause-dependency determination method based on Support Vector Machines (SVM). We extract four features from clauses, and determine the dependency among clauses. The experimental results show that the proposed method outperforms the previous methods, with the precision of 81.97 percent. We also analyzed the 3 features-- surface form of the ending of a predicate, subject-sharing information, and comma-- contributed to the performance. Especially, surface form

[1] Korea Information Base provided by KORTERM/KAIST

Table 2: Performance comparison b/w our method & previous methods

	Our Proposed Method	**Method using 'Nearest Modifiee Principle'**	**Method using Noma[4]'s rules**
Precision of dependency among clauses	81.97%	62.06%	70.60%

Table 3: Precision change when one kind of feature is removed

Features	**Precision change**
Using all 4 features	0.00%
3 features without surface form	-3.63 %
3 features without semantic information	+1.91%
3 features without subject-sharing information	-1.99%
3 features without comma	-2.20%

information gave most significant contribution to the performance. However, semantic code feature decreased the performance.

We plan to continue our research as following. We will analyze the error types of clause-dependency after this method, and try to improve the performance using deep-level knowledge.

Acknowledgements

This work was supported by the Sungshin Women's University Research Grant of 2006.

References:

[1] S. Shirai, S. Ikehara, A. Yokoo, and J. Kimura. "A new dependency analysis method based semantically embedded sentence structures and its performance on Japanese subordinate clauses." Transactions of Information Processing Society of Japan, 36(10):2353-2361, 1995

[2] D. Kawahara and S. Kurohashi. Corpus-based Dependency Analysis of Japanese Sentences using Verb Bunsetsu Transitivity, In Proceedings of the 5th Natural Language Processing Pacific Rim Symposium, pp. 387-391, 1999

[3] F. Minami. "Gendai Nihongo no Kouzou' (structures of Modern Japanese Language), Taishuukan shoten, 1974

[4] Noma Hideki, "The relations between Korean surface forms and grammars (in Korean)", 2002

[5] L. Danlos,, "Sentences with two subordinate clauses : syntax, semantics and underspecified semantic representation'. In Proceedings of the TAG+7 Workshop, p.140-147, 2004

[5] Yoon-Hyung Roh, Young Ae Seo, Ki-Young Lee, Sung-Kwon Choi: Long Sentence Partitioning using Structure Analysis for Machine Translation. Proceeding on NLPRS, p.646-652, 2001.

[6] T. Utsuro, S. Nishiokayama, M. Fujio, and Y. Matsumoto, "Analyzing dependencies of Japanese subordinate clauses based on statistics of scope embedding preference," Proc. 1st Conference of the North. American Chapter of the ACL, pp.110–117, 2000

[7] M. Kim, S. Kang and J. Lee, "Text chunking by rule and lexical information *(in Korean)*", Proc. 12th Hangul and Korean Information Processing Conference, Chonju, Korea. pp. 103~109. 2000.

[8] T. Kudo and Y. Matsumoto, "Chunking with Support Vector Machines", Proc. 2nd meeting of North American Chapter of Association for Computational Linguistics (NAACL), Pittsburgh, PA, USA, pp.192-199, 2001

[9] T. Kudo and Y. Matsumoto, "Japanese Dependency Structure Analysis Based on Support Vector Machines", Proc. 2000 SIDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), Hongkong, China, pp.18-25, 2000

[10] H. Yamada and Y. Matsumoto, "Statistical Dependency Analysis with Support Vector Machines", Proc. 8th International Workshop on Parsing Technology, Nancy, France, pp.195-206, 2003

[11] Chih-Chung Chang and Chih-Jen Lin, 2001. LIBSVM :. a library for support vector machines. Software. available at http://www.csie.ntu.edu.tw/cjlin/libsvm