International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:1, 2007

Resolving Dependency Ambiguity of Subordinate Clauses using Support Vector Machines

Sang-Soo Kim, Seong-Bae Park, and Sang-Jo Lee

***Abstract*—**In this paper, we propose a method of resolving dependency ambiguities of Korean subordinate clauses based on Support Vector Machines (SVMs). Dependency analysis of clauses is well known to be one of the most difficult tasks in parsing sentences, especially in Korean. In order to solve this problem, we assume that the dependency relation of Korean subordinate clauses is the dependency relation among verb phrase, verb and endings in the clauses. As a result, this problem is represented as a binary classification task. In order to apply SVMs to this problem, we selected two kinds of features: static and dynamic features. The experimental results on STEP2000 corpus show that our system achieves the accuracy of 73.5%.

***Keywords*—**Dependency analysis, subordinate clauses, binary classification, support vector machines.

I. Introduction

IN Korean, the dependency analysis of clauses is known as one of the most difficult tasks in parsing sentences because of the characteristics of Korean. The characteristics of Korean are that (i) it is a partially free word-order, (ii) the omission of components is common, (iii) it is a head-final language, and (iv) the spacing unit is a composite of one or more words. Especially, what makes the clause dependency analysis difficult is the third factor. The endings (*Eomi*) can be freely combined with a verb, and they contain the semantic relationship with other verbs.

The steps of parsing Korean sentences are as follows. First, the input sentence is analyzed into the morphemes, and then the part-of-speech (POS) of the morphemes is determined by some means. Finally, the syntactic relation is analyzed using the results of the previous steps. Due to the characteristics of Korean, the dependency grammar rather than phrase-structure grammar is generally used in parsing Korean [1]. This process is not much different from other languages. However, since each word used in a sentence becomes a processing unit, the complexity of parsing gets too large especially with long sentences, which results in severe ambiguities in parsing Korean.

Recently, in order to solve this problem, enlarging the processing unit gains much interest from researchers of Korean language processing. Many kinds of research results are reported on Korean text chunking [2,3], and they gives relatively stable results. In addition, some researchers have studied to find the boundaries of a clause [4]. When the clause boundaries are known, intra-clause parsing is a simple task compared to inter-clause parsing. This is because Korean is head-final. However, the relation between clauses is not determined by the information given by text chunking and clause boundaries.

Due to the freedom of word ordering in Korean sentences, it is extremely difficult to determine the relation between clauses by seeing just the neighbor words. That is, it has been believed that the surface form of a clause is not sufficient to analyzing the relation among clauses. As a result, many previous works on parsing Korean have focused on how to use semantic information of verb phrases in determining the clause relation [1]. However, it is a very expensive and time-consuming task to build semantic knowledge for the task.

In this paper, we propose a novel method of analyzing dependency relation of Korean subordinate clauses without external knowledge. For this task, we witnessed that the most important component in determining the dependencies is the base verb phrase composed of a verb and a few endings, rather than the complement and supplement components within a clause. Therefore, in order to solve the problem, we assume that the dependency relation of Korean subordinate clauses is the dependency relation of base verb phrases. In addition, we formulate the dependency analysis of Korean clauses to a binary classification task. As a classifier for this task, we adopt a support vector machine (SVM) which is known as the best classifier for many kinds of real-world classification problems.

The rest of this paper is organized as follows. Section 2 surveys the previous work on clause recognition and analysis of inter-clause relation, and Section 3 introduces how a support vector machine works which is adopted as a base learner for the task. Section 4 describes the proposed method for clausal dependency analysis using support vector machines. Section 5 explains the corpus used in the experiments and presents the experimental results. Finally, Section 5 draws conclusions and suggests some future work.

This research was supported in part by MIC & IITA through IT Leading R&D Support Project, and by grant No. R01-2006-000-11196-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

Authors are with Department of Computer Engineering, Kyungpook National University, Daegu 702-701, Korea (corresponding author to provide e-mail: sskim@sejong.knu.ac.kr).

II. Related Work

There have been a number of studies for analyzing dependency relation of subordinate clauses in the clause identification and the dependency structure analysis. The clause identification is a task of recognizing the embeddedness of clauses, while the clause identification is to find the starting and ending points of clauses.

In 2001, there was a competition for this task at the Conference on Computational Language Learning (CoNLL). The best two methods are a boosting tree [5] and a hidden Markov Model [6]. However, unlike western languages, in Korean the dependency relation is not easily determined even if the clause boundaries are identified.

Most previous work on dependency analysis of sentences has focused on the words rather than clauses. That is, instead of finding the dependency relation among clauses, the relation among verb phrases within a clause has been the core of the research. Uchimoto et al. used a maximum entropy model and various kinds of features to identify dependency structure of sentences [7]. They reported the experimental results on the relationship between feature types and dependency analysis. Kudo and Matsumoto formulated the analysis of dependency structure as a binary classification task, and adopted support vector machines as a classifier [8]. The features used in training support vector machines are grammatical features such as lexicons and part-of-speech tags, and some functional features such as functional words and inflection information.

Gao and Suzuki solved the problem of analyzing dependency relation by training a language model through an unsupervised learning [9]. Utsuro et al. classified text chunks into several types according to the functional words of the final word in a sentence. With the classified type, they determined the dependency relation among chunks [10].

In Korean language processing, most research on syntactic analysis has been focused on the Josa and Eomi, and their dependency relation. As a result, most works are based on the hand-crafted rules [1]. Especially, the research on the subordinate clauses was performed on the recognition simple sentence and restoration of the omitted components in the simple sentence. The first effort to use a machine learning algorithm in handling clauses was done for clause boundary detection. Lee et al. extracted *n*-gram information from a sentence, and then recognized the boundary of a clause using the information [11]. However, their work was limited to detection of clauses, and did not suggest any method for analyzing their dependency.

The main reason why the machine learning algorithms are rare in handling Korean clauses is that there is no standard large-scale dataset for the task. Recently the great funding of the Korean government in writing a large-scale tree-tagged corpus makes it possible to transform the corpus into the data for clause detection and their dependency analysis.

III. Support Vector Machines

Support Vector Machine (SVM) proposed by Vapnik is a kind of machine learning algorithms, and is well known as the most successful binary classifier, and have been applied to many classification tasks. In the field of natural language processing, it has been successfully applied to text categorization, spam-mail filtering and chunk identification, and it is reported to accomplish high performance without falling into over-fitting even with a large number of features [12, 13].

Assume that the training data with either positive or negative class as follows:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)$$

$$\mathbf{x}_i \in R^n, \quad y_i \in \{+1, -1\}$$

where $\mathbf{x}_i$ is a feature vector of the *i*-th training datum in an *n*-dimensional space, and $\mathbf{y}_i$ is its class label.

In the basic SVM framework, the hyperplane is defined as follows:

$$(w \cdot \mathbf{x}) + b = 0, \quad w \in R^n, \quad b \in R.$$

According to the hyperplane definition, there could be the infinite number of hyperplanes that can separate training data into two classes correctly.

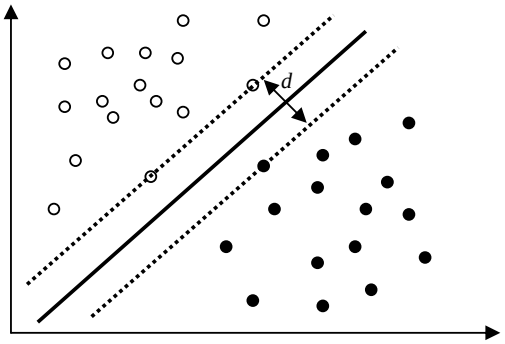

Fig. 1 The margin of a hyperplane

Among such hyperplanes, we define the optimal hyperplane as the one with the largest margin between two classes. Fig. 1 illustrates the notion of the margin. The solid line, hyperplane, correctly divides training data into two classes without misclassification. Two dash lines which are parallel with the hyperplane represent the distance between hyperplane and the closest instance. The distance between each parallel dash lines, *d*, is called the *margin*. Thus, assuming that the nearest distance is 1, the margin can be rewritten as:

$$(w \cdot x) + b \leq +1$$

$$(w \cdot x) + b \geq -1$$

$$d = \frac{2}{\| w \|}$$

Therefore, SVM generates a hyperplane which maximizes a margin by minimizing ||w|| under the constraints:

$$y[(w \cdot \mathrm{x}_i) + b] \geq 1$$

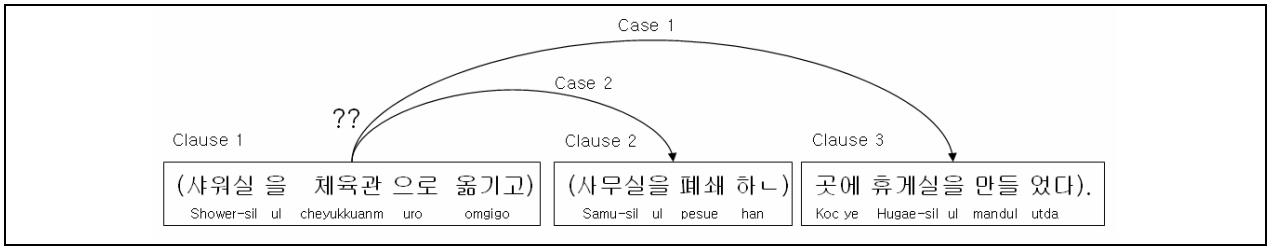

Fig. 2 An example of a dependency relation between clauses

SVMs have an advantage over conventional machine learning algorithms such as neural networks or decision trees. SVMs show higher generalization performance independent of the dimension of feature vectors. Conventional machine learning algorithms usually require careful feature selection, which is often optimized heuristically to avoid over-fitting. SVMs also can carry out their learning with all combinations of given features without increasing computational complexity by introducing the kernel function.

IV. Analyzing Dependency Relation of Subordinate Clauses

A. The Probability Model and Generating Training Data

Let a sequence of clauses be $\{c_1, c_2, ..., c_n\}$ denoted by C, and the sequence dependency patterns be {Dep(1), Dep(2), …, Dep(n-1)} denoted by D, where Dep(i)=j implies that the clause c_i modifies the clause c_j. In Koran under this framework, this dependency relation has to satisfy some constraints. A clause has only one dependency relation except for the rightmost one. It means that a clause modifies only one clause.

A dependency relation is defined as a searching problem for dependency pattern D that maximizes the conditional probability P(D|C). That is,

$$D_{best} = \arg\max_{D} P(D \mid C)$$

If we assume that the dependency probability is independent one another, P(D|C) can be rewritten as:

$$P(D \mid C) = \prod_{i=1}^{m-1} P(Dep(i) = j \mid \mathrm{f}_{ij}\})$$

$$\mathrm{f}_{ij} = \{f_1, ..., f_n\} \in \mathrm{R}^n$$

where f_{ij} is an n-dimensional feature vector that represents relation between clauses.

In order to use SVMs in analyzing the clausal dependency, we generate positive and negative examples. We adopt simple and effective method for this purpose.

$$\bigcup_{\substack{1 \le i \le m-1 \\ i+1 \le j \le m}} (\mathrm{f}_{ij}, y_{ij}) = \{(\mathrm{f}_{12}, y_{12}), (\mathrm{f}_{23}, y_{23}), ..., (\mathrm{f}_{m-1\,m}, y_{m-1\,m})\}$$

$$\mathrm{f}_{ij} = \{f_1, ..., f_n\} \in \mathrm{R}^n$$

$$y_{ij} \in \{Dep(+1),\ Not-Dep(-1)\}$$

TABLE I
Features used for Analyzing Dependency Relation

Static Features	Lexicon Information	**Left Clause** A word of verb POS tag of verb A word of endings POS tag of endings **Right Clause** A word of verb POS tag of verb A word of endings POS tag of endings
	Position Information	Distance between left and right clause Position index of left and right Clauses
Dynamic Features		A syntactic relation between clauses

According to the above equation, we generate pairs of two clauses in the training data, and then take a pair of clauses that are in a dependency relation as a positive example, and two clauses that appear in a sentence but are not with a dependency relation as a negative example.

Fig. 2 shows an example of dependency relation extraction between clauses. In this example, clause 1, 2 and 3 means “shower-room was moved to the gymnasium”, “office-room was closed,” and “a rest room was made in the place (original place of shower and office room).” In this case, we can generate one positive example (Case 2 in Fig. 2) and one negative example (Case 1 in Fig. 2).

B. Feature Selection for Analyzing Dependency Relation

In Korean language, the clauses are divided into three types that are one to modify other clause (called *conjunctive clause*), one to modify a noun phrase (called *prenominal clause*), and one to imply the end of sentence (called *final ending clause*). Among these clause types, prenominal clause and conjunctive clause make dependency relation. We select dependency relation that conjunctive clause was depend on other clause, because prenominal clause make a simple dependency relation with to modify a next appearing noun phrase. The conjunctive clause makes the dependency relation very complex and, thus, it is difficult to recognize dependency relation. The relation can be determined not according to simple syntactic information such as verb type and position in sentence but according to the

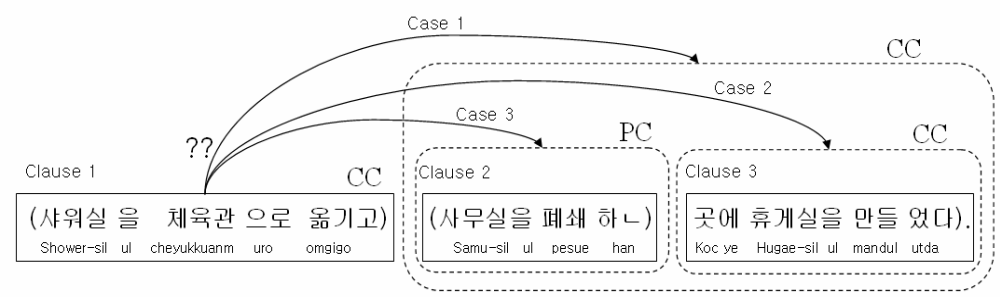

Fig. 4 An example of determining dependency relation using dynamic features

context of sentence and the inflection of endings.

In the previous section, we assume that the dependency relation of Korean subordinate clauses is the dependency relation of verb phrase, verb and endings, in the clauses. According to this assumption, we select two features that are static and dynamic features. The feature set is shown in Table I.

We define lexicon and position information appearing in a sentence as the static information. The lexicon information is a word and POS tags of verb and endings in the pair of left and right clauses. The positional information is the distance between clauses, and position index is the location of clauses in a sentence. We expect that this static features weakly represent the semantic information between clauses. Fig. 3 is shown the static features for Fig. 2.

No	Lexicon information		Position information (Distance, Position Index)
	Left Clause	Right Clase	
1	옮기/pvg 고/ecc	하/px ㄴ/etm	1, 1
2	옮기/pvg 고/ecc	하/xsv 었/ep 다/ef	2, 2

Fig. 3 The example static features

The dynamic features are the syntactic information in a sentence. Therefore, we make a simple CKY cart parser so that it captures syntactic information in the sentence. Table 2 shows a rule set for the chart parser. 'CC' implies a conjunctive clause and 'PC' implies a prenominal clause.

TABLE II
THE RULE USED FOR CHART PARSING

Rule 1: CC → CC CC
Rule 2: CC → PC CC
Rule 3: PC → PC PC
Rule 4: PC → CC PC

With dynamic features we can apply a syntactic relation of clauses to training the support vector machines. The syntactic relation states if a clause is composed of just one simple clause or more than one clause.

Fig. 4 shows an example of an analyzing dependency relation using the dynamic feature. The static feature of Case 1 and Case 2 are same, but the dynamic features are different. The dynamic feature of Case 1 is 'PC CC' determined by rule 2, but that of Case 2 is 'CC'. Table III shows the whole features for Fig. 3.

TABLE III
THE WHOLE FEATURES

No	Static features		Dynamic features
	Lexicon information	Position information	
1	옮기/pvg 고/ecc 하/px ㄴ/etm	1, 1	PC CC
2	옮기/pvg 고/ecc 하/px ㄴ/etm	1, 1	CC
3	옮기/pvg 고/ecc 하/xsv 었/ep 다/ef	2, 2	PC

V. EXPERIMENTS

For the evaluation of the proposed method, a data set for dependency analysis of clauses in Korean is prepared. This dataset is derived from the parse corpus, which is a product of STEP2000 project supported by the Korean government. The corpus consists of 6,934 sentences with 26,876 clauses. The corpus is divided into two parts: training (90%) and test (10%) set. Table IV shows a simple statistics on the corpus.

TABLE IV
COUNTS ON THE DATASET

Information	Training Set	Test Set
No. of all sentences	6,240	694
No. of all clauses	24,226	2,650
No. of prenominal and final ending clauses	15,457	1,666
No. of conjunctive clauses	8,769	984

Fig. 5 shows an example of dependency relation in the subordinate clause dataset. For the format of this dataset, we follow that of CoNLL-2001 shared task and additionally add the dependency relation of clauses to it. Each instance in the training and test data consists of six columns. The first column contains the lexicon, the second presents a part-of-speech tag. The third column contains the chunk tag. The verb phrases in these columns are used as static features. The fourth and fifth contain a beginning, S, and an ending, E, of clauses. The sixth column gives the relation index of clauses.

We apply SVM ^{Light}[14] for support vector machine, and experiment on three cases. The first and second case used only words and POS tags of clauses and the all of static features. The last case used both static features and dynamic features. The evaluation measure is defined as:

$$Accuracy = \frac{correctly\ recognized\ dependency\ relation\ of\ clauses}{total\ dependency\ relation} \times 100$$

When a clause makes several pairs of dependency relation with more than one clause, we select a pair which has the largest margin. Table V shows the experimental results. The base line is the model that determines the governor of a clause as the nearest one.

TABLE V
THE EXPERIMENTAL RESULTS

Features		Accuracy (%)
Base Line		57.50
Case 1	Only words and POS tags of clauses	64.40
Case 2	All of Static features	68.59
Case 3	All of Static and Dynamic features	73.50

In Case 1, when only words and POS tags of clauses are used, the accuracy is just 68.59%, That is, the proposed model improves 6.09% over the base line. It implies that the verb and endings have a dependency relation weakly. The second case which uses all of the static features shows 68.59% of accuracy. It means that the positional information in static features affect the dependency relation. In the last case, the results with both static and dynamic features are far better than those without dynamic features. That is, the model with dynamic features outperforms that with static features only.

The performance of our approach is a little bit lower than a performance of other researches that analyze the dependency relation in Japanese and European languages. It seems that our approach select only a relation of clauses without relations of word and phrases. It is easier to analyze the relations of word and phrases than to analyze the relation of clauses.

VI. CONCLUSION

We have proposed a method for analyzing dependency relation of Korean subordinate clauses based on Support Vector Machines (SVMs). In other to solve this problem, we assume that the dependency relation of Korean subordinate clauses is the dependency relation of verb phrase, verb and endings, in the clauses. We formulate this problem as a binary classification task. We selected two kind of features, static and dynamic features, for applying SVMs to this problem. The static features are word, POS tag, and the positional information, while the dynamic features include the syntactic information of the caluses. For extracting the dynamic information, we make a simple CKY chart parser with simple rules. The experimental results on STEP2000 corpus show that our system achieves the accuracy of 73.5%.

1	샤워실	ncn	B-NP	S	X	0	shower room
2	을	jco	I-NP	X	X	0	*POST*
3	체육관	ncn	B-NP	X	X	0	gymnasium
4	으로	jca	I-NP	X	X	0	*POST*
5	옮기	pcg	B-VP	X	X	0	move
6	고	ecc	I-NP	X	E	11	*ENDING*
7	사무실	ncn	B-NP	S	X	0	office room
8	을	jco	I-NP	X	X	0	*POST*
9	폐쇄	ncpa	B-VP	X	X	0	closed
10	하	xsv	I-VP	X	X	0	*ENDING*
11	ㄴ	etm	I-VP	X	E	12	*ENDING*
12	그곳	npd	B-NP	X	X	0	that place
13	에	jca	I-NP	X	X	0	*ENDING*
14	휴게실	ncn	B-NP	X	X	0	rest room
15	을	jco	I-NP	X	X	0	*POST*
16	만들	pvg	B-VP	X	X	0	Make
17	었	ep	I-VP	X	X	0	*ENDING*
18	다	ef	I-VP	X	X	0	*ENDING*
19	.	sf	O	X	E	-1	

Fig. 5 An example of dependency relation in the subordinate clause dataset

REFERENCES

[1] K.-J. Seo, *A Korean language parser using syntactic dependency relations between word-phrases*, M.S. Thesis, KAIST, 1993.

[2] S.-B. Park and B.-T. Zhang, "Text Chunking by Combining Hand-Crafted Rules and Memory-Based Learning," In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 497--504, 2003.

[3] H.-P. Shin, "Maximally Efficient Syntactic Parsing with Minimal Resources," In *Proceedings of the Conference on Hangul and Korean Language Information Processing*, pp. 242-244, 1999. (In Korean)

[4] H.-J. Lee, S.-B. Park, S.-J. Lee, and S.-Y Park, "Clause Boundary Recognition Using Support Vector Machines," In *Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence*, pp. 505--514, 2006.

[5] X. Carreras and L. Marquez, "Boosting Trees for Clause Splitting," In *Proceedings of the 5th Conference on Computational Natural Language Learning*, pp. 1-3, 2001.
[6] A. Molina and F. Pla, "Clause Detection using HMM," In *Proceedings of the 5th Conference on Computational Natural Language Learning*, pp. 70-72, 2001.
[7] K. Uchimoto, S. Sekine, and H. Isahara, "Japanese Dependency Structure Analysis Based on Maximum Entropy Models," In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 196-203, 1999.
[8] T. Kudo and Y. Matsumoto, "Japanese Dependency Structure Analysis Based on Support Vector Machines," In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 18-25, 2000.
[9] J. Gao and H. Suzuki, "Unsupervised Learning of Dependency Structure of Language Modeling," In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 521-528, 2003.
[10] T. Utsuro, S. Nishiokauama, M. Fujio, and Y. Matsumoto, "Analyzing Dependencies of Japanese Subordinate Clauses based on Statistics of Scope Embedding Preference," In *Proceedings of the 1st Conference on North American Chapter of the Association for Computational Linguistics*, pp. 110-117, 2000.
[11] H.-J. Lee, S.-B. Park, S.-J. Lee, and S.-Y Park, "Clause Boundary Recognition Using Support Vector Machines," In *Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence*, pp. 505-514, 2006.
[12] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.
[13] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," In *Proceedings of the European Conference on Machine Learning*, pp. 137--142, 1998.
[14] T. Joachims, *Making Large-Scale SVM Learning Practical*, LS8, Universitaet Dortmund, 1998.