Combining Rule-based Method and k-NN for Chunking Korean Text

Seong-Bae Park and **Byoung-Tak Zhang**
Artificial Intelligence Lab. (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
E-mail : {sbpark,btzhang}@scai.snu.ac.kr

Abstract

We present a chunking method for Korean which uses both a rule-based method and a machine-learning method. Given a sentence to be chunked, our method first divides it into chunks based on the heuristic rules, and then the chunks are verified by a machine-learning method, k-nearest neighbor algorithm. The method can therefore improve the preexisting simple rule-based chunkers. An evaluation of the proposed method on the sentences gathered from Korean homepages yields 2.3% of improvement over the simple method only with heuristic rules.

Keywords : chunking Korean, k-nearest neighbor, heuristic rules, combined system

1 Introduction

Text chunking is to divide sentences into non-overlapping segments on the basis of fairly superficial analysis. Abney [1] proposed that chunk can be an useful intermediate step for a full parsing. Let us consider the following sentence in Korean.

> 작은 세 마리의 곰이 마당에서 놀다가 잠든 듯 하다.
> *small three bear*-NOM[1] *playground*-LOCA *play*-AFTER *sleep seem*
> (Three small bears seem to sleep after playing at the playground.)

This sentence can be chunked as follows.

- [**NP** 작은 세 마리의 곰이] [**NP** 마당에서] [**VP** 놀다가] [**VP** 잠든 듯하다.]
  ([**NP** *small three bears*-NOM] [**NP** *playground*-LOCA] [**VP** *play*-AFTER] [**VP** *sleep seem*])

As we regard a chunk as syntactically correlated parts of words, the eight words in the above example are grouped into four chunks. Because the complexity of most natural language parsers is O(n^3) where n is the length of an input sentence, the chunking achieves great efficiency in full parsing.

Since Ramshaw and Marcus applied the machine learning method in text chunking [9], many researchers have used a statistics-based or machine learning methods to capture the best hypothesis on text chunking [2, 3, 4, 5, 8, 13]. On the other hand, the rule-based methods are widely used in chunking Korean [7, 12]. The well-developed postpositions and endings of Korean enable the simple heuristic rules to show high performance. However, the rules obtained by thorough observation on the target text give no guarantee of the optimum. This fact implies that the rules must be changed if the text which is used to construct them changes. Thus, more careful exploration in constructing the chunking rules is required. This is important because the misguided chunks incur fatal and unrecoverable errors in syntactic analysis.

In this paper, we describe a novel method on chunking Korean which combines a rule-based method and a machine learning method. Since the application of the rule-based method gives high accuracy by itself, it is used as a base method and then augmented by a machine learning method, k-nearest neighbor algorithm.

The rest of this paper is organized as follows. Section 2 gives the simple introduction on a Korean chunking system. Section 3 explains how the Korean sentences can be chunked based on heuristic rules. Section 4 describes a Korean chunking system combining with heuristic rules and k-NN, the machine learning technique. Section 5 presents the experimental results and Section 6 draws conclusions.

2 Overview of Korean Chunking

2.1 Characteristics of Korean

Korean is a head-final and agglutinative language. Due to its agglutinative property, the boundary of chunks can be determined relatively easily compared to other languages such as English or Chinese. Since the postpositions give much information on grammatical relation, they are used in chunking Korean. For a noun phrase, for instance, it is certain that the noun with a postposition except possessive ones become the end of a noun phrase. But, more observation is needed for the start of a noun phrase. Since the nouns without postpositions also can be the end of a noun phrase, the successive nouns without postpositions can be divided into two chunks. Thus, it is difficult to find the start of a noun phrase.

By the way, it is more difficult for verb phrase. Since

[1] In this paper, we use four case markers such as:
NOM = nominative, OBJ = objective, AFTER = time after, LOCA = locative

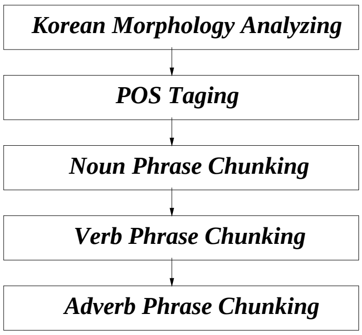


Figure 1: The procedure for chunking a Korean sentence into chunks.

each verb has one or more endings, it is difficult to find out which kinds of endings yield the start or the end of verb phrases. Thus, it is required to construct the separated knowledge for various kinds of phrase.

2.2 System Structure

Figure 1 shows an overall structure of the system to chunk given a Korean sentence. The morphological analysis is, in general, the first step in natural language processing. Because Korean is an agglutinative language, not only the base form of a given word and part of speech but also various kinds of syntactic features are revealed by the Korean morphological analyzer. The morphological analyzer usually outputs ambiguous results so that the POS (part-of-speech) tagger is needed to resolve such ambiguity.

Since a Korean sentence can be, in general, divided into noun phrases, verb phrases and adverbial phrases, in the proposed system are there three chunkers which take charge of noun phrase, verb phrase and adverb phrase respectively. Each chunker is further explained in the following section.

3 Chunking Korean by Heuristic Rules

3.1 Noun Phrase Chunking

With the well-developed postpositions of Korean, the noun phrase chunking can be accomplished with ease [7, 12]. The heuristic rules for Korean noun phrase chunking can be formulated as follows:

- Rule 1 : (Determiner)* NP
- Rule 2 : (Noun)* NP
- Rule 3 : (Pronoun)* NP
- Rule 4 : (Possessive Postposition)* NP
- Rule 5 : (Relative Postfix)* NP
- Rule 6 : (Relative Ending)* NP

	No. of Cases	Probability
Forming Chunk	775	0.61
Not Forming Chunk	496	0.39

Table 1: The possibility that an adverb sequence forms a chunk.

Since determiner, noun and pronoun play the similar syntactic role in Korean as the indeclinable parts of speech, they forms noun phrase chunk when they appear in succession without any postposition. Rule 1-3 imply this.

All nouns with postposition become the end of noun chunk, but there is only two exceptions. When the postposition is the possessive one, it is still in the mid of noun phrase (Rule 4). In the example sentence above, '마리의 (*mari-eui*)' is decomposed into '마리 (*mari*)' and '의 (*eui*)', where '의 (*eui*)' is the possessive postposition, so that it is in the mid of the first noun phrase chunk. The other exception is when the preceding words is finished with *relative postfix* '적 (*jeok*)' (Rule 5). Such a word in general does not constitute an independent phrase. Rule 6 states that a simple relative clause without any sub-constituents also does not constitute an independent phrase.

In rule 5, we include the cases where a *copular postposition* '이 (*yi*)' and a *relative ending* appear in succession after the relative postfix '적 (*jeok*)'. This is because such cases play the same grammatical role as the single *relative postfix* does.

3.2 Verb Phrase Chunking

The verb phrase chunking has been studied for a long time in the name of *compound verb processing* in Korean and shows relatively high accuracy. Shin and Kim used a finite state automaton for verb phrase chunking [7, 12], while Kim used a knowledge-based rules [6]. For the consistency with noun phrase chunking, we use the heuristic rules based on knowledge of Korean verbs. The rules used in this paper are the ones proposed by Kim [6] and the further explanation on the rules is skipped.

3.3 Adverb Phrase Chunking

When the adverbs appear in succession, they have a tendency to form an adverb phrase chunk. Though a adverb sequence is not always an adverb phrase chunk, it usually forms a chunk. Table 1 shows this empirically. The usage of the successive adverbs is investigated from 3,002 sentences collected from Korean textbooks for the elementary school, and 1,271 cases where two adverbs appear in succession are observed in the sentences. Among them, 775 cases form a chunk whereas only the remaining 496 cases do not form a chunk. Thus, it can be told that the possibility that an adverb sequence forms a chunk is far higher than that of not forming. In this paper, the cases where the successive adverbs do not form a chunk are specially handled by the machine learning technique explained below.

Type	Example
B-NP, I-NP & B-NP	[**이 황금**] [**같은** 기회를] [놓치지 마라] ([*this gold*] [*like opportunity*-OBJ] [*miss not*]) ⇒ [이] [황금 같은] [기회를] [놓치지 마라.] ([*this*] [*gold like*] [*opportunity*-OBJ] [*miss not*])
B-NP & B-NP	[그녀는] [**코가**] [**예쁜** 소녀] [이다.] ([*she*-NOM] [*nose*] [*pretty girl*] [*be*]) ⇒ [그녀는] [코가] [예쁜] [소녀] [이다.] ([*she*-NOM] [*nose*] [*pretty*] [*girl*] [*be*])
B-NP & I-NP	[한국에] [있는] [**동안 부모님을**] [뵈었다] ([*Korea*-LOCA] [*stay*] [*while parents*-OBJ] [*visit*]) ⇒ [한국에] [있는 동안] [부모님을] [뵈었다] ([*Korea*-LOCA] [*stay while*] [*parents*-OBJ] [*visit*])
I-NP & I-NP	[그는] [오늘 **아침 바닷가에서**] [떨고 있었다.] ([*he*-NOM] [*today morning beach*-LOCA] [*trembling*]) ⇒ [그는] [오늘 아침] [바닷가에서] [떨고 있었다.] ([*he*-NOM] [*today morning*] [*beach*-LOCA] [*trembling*])
B-ADVP & I-ADVP	[나는] [그를] [**때때로 조용히**] [바라보았다.] ([*I*-NOM] [*he*-OBJ] [*sometimes quietly*] [*watch*]) ⇒ [나는] [그를] [때때로] [조용히] [바라보았다.] ([*I*-NOM] [*he*-OBJ] [*sometimes*] [*quietly*] [*watch*])

Table 2: The types of errors caused by heuristic rules.

4 Enhancing Chunkers Using Heuristic Rules

4.1 Errors of Heuristic Rules

Though the heuristic rules show considerably high accuracy [7, 12], they make some errors since chunking is not a linear problem of simple rules. Table 2 shows the errors which can be caused by the proposed heuristic rules. We classify the errors into five types based on the chunk labels assigned by the rules. For instance, let us consider the first error type, *B-NP*[2], *I-NP & B-NP*. In the example of this type, '이 (*this*)' is labeled to be *B-NP*, '황금 (*gold*)' to be *I-NP*, and '같은 (*like*)' to be *B-NP*, where the correct label of '황금 (*gold*)' is *B-NP* and the correct label of '같은 (*like*)' is *I-NP*. For other error types, Table 2 shows the examples mislabeled by the heuristic rules. The underline in the examples indicates the mislabeled words.

Training algorithm:

- For each training example <x, $f(x)$>, add the example to the list *training_examples*.

Classification algorithm:

- Given a query example x_q to be classified,
  - Let $x_1, \ldots, x_k$ denotes the k examples from *training_examples* that are nearest to x_q.
  - Return

$$\hat{f}(x_q) = \arg\max_{s \in S} \sum_{i=1}^{k} \delta(s, f(x_i)),$$

where $\delta(a, b) = 1$ if $a = b$ and where $\delta(a, b) = 0$ otherwise.

Table 3: The k-NN algorithm for a discrete-valued function $f : R^n \to C$.

To ameliorate the chunking problem, the k-NN algorithm, a machine learning technique supports the heuristic rules by the way of the confidence of the labels assigned by it. In our problem settings, the number of training examples for k-NN in small, because only the examples which are misclassified by the existing rules are used for training. But, the k-NN achieves high accuracy with only a small number of training examples. since its learning is an instance-based method. In addition, because it is a kind of lazy learning method, it shows higher accuracy as the number of training examples increases. Especially, k-NN can pay attention to the particular local problem, so that it is proper for the cases where the problem is further divided into sub-problems. Thus, k-NN is an appropriate learning algorithm for our problem settings, since it is going to be applied only when the heuristic rules are not good enough to determine the chunk labels.

4.2 k-NN to Enhance Exiting Chunkers

The k-NN learning algorithm [4] assumes that all examples correspond to points in the n-dimensional space R^n, where n is the number of features used to discriminate examples. The nearest neighbors of

[2] The chunk tags used in this paper are consistent with those of CoNLL 2000 shared task. There are six tags for chunks, which are *B-NP, I-NP, B-VP, I-VP, B-ADVP, and I-ADVP*, since we consider three types of phrases in chunking Korean. For each phrase, B-CHUNK implies the first word of the chunk and I-CHUNK means each other word in the chunk

Attributes	Explanation
w_i	lexicon of w_i
w_{i-1}	lexicon of w_{i-1}
w_{i-2}	lexicon of w_{i-2}
POS$_i$	POS of w_i
POS$_{i-1}$	POS of w_{i-1}
POS$_{i-2}$	POS of w_{i-2}
E$_{i-1}$	Postposition or Ending of w_{i-1}
R$_{i-1}$	Chunking Label of w_{i-1}
R$_{i-2}$	Chunking Label of w_{i-2}

Table 4: The attributes of k-NN for Korean chunking system.

an example are defined in terms of the standard Euclidean distance. Thus, the distance between two examples x_i and x_j, $D(x_i, x_j)$ is defined to be

$$D(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (v(x_i) - v(x_j))^2}$$

and $v(x_i)$ denotes the value of example x_i.

Let us consider learning a discrete-valued target function of the form $f : R^n \rightarrow C$, where C is the finite set $\{c_1, \ldots, c_s\}$. The k-NN algorithm for this problem is given in Table 3. The value of $\hat{f}(x_q)$ returned as an estimate of $f(x_q)$ is the most common value of f among the k training examples nearest to x_q.

In this paper, five k-NNs are constructed in accordance with Table 2. The training examples for k-NN are only the ones which are not correctly labeled by the heuristic rules. The sentence in the corpus are first labeled by the heuristic rules, and those sentences with mislabeled words are gathered separately and relabeled manually by human experts. Those relabeled sentences are further divided into five sets according to the category of mislabeled words. Each of five k-NNs are trained with each set of relabeled sentences.

We assume that the k-NNs classify only the examples only when they can determine with high confidence the label of a concerned example x_q. That is, the result of k-NN is accepted only when the average distance between x_q and k nearest neighbors is larger than the predefined threshold θ. This can be formulated as

$$\frac{\sum_{i=1}^{k} d(\mathbf{x}_i, \mathbf{x}_q)}{k} \geq \theta.$$

From the viewpoint of k-NN, it determines the label of x_q only when x_q is very similar to the examples whose labels are already known.

Though the local lexical information, in general, may not give reliable linguistic knowledge in Korean, it is very efficient and reliable information in this problem since the examples are considered locally in the chunks. For this reason, we can use n-gram like attributes for k-NN of Korean chunking. We select nine attributes (Table 4). For lexicons, part of speech tags,

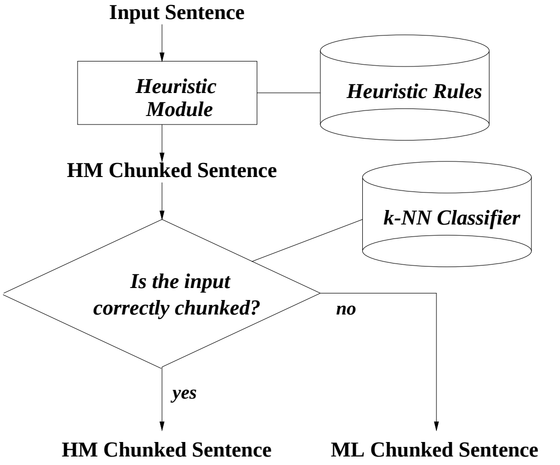


Figure 2: The procedure for Korean chunking system.

and chunk tags, we use *trigram*-like attributes. Three of the attributes, which are w_i, w_{i-1} and w_{i-2}, are lexical features. Since the chunk labels determined by k-NN are used only when the confidence of k-NN is high, we do not use any kind of smoothing methods in comparing lexicons for those features. Another three attributes are the part-of-speech tags of w_i's. E_{i-1} is a postposition or an ending of w_{i-1}. R$_{i-1}$ and R$_{i-2}$ are two previous labels. It is a postposition if w_{i-1} is a noun or a pronoun, whereas it is an ending if w_{i-1} is a verb or an adjective. This attribute is selected because, as shown in Section 3, it holds much information on chunking, especially in noun phrases and verb phrases. Since k-NN measures the similarity in the Euclidean space, it performs best when the attributes are orthogonal. Though the attributes presented may not be orthogonal, k-NN shows an improved accuracy empirically.

4.3 Combined Chunking System

The chunking system combined by heuristic rules and k-NN is shown in Figure 2. For a given input sentence, the heuristic module which uses only the heuristic rules first determines the chunk label of each word. The labels assigned by the rules are reaffirmed by one of five k-NNs. The k-NN to be selected is determined by the current labels assigned by the rules.

Assume that x_q is the word whose label is assigned by the rules. For each x_q in the input sentence, the label of x_q is set by k-NN if the average distance among x_q and k nearest neighbors is larger than the predefined θ. Otherwise, the label of x_q is set by the heuristic rules.

For instance, let us consider the last example in Table 2, where '때때로 (*sometimes*)' is labeled to be *B-NP* and '조용히 (*quietly*)' to be *I-NP* by the heuristic rules. The k-NN which handles the *B-NP & I-NP* problem determines that the label of '조용히 (*quietly*)' is *B-NP* and the average distance is 2.70 where θ is 2.24. Since the average distance is larger than θ, '조

Type	Heuristic Rules		Combined k-NN	
	No. of Errors	Ratio	No. of Errors	Ratio
B-NP, I-NP & B-NP	5	8.48%	4	11.11%
B-NP & B-NP	6	10.17%	3	8.34%
B-NP & I-NP	9	15.25%	4	11.11%
I-NP & I-NP	30	50.85%	17	47.22%
B-ADVP & I-ADVP	9	15.25%	8	22.22%
Total	59	100.00%	36	100.0%

Table 5: The error distribution in the training set.

	Training Set	**Test Set**
Baseline	95.3%	95.5%
Combined k-NN	97.2%	97.8%

Table 6: The experimental result of chunking Korean. The baseline is when we use only heuristic rules, and the combined k-NN is when we use both the rules and k-NNs.

용히 (*quietly*)' is relabeled to be *B-NP*.

5 Experiments

We collected 1,911 sentences from the Internet homepages written in Korean for the experiments. 1,273 sentences among them are used as a training set and the remained 638 sentences are used as a test set. An average length of the sentences is 8.9 *ojeols*[3].

For 1,273 training sentences, the heuristic rules reports 59 errors. Table 5 shows the distribution of each error type in Table 2. It reports that the most errors in the training set occurs in the noun phrase, while no error is reported in the verb phrase. The most difficult type among them is *B-NP & I-NP*, since every noun whose previous noun has no postposition is classified into *I-NP*. The k-NN learns from the misclassified examples that some previous words, e.g. time-related words, forms an independent noun phrase. For adverb phrases, k-NN give much improvement over the heuristic rules. It is usually determined by the lexicons of two adverbs whether two successive adverbs form a chunk or not. Since there are the small number of examples for this case and we do not apply any smoothing technique to comparing lexicons, it is difficult for k-NN to shows great improvement.

Table 6 summarizes the performance of the proposed method. The baseline implies the model which uses only heuristic rules and it shows the accuracy of 95.5%. When we augment the rules with k-NN (noted as 'Combined k-NN' in Table 6), we could achieve the accuracy of 97.8%, which is 2.3% of improvement over the baseline. This result is obtained when k, the number of the nearest neighbor, is 1. The improvement seems to be minute, but such an improvement is meaningful because chunking is in general an intermediate step of full parsing.

Two experiments that demonstrate the effectiveness of the number of the nearest neighbors and the threshold θ are presented. Figure 3 displays the accuracy of the proposed method. Figure 3(a) shows the accuracy curve when $k = 1$, and Figure 3(b) is when $k = 3$. In both figures, x-axis represents the value of θ and y-axis represents the accuracy. The accuracy curve shows the similar aspect regardless of k. As the value of θ increases, the accuracy curve goes up and gets nearly flat after $\theta = 2.2$. When θ is greater than 2.5, the accuracy rather decreases. Such a phenomenon takes place because the number of candidates of which label is determined by k-NN is reduced if θ is too high. The difference between the highest accuracy and the accuracy of baseline is the improvement which we can obtain by using k-NN in addition to the heuristic rules. In Figure 3(a), the highest accuracy is 97.8% when $\theta = 2.5$ and the accuracy of the baseline is 95.5%. Thus, we achieve 2.3% of accuracy improvement.

6 Conclusions

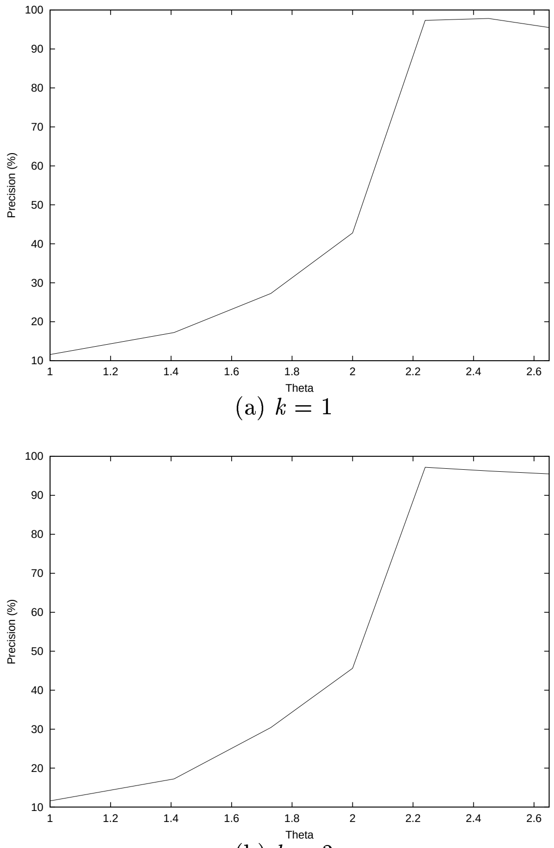
In this paper we presented a novel method to chunk Korean sentences using both heuristic rules and a machine learning algorithm, k-nearest neighbor algorithm. Given a sentence to be chunked, our method first divides it into chunks based on the heuristic rules, and then the chunks are verified by k-NN. When k-NN can determine the chunk label with high confidence, the chunk is determined by k-NN. Otherwise, it is set by the heuristic rules, since the heuristic rules by themselves can achieve high performance. Thus, the use of k-NN enhances the heuristic rules.

The proposed model is evaluated on the sentences gathered from Korean homepages and shows 97.8% of accuracy, which is 2.3% of improvement over the simple method using only heuristic rules. Such a minute improvement is important in chunking, since chunking is an intermediate step of full parsing and the errors in this step can not be recovered in the following steps of language processing.

Acknowledgements

This result was supported in part by the Korean Ministry of Education under the BK21-IT Program and by the Korean Ministry of Information and Communication through IITA under grant 00-023.

[3] The *ojeol* in Korean is the mixture of words and is a basic unit of spacing.



(a) $k = 1$

(b) $k = 3$

Figure 3: The chunking accuracy curve according to θ value.

References

[1] S. Abney, "Parsing by Chunks," In *Principle-Based Parsing*, Kluwer Academic Publishhers, 1991.

[2] S. Argamon, I. Dagan and Y. Krymolowski, "A Memory-based Approach to Learning Shallow Natural Language Patterns," In *Proceedings of COLING/ACL 1998*, pp. 67–73, 1998.

[3] CoNLL, "Shared Task for Computational Natural Language Learning (CoNLL)," 2000. http://lcg-www.uia.ac.be/conll2000/chunking.

[4] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, Vol. 13, pp. 21–27, 1967.

[5] R. Florian, J. Henderson, and G. Ngai, "Coaxing Confidences from an Old Friend: Probabilistic Classifictions from Transformation Rule Lists," In *Proceedings of EMNLP/VLC-2000*, pp. 26–34, 2000.

[6] K.-C. Kim, K.-O. Lee and Y.-S. Lee, "Korean Compound Verbals Processing driven by Morphological Analysis," *Journal of KISS*, Vol. 22, No. 9, pp. 1384–1393, 1995.

[7] M.-Y. Kim, S.-J. Kang and J.-H. Lee, "Dependency Parsing by Chunks," In *Proceedings of the 27th KISS Spring Conference*, pp. 327–329, 1999.

[8] G. Ngai and D. Yarowsky, "Rule Writing or Annotation: Cost-efficient Resource Usage for Base Noun Phrase Chunking," In *Proceedings of ACL 2000*, pp. 117–125, 2000.

[9] L. Ramshaw and M. Marcus, "Text Chunking Using Transformation-Based Learning," In *Proceedings of the Third ACL Workshop on Very Large Corpora*, pp. 82–94, 1995.

[10] Erik T.-K. Sang, "Noun Phrase Representation by System Combination," In *Proceedings of ANLP-NAACL 2000*, pp. 50-555, 2000.

[11] H.-P. Shin, "The VP-Barrier Algorithm for a Robust Syntatic Parsing in Head-Final Languages," In *Proceedings of Natural Language Processing Pacific Rim Symposium*, pp. 475–478, 1999.

[12] H.-P. Shin, "Maximally Efficient Syntatic Parsing with Minimal Resources,", In *Proceedings of the Conference on Hangul and Korean Language Infomration Processing*, pp. 242–244, 1999.

[13] G. Zhou and J. Su, "Error-driven HMM-based Chunk Tagger with Context-dependent Lexicon," In *Proceedings of EMNLP/VLC-2000*, pp. 71–79, 2000.