# NEW PARSING METHOD USING GLOBAL ASSOCIATION TABLE

**Juntae Yoon** and **Seonho Kim** and **Mansuk Song**
{queen,pobi,mssong}@december.yonsei.ac.kr
Department of Computer Science
Yonsei University, Seoul, Korea

**Abstract**

This paper presents a new parsing method using statistical information extracted from corpus, especially for Korean. The structural ambiguities are occurred in deciding the dependency relation between words in Korean. While figuring out the correct dependency, the lexical associations play an important role in resolving the ambiguities. Our parser uses statistical cooccurrence data to compute the lexical associations. In addition, it can be shown that sentences are parsed deterministically by the global management of the association. In this paper, the global association table(GAT) is defined and the association between words is recorded in the GAT. The system is the hybrid semi-deterministic parser and is controlled not by the condition-action rule, but by the association value between phrases. Whenever the expectation of the parser fails, it chooses the alternatives using a chart to remove the backtracking.

## 1 Introduction

The association of words takes an important role in finding out the dependency relation among them. The associations among words or phrases are indicators of the lexical preference. Many works have shown that the association value computed with statistical information makes good results in resolving structural ambiguities(Hindle, 1993; Magerman, 1995; Collins, 1996). Statistical information has led recent researches for syntactic analysis not to the problem of recognizing sentences by given grammar but to that of finding the correct one in multiple parse trees.

A chart parser has been used to produce all possibilities when a sentence is analysed. However, it generates too many structures trying to find a correct one. While reading a sentence, in many cases, a reader can make decisions without examining the entire sentence. A deterministic parser has been worked with the determinism hypothesis for natural language parsing(Marcus, 1980; Faisal et al., 1994). The deterministic parser makes errorneous results because of the limited lookahead, however.

This paper presents a new parsing method that uses the lexical association for parsing sentences semi-deterministically. First, a global association table(GAT) is defined to record and manage the association. As all the associations can be globally observed through the GAT, the parser can obviate the error caused by the limited lookahead. The associations among words are estimated on the basis of lexical association calculated using data acquired from corpus. Next, a parsing algorithm is described using the GAT. The parser selects the action according to the association among the nodes presented by the GAT. That is, the parser is controlled not by condition-action rules, but by the associations between phrases. It merges one phrase with another phrase that has the highest association value, or will wait until it meets the most probable candidate indicated by the GAT. To recapitulate, our system is the parser with the lookahead buffer of sentence length. Experiments show that it doesn't lose accuracy as well as it is as efficient as the deterministic parser.

## 2 The Characteristics of Korean

### 2.1 Structures of Korean sentences

Korean is an agglutinative language and has different features from an inflectional language such as English. A sentence consists of a sequence of *eojeols* composed of a content word and functional words. A content word

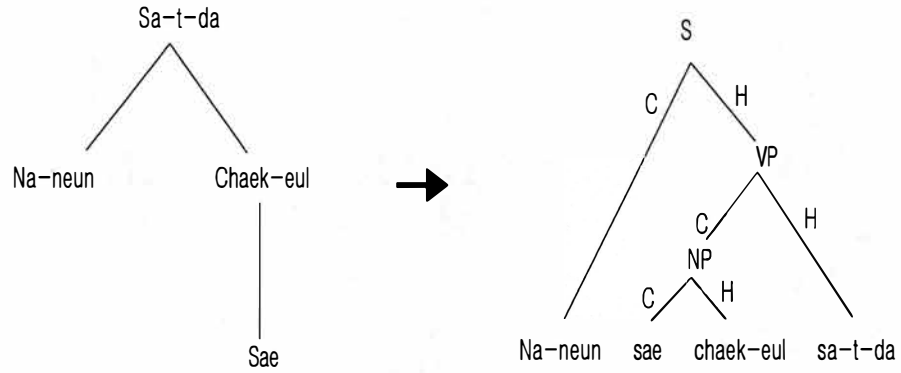


Figure 1: the dependency tree and the binary tree for ex 1)

determines the characterization of a phrase. For instance, an eojeol whose content word is a verb, an adjective, or a copula, functions as a predicate. A functional word directs the grammatical function of an eojeol. For examples, '*eul/reul*' is the postposition(functional word) that makes a nominal eojeol an object. '*Seupnika*' is the sentential ending for a question and '*t/eot*' converts a predicate to the past tense in Korean.

ex 1) *Na-neun sae chaek-eul sa-t-da.*
I new book bought.
— I bought a new book.

'*Na-neun*' is the subject of the above sentence, and '*chaek-eul*', the object.

Second, Korean is an SOV('*Subject Object Verb*' order) language, where the head always follows its complements. In Korean, a head eojeol follows its complement eojeols. A new phrase is generated when one or more eojeols are merged, and the head of the phrase is always the last eojeol of the phrase.

ex 2) *Keu-ga norae-reul booreu-myu hakkyo-e ga-t-da.*
He a song singing to school went
— He went to school singing a song.

Both '*booreu-myu*' and '*ga-t-da*' have verbs as their content words. Predicative eojeols are the heads for nominal eojeols and follow the eojeols, '*keu-ga*', '*norae-reul*', and '*hakkyo-e*' , respectively.

Third, the grammatical dependency relation is determined decisively by functional words. For example, '*in the box*' in English can modify both a noun and a verb. In contrast, in order that '*sangja*', which means *box*, modifies a noun, it has to have the postposition, '*ui*' in Korean. Whenever it has another postpositions, it is the complement of a verb. There is syntactic levels in Korean, and the dependency relation of an eojeol is fixed according to the level.

## 2.2 Syntactic analysis

The dependency tree of a sentence is built up through parsing. For the operation of the deterministic parser such as CREATE, the dependecy tree is represented by the binary tree for phrase structure that has two children nodes for a complement and a head. The complement node is the dependent node of the dependency tree and the head node is the head of the dependent node. Consequently, the parser uses binary grammar described with the feature structure about the morphemes that constitute an eojeol. The parent node inherits the feature of its head node. Since the head follows its complement in Korean, the root node of the parse tree inherits the feature of the last eojeol in the sentence.

(Figure 1) shows the dependency tree and our parse tree of the sentence given in example 1. '*Na-neun*(I)' is a nominal eojeol and dependent on the predicative eojeol, '*sa-t-da*(bought)'. The feature of '*sa-t-da*' is placed in the root node of the parse tree because it is the head of the sentence.

| | |
|---|---|
| saengsan/NN(production) | soodan/NN(method) |
| dambae/NN(tobacco) | nongsa/NN(farming) |
| sooyo/NN(demand) | byunhwa/NN(variation) |
| france/NN(France) | moonhak/NN(literature) |

Figure 2: Examples of co-occurrence pairs of two nouns

# 3 Global Association Table

The global association table(GAT) has the association between words or eojeols that have dependency relation. The association can be estimated in various ways; for example, if it is likely that a word depends on the word nearest to it, the estimation function would be given as follows.

$$Assoc(\epsilon_i, \epsilon_j) = 1/d$$

Let the row and the column of the GAT represent eojeols occurring to the left-hand side and to the right-hand side, respectively, in the parsing process. The left-hand side eojeol is a complement, and the right-hand side, the candidate for its head. $GAT(i, j)$ indicates the degree of association in case the $i$th eojeol is dependent on the $j$th eojeol. Because the head follows its complement in Korean, and the table is a triangular matrix.

## 3.1 Estimation function for association

Two kinds of co-occurrence data were extracted from 30 million eojeol corpus. One is for compound noun analysis, the other is for dependency analysis of verb and noun. The associations of modifier-head relations such as an adverb and a verb, or a pre-noun and a noun, are estimated by distance. Distance measure is also used for the case there is no co-occurrence data, which is caused by data sparseness. The distance has been shown to be the most plausible estimation method without any linguistic knowledge(Collins, 1996; Kurohashi et al., 1994). First of all, co-occurrence pairs of two nouns were collected by the method presented in (Pustejovksy et al., 1993). Let $N$ be the set of eojeols consisting of only a noun and $NP$ the set of eojeols composed of a noun with a postposition. From $\epsilon_1, \epsilon_2, \epsilon_3$ ($\epsilon_1 \notin N$, $\epsilon_2 \in N, \epsilon_3 \in NP$), we can obtain complete noun compounds, $(n_2, n_3)$ such that $n_2$ and $n_3$ are the nouns that belong to the eojeols, $\epsilon_2$ and $\epsilon_3$, respectively. The parser analyzes compound nouns, based on the complete noun compounds. (Figure 2) shows an example of compound noun pairs.

The association between nouns is computed using the co-occurrence data extracted by the above method. Let

$$N = \{n_1, \ldots, n_m\}$$

$N$ be the set of nouns. Given $n_1, n_2 \in N$, association score, $Assoc$, between $n_1$ and $n_2$ is defined to be

$$\begin{aligned} Assoc_{NN}(n_1, n_2) &= P(n_1, n_2) \\ &= \frac{freq(n_1, n_2)}{\sum_i \sum_j freq(n_i, n_j)} \end{aligned} \quad (1)$$

As mentioned above, the distance measure is suggested without any cooccurrence data. Therefore, these estimators are sequentially applied for two eojeols in the following way. Here, $\epsilon_i$ and $\epsilon_j$ are the $i$th and the $j$th nominal eojeols, and $n_i$ and $n_j$ the nouns that belongs to the nominal eojeols.

$$\begin{aligned} &If\ Assoc_{NN}(n_i, n_j) \neq 0 \\ &\quad Assoc(\epsilon_i, \epsilon_j) = Assoc_{NN}(n_i, n_j) \\ &else\ Assoc(\epsilon_i, \epsilon_j) = 1/d \end{aligned}$$

Because the associations are calculated and compared for all $\epsilon_j$ on which $\epsilon_i$ have the possiblity to be dependent, the compound noun analysis is based on the dependency model rather than the adjacency model(Kobayasi et al., 1994; Lauer, 1995). Because the two estimate functions are used, the extra-comparison routine is required. It will be explained in the next section.

Second, the co-occurrence pairs of nominal eojeols and predicative eojeols were extracted by the partial parser from a corpus. (Figure 3) shows an example of the triples generated from the text. In (Figure 3), the triple,

| | | |
|---|---|---|
| gada/VB(go) | gajok/NN(family) | ga(SUBJ) |
| gada/VB(go) | keu/PN(he) | ga(SUBJ) |
| gada/VB(go) | kang/NN(river) | e(TO) |
| masida/VB(drink) | maekjoo/NN(beer) | reul(OBJ) |
| masida/VB(drink) | mool/NN(water) | reul(OBJ) |
| masida/VB(drink) | neo/PN(he) | ga(SUBJ) |

Figure 3: Examples of triples extracted from the text

⟨*masida*/VB, *mool*/NN, *reul*/OBJ⟩ indicates that the verb, '*masida*', and the noun, '*mool*' which mean 'drink' and 'water' respectively, co-occur under the grammatical circumstance, '*reul*' which is the postposition that makes a noun an object.

The association between a verb and a noun is evaluated based on the triples obtained by the above method. Let

$$V = \{v_1, \ldots, v_l\},\ N = \{n_1, \ldots, n_m\}$$
$$S = \{ga, reul, e, \ldots\}$$

$V, N, S$ be the sets of predicates, nouns and syntactic relations respectively. Given $v \in V, s \in S, n \in N$, association score, $Assoc$, between $v$ and $n$ with syntactic relation $s$ is defined to be

$$Assoc_{VN}(n, s, v) = \lambda_1 P(n, s|v) + \lambda_2 P(s|v) \quad (\lambda_1 \gg \lambda_2) \tag{2}$$

The conditional probability, $P(n, s|v)$ measures the strength of the statistical association between the given verb, $v$, and the noun, $n$, with the given syntactic relation, $s$. That is, it favors those that have more co-occurrences of nouns and syntactic relations for verbs. However, the above formula, including the maximum liklihood estimator, suffers from the problem of data sparseness. To back off the estimation, it is introduced the probability, $P(s|v)$ that means how much the verb requires the given syntactic relation.

The association measure based on the distance between two eojeols is used without any co-occurrence data. These estimators are applied sequentially in the following way. Let us suppose that $\epsilon_i$ be the $i$th eojeol and $\epsilon_j$ the $j$th eojeol. In addition, $n_i$ and $s_i$ are the noun and the postposition in the nominal eojeol, $\epsilon_i$, and $v_j$ the verb in the predicative eojeol, $\epsilon_j$, respectively.

$$If\ Assoc_{VN}(n_i, s_i, v_j) \neq 0$$
$$Assoc(\epsilon_i, \epsilon_j) = Assoc_{VN}(n_i, s_i, v_j)$$
$$else\ Assoc(\epsilon_i, \epsilon_j) = 1/d$$

## 3.2 Making GAT

The association value of two eojeols is recorded in the GAT only when the eojeols have dependency relation. Above all, the dependency relation of two eojeols is checked, therefore. For two eojeols to have dependency relation indicates that they have a possibility to be combined in parsing process. For example, a nominal eojeol with the postposition for case mark depends on a predicative eojeol that follows them. Second, if a dependency relation can be assigned to two eojeols, the association value is calculated using the estimators described in the previous section.

The association is represented by a pair, ⟨*method, association-value*⟩. If a sentence consists of $n$ eojeols, the GAT used is the $n \times n$ triangular matrix. As mentioned in the previous section, each eojeol has its own syntactic level in Korean, and an eojeol can be combined with either a predicate or a noun. This follows that an eojeol doesn't have dependency relation to the nominal eojeol, whenever it is dependent on the predicative eojeol, *vice versa*. Because the different estimators are applied for the analysis of compound noun and predicate-argument, any collision doesn't take place in the comparison of the association. $Assoc_{NN}$ is used as the estimator for compound noun and $Assoc_{VN}$, for predicate-argument. The GAT is sorted by the association to look up the most probable phrase in the parsing process. Thus, the global association table is implemented by the global association list. The algorithm to generate the GAT is represented in (Figure 4).

for each eojeol $\epsilon_i$ $0 <= i <= n-2$

1. for each eojeol $\epsilon_j$ $i+1 <= j <= n-1$

if (depend_on $(\epsilon_i, \epsilon_j)$)

compute $g_i(j) = < method, Assoc(\epsilon_i, \epsilon_j) >$

2. sort $g_i(i+1), \ldots, g_i(n-1)$ and refer it to $G(i)$

Figure 4: The algorithm for making GAT

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 | - | (2,0.1) | (1,1/2) | - | (1,1/3) | - | - |
| 1 | - | - | (1,1) | - | (1,1/2) | - | - |
| 2 | - | - | - | (2,0.11) | - | (1,1/2) | (2,0.02) |
| 3 | - | - | - | - | (2,0.15) | - | - |
| 4 | - | - | - | - | - | (1,1) | (2,0.52) |
| 5 | - | - | - | - | - | - | (1,1) |
| 6 | - | - | - | - | - | - | - |

Table 1: The global association table(GAT) for the example sentence, ex 3

The following example is represented by (Table 1),

ex 3) (0)*computer* (1)*hwamyon-ui* (2)*gusuk-e* (3)*natana-n* (4)*sutja-ga* (5)*paru-ge* (6)*olaga-t-da.*
(0)computer (1)of screen (2)in the corner (3)appeared (4)the number (5)fast (6)scrolled up
— The number to appear in the corner of computer screen scrolled up fast.

In (Table 1), '-' mark means that two eojeols have no dependency relation. The first element of the pair is the method of the measurement and the second is the association value. The pair, (1, 1/2) in GAT(0,2), indicates that the measure by distance is 1/2. The pair (2, 0.11) in GAT(2,3) means that the association value is 0.11 and estimated with co-occurrence relation. The *method* has the priority for the comparsion of the association. Therefore, (2,0.02) is greater than (1,0.5) because *method* of the first is greater than that of the second. Since the row of the table is sorted for parsing, GAT[2] can be represented in the form of a list of eojeols as follows.

GAT[2] — ⟨3,(2,0.11)⟩ — ⟨5,(2,0.02)⟩ — ⟨6,(1,1/2)⟩

The association list in the above lets the parser know that the eojeol $\epsilon_2$ has the possibility to merge with the eojeol, $\epsilon_3$, $\epsilon_5$ or $\epsilon_6$, and the most probable one is $\epsilon_3$. The function, $max(G(i))$ is defined to return the most probable candidate for the head of the $i$th eojeol, $\epsilon_i$, in the GAT.

# 4 Parsing Algorithm

## 4.1 Parsing algorithm

The parser presented here consists of a stack and a buffer. A two-item lookahead buffer is enough to make decisions in regard to Korean. The grammatical structures lie in the parsing stack and a set of actions are operated on the buffer. Unlike the deterministic parser where the set of rules directs the operation, this system parses by the association value of the GAT.

Since a head follows its complement in Korean, the head of a phrase is the last eojeol of the phrase. A phrase is generated when two eojeols or two phrases are merged. In this case, Head Feature Inheritance takes care of the assignment of the same value as the head feature. Suppose an eojeol, $\epsilon_1$, and an eojeol, $\epsilon_2$, merge and a new phrase $P_1$ be generated, as shown in (Figure 5). As the head of $P_1$ is $\epsilon_2$, the parser uses the subscription of $\epsilon_2$ as the index to the GAT, that is, 2.

Basic operations are CREATE, ATTACH, and DROP. However, its operation is conditioned not by rule matching but by the value of the GAT as shown in the following description. The function, $position(max(G(i)))$ returns the sentential position of the most probable candidate for the head of the $i$th eojeol, $\epsilon_i$.

**CREATE** If the most probable candidate for the head of the eojeol, $\epsilon_i$, is $\epsilon_j$, that is, $j = position(max(G(i)))$,

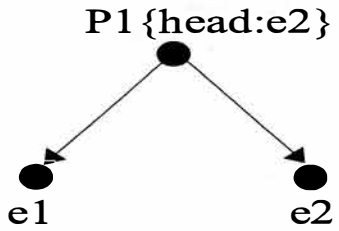


Figure 5: the index to which the parent node refers

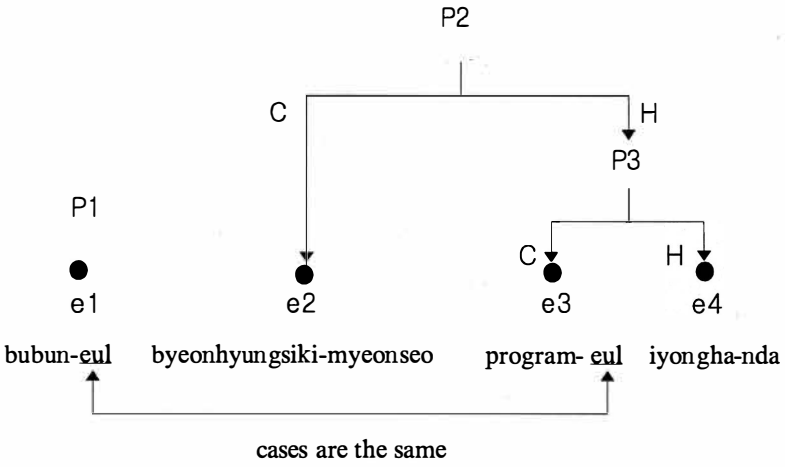


Figure 6: example of failure of the prediction

then merge $\epsilon_i$(or the phrase where the last eojeol is $\epsilon_i$) with $\epsilon_j$(or the phrase where the last eojeol is $\epsilon_j$), and generate a new phrase.

**ATTACH** If the phrase where the $\epsilon_j$ is the last eojeol is not the most probable candidate for the head of the eojeol $\epsilon_i$, that is, $\epsilon_j \neq max(G(i))$ then wait until $\epsilon_i$ meets the most probable candidate indicated by the GAT.

**DROP** DROP operation is accompanied with CREATE operation in our system because the complement precedes the head and thus the top node of the stack must be dropped and checked for dependency immediately after a new node is generated.

The GAT provides the parser with the prediction of the best candidate for the head of the $i$th eojeol, $\epsilon_i$. This is easy because the GAT is already sorted; however, the expectation is not always correct because the value of the GAT is calculated whenever there is a possible dependency relation between one eojeol and another. That is, the parser constructs the GAT as preparsing and it may happen that the two eojeols or phrases which have the possiblity to have dependency relations cannot be merged in parsing. The violation of the '*one case per clause*' principle, is the case.

ex 4) *bubun-eul*(part/OBJ) *byeonhyeongsiki-myeonseo*(change) *program-eul*(program/OBJ)
*iyongha-nda*(use).
→ The part being changed, the program is used.

In (Figure 6), $\epsilon_1$ and $\epsilon_3$ are nominal eojeols, and $\epsilon_2$, $\epsilon_4$ are predicative eojeols apiece. The phrase $P_1$ consists of $\epsilon_1$, and the phrase $P_2$ consists of three eojeols, $\epsilon_2$, $\epsilon_3$, $\epsilon_4$. Let the most probable candidate, suggested by the GAT, for both $\epsilon_1$ and $\epsilon_3$ be $\epsilon_4$. However, $\epsilon_1$ and $\epsilon_3$ have the same grammatical case because they contain

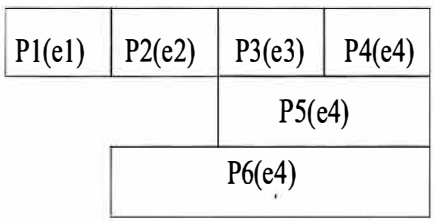


Figure 7: the content of chart and selection for the next candidate

```
For phrases P_i and P_j, let their heads ε_i and ε_j respectively.
if (lookahead = nil and there is one parse tree in the stack)
        return SUCCESS
else if (GAT(●) = NULL)
        return FAIL
else
        if (position(G(i)) = j)
        begin
                if (isunifiable(P_i, P_j) = TRUE)
                        CREATE;
                else ALTER;
        end
        else ATTACH;
```

Figure 8: The parsing algorithm using the GAT

the postpositions marking the same case. The phrase $P_1$ and the phrase $P_2$ cannot be merged because of the violation of '*one case per clause*' principle. This means the prediction of the GAT is incorrect, and consequently an analysis with the alternatives is required. If the next candidate is $\epsilon_2$, the grammatical structure in the buffer must be erroneous. The chart presents the phrase suitable for the alternative execution into the buffer. The chart allows the parser to store the partial structure to remove the backtracking. ALTER operation occurs in this case.

**ALTER** is required if an eojeol $\epsilon_i$ cannot be merged with the eojeol $\epsilon_j$ which is the prediction of the candidate for the head of $\epsilon_i$.

The operation being executed, the structure in the lookahead buffer is backed up into the chart. When ALTER operation is needed, another candidate taken from the chart, has to be put in the buffer. The next candidate, $C(\epsilon_i)$ is chosen in the following way. Let $i$ be the left-hand position and $k$ the right-hand position of the errorneous prediction in the GAT.

$$C(\epsilon_i) = \text{the phrase that the left-hand position is } i \text{ and the right-hand position is } max(i+1 \leq j \leq k) \text{ in nodes in the chart.}$$

Then, the phrase $P_2$ including $\epsilon_2$ is the next candidate in (Figure 7). The parsing algorithm with the GAT is described in (Figure 8).

## 4.2 Parsing

The complexity of making the GAT is $O(N^3 log_2(N))$, where $N$ is the number of eojeols. This is due to the sorted $n \times n$ table. The average complexity of the parser is linear, according to the experiments.

| | OP | Stack Top | | First Lookahead | |
|---|---|---|---|---|---|
| | | Constituents | Head | Constituents | Head |
| 1 | A | (computer) | computer | (hwamyon-ui) | hwamyon-ui |
| 2 | B | (computer hwamyon-ui) | hwamyon-ui | (han) | han |
| 3 | A | (han) | han | (gusuk-e) | gusuk-e |
| 4 | A | (computer hwamyon-ui) | hwamyon-ui | (han gusuk-e) | gusuk-e |
| 5 | A | ((computer hwamyon-ui) (han gusuk-e)) | gusuk-e | (natana-n) | natana-n |
| 6 | A | (((computer hwamyon-ui) (han gusuk-e)) natana-n) | natana-n | (sutja-ga) | sutja-ga |
| 7 | B | ((((computer hwamyon-ui) (han gusuk-e)) natana-n) sutja-ga) | sutja-ga | (paru-ge) | paru-ge |
| 8 | A | (paru-ge) | paru-ge | (olaga-t-da) | olaga-t-da |
| 9 | A | ((((computer hwamyon-ui) (han gusuk-e)) natana-n) sutja-ga) | sutja-ga | (paru-ge olaga-t-da) | olaga-t-da |

Figure 9: an example of analyzing the sentence in (ex 3). (A) Create & Drop operation (B) Attach operation

| | Chart Parser first S found | Parser Using GAT |
|---|---|---|
| The Total Number of Generated Nonterminals | 2,561,613 | 10,582 |
| The Average Number of Generated Nonterminals | 6404 | 26.5 |

Table 2: The number of nodes generated by test parsers

(Figure 9) represents the analysis steps of the sentence in (ex 3). The head on the stack top is the complement, and the candidate for the head of it lies in the head part of the buffer. In the seventh row of the figure, the ATTACH operation is executed by the GAT in (Table 1), because the lookahead is not the best candidate for the head of the complement on the stack top. The eojeol, 'sutja-ga', has to wait until it meets its best candidate. A new phrase are created in the row (9). The eojeol, 'olaga-t-da' is the best candidate for the eojeol, 'sutja-ga', which was estimated by the GAT. (Figure 10) represents the parse tree of ex 3). The sentence is written in Korean.

# 5 Experimental Results

For testing purposes, 400 sentences were randomly extracted from 3 million corpus. First, our parser is compared to the chart parser to show the efficiency of our algorithm. The number of the nodes generated by each parser is represented in (Table 2). Because of ths size of the searching space, the results from the chart parser are calculated whenever the first S is found. The average number of the prediction failure is 0.26 per sentence. That is, The parser has to search for the alternative in the chart once in four sentences. This makes the complexity of the parser a constant. (Figure 11) shows the occurrence of ALTER operation over the number of words. The average number of ALTER is about 0.36 for the sentences with more than 20 words, which means our parser is efficient.

Second, the precision is given in (Table 3). The precision is defined as the ratio of the precise dependency relation between eojeols in parse trees. No label is attached because the final output is the tree that represents the dependency relation among words. Thus, the number of erroneous and correct relations is considered, which can be estimated by the number of crossing brackets(Table 3).

**Crossing Brackets** number of constituents which violate constituent boundaries with a constituent in the correct parse.

The cause of the incorrect analysis can be largely classified by two reasons. One of the failures is caused by statistical information. We collected the data from 30 million eojeol corpus. The total number of the data is 2

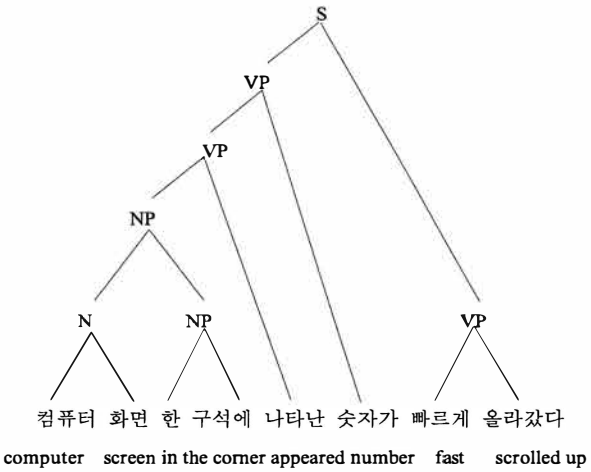


Figure 10: the parse binary tree for ex 3) (in Korean)

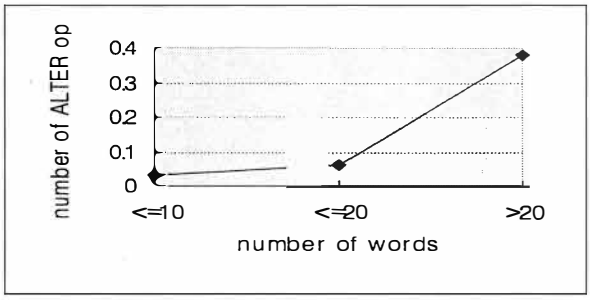


Figure 11: The number of the ALTER operation for words

| CBs | | 0 CBs | | $\leq$ 2 CBs | |
|---|---|---|---|---|---|
| (a) | (b) | (c) | (d) | (c) | (d) |
| 378 | 0.95 | 176 | 44.0 | 323 | 80.8 |

Table 3: The precision of the parser (a) the number of crossing brackets (b) the average number of crossing brackets per sentence (c) the number of sentences (d) the percentage

million and the average frequency of the co-occurrence data is 2.5. The triples to have frequencies greater than 2 are 400,000. The frequency of most data is 1, which was the cause of the errorneous results. In addition, the association value of adjuncts and subordinate clauses is estimated by distance. The distance estimator was good but not the best. Semantic information such as thesaurus will help reduce the space of parameters.

Second, liguistic information is needed , e.g., such as light verbs or the lexical characteristics of individual words. Our parser is the hybrid system which uses both rules and statistical information. The linguistic research is prerequired for this, even if these can be partially resolved by statistical methods. However, the parser is satisfactory in spite of some erroneous results in that the association value can be computed in various ways, and the parser can be extended using this.

# 6 Conclusion

We have shown that it is possible to make decision semi-deterministically using the global association table. The GAT is a very effective structure in that it is triangular matrix because Korean is an SOV language and the dependency relations between words is important. It would have to be transformed for parsing English, because phrase structure grammar is needed for parsing English.

There are many possiblities for improvement. The method described for calculating the lexical association in the GAT can be modified in various ways. The GAT and the parser can be extended if the distance measure and the coordinate conjunctive structure are considered.

# References


Allen, J. (1995). *Natural Language Understanding.* Benjamin Cummings.

Brill, E. (1993). *A Corpus-Based Approach to Language Learning* Department of Computer and Information Science, University of Pennsylvania.

Briscoe, T., Waegner, N. (1992). *Robust Stochastic Parsing Using the Inside-Outside Algorithm* In *Workshop notes from the AAAI statistically-based NLP Techniques Workshop.*

Charniak, E. (1993). *Statistical Language Learning* MIT Press.

Collins, M. J. (1996). *A New Statistical Parser Based on Bigram Lexical Dependencies* In *Proceedings of 34th Annual Meeting of Association for Computational Linguistics.*

Faisal, K. A. and Kwasny, S. C. (1990). *Design of a Hybrid Deterministic Parser* In *Proceedings of COLING-90.*

Framis, F. R. (1994). *An Experiment on Learning Appropriate Selectional Restrictions from a Parsed Corpus.* In *Proceedings of COLING-94.*

Gazdar G., and Mellish C. (1993). *Natural Language Processing in LISP.* Addison Wesley.

Hindle, D. and Rooth, M. (1993). *Structural Ambiguity and Lexical Relations* Computational Linguistics

Kobayasi Y., Tokunaga T., and Tanaka H. (1994). *Analysis of Japanese Compound Nouns using Collocational Information* In *Proceedings of COLING-94.*

Kurohashi S. and Nagao M. (1994). *A Syntactic Analysis Method of Long Japanese Sentences Based on the Detection of Conjunctive Structures.* Computational Linguistics

Lauer, M. (1995). *Corpus Statistics Meet the Noun Compound: Some Empirical Results* In *Proceedings of 33rd Annual Meeting of Association for Computational Linguistics.*

Marcus, M. (1980). *A Theory of Syntactic Recognition for Natural Language.* Cambridge, MA: MIT Press.

Magerman, D. M. (1995). *Statistical Decision-Tree Models for Parsing.* In *Proceedings of 33rd Annual Meeting of Association for Computational Linguistics.*

Pustejovsky, J., Bergler, S., and Anick, P. (1993). *Lexical Semantic Techniques for Corpus Analysis.* Computational Linguistics

Resnik, P. (1992). *Wordnet and Distributional Analysis: A Class-Based approach to Lexical Discovery* In *Proceedings of AAAI Workshop on Statistical Methods in NLP.*

Tomita, M. (1986). *Efficient Parsing for Natural Language* Boston: Kluwer Academic Publishers