English Syntactic Disambiguation Using Parser's Ambiguity Type Information

Jae Won Lee, Sung-Dong Kim, Jinseok Chae, Jongwoo Lee, and Do-Hyung Kim

This paper describes a rule-based approach for syntactic disambiguation used by the English sentence parser in E-TRAN 2001, an English-Korean machine translation system. We propose Parser's Ambiguity Type Information (PATI) to automatically identify the types of ambiguities observed in competing candidate trees produced by the parser and synthesize the types into a formal representation. PATI provides an efficient way of encoding knowledge into grammar rules and calculating rule preference scores from a relatively small training corpus. In the overall scoring scheme for sorting the candidate trees, the rule preference scores are combined with other preference functions that are based on statistical information. We compare the enhanced grammar with the initial one in terms of the amount of ambiguity. The experimental results show that the rule preference scores could significantly increase the accuracy of ambiguity resolution.

Manuscript received May 16, 2002; revised Jan. 18, 2003.

This work was financially supported by Hansung University in the year of 2002.

Jae Won Lee (phone: 82 2 920 7610, email: jwlee@cs.sungshin.ac.kr) and Do-Hyung Kim (email: dkim@cs.sungshin.ac.kr) are with School of Computer Science and Information, Sungshin Women's University, Seoul, Korea.

Sung-Dong Kim (email: sdkim@hansung.ac.kr) is with Department of Computer System Engineering, Hansung University, Seoul, Korea.

Jinseok Chae (email: jschae@incheon.ac.kr) is with Department of Computer Science and Engineering, University of Incheon, Incheon, Korea.

Jongwoo Lee (email: jwlee44@daisy.kw.ac.kr) is with Department of Computer Engineering, Kwangwoon University, Seoul, Korea.

I. Introduction

E-TRAN 2001 [1] is an English-Korean machine translation system developed for domain-independent translation that requires both broad coverage and high accuracy. Increasing coverage usually also increases the number of parse trees for sentences previously covered and results in a lower accuracy for these sentences. We address two issues to increase both parsing coverage and accuracy. The first aims to reduce ambiguity by managing grammar rules in a more efficient way or improving parsing technology. The other aims to use rational criteria for sorting candidate trees in a preference order. Reference [2] reduced ambiguity using constraint functions that prevent a structure from being built for a given syntactic context. However, it was not clear which kinds of structures could be prevented without any loss of coverage. The study in [3] also tried to reduce the amount of ambiguity using strong constraints. Given a fixed amount of ambiguity, the accuracy of ambiguity resolution ultimately depends on an estimation function (in probabilistic approaches) or a preference function (in rule-based approaches). The problem of ambiguity resolution is also important in the area of speech recognition [4].

Many earlier probabilistic approaches used less constraining grammars to increase coverage and relied on an estimation function based on the probabilities of constituents to choose the most likely interpretation. They usually learned statistical parameters automatically from tagged corpora [5], [6]. However, the variety of parse types generated by these systems was limited and creating the requisite training corpus was difficult. Probabilistic parsers combined with hand-coded linguistically fine-grained grammars have seen considerable progress in recent years [7], [8]. However, such attempts have so far been confined to relatively small-scale applications.

Rule-based parsers generally use a preference function for ambiguity resolution to rank competing candidate analyses, but when applied to large-scale applications, they usually fail to offer satisfactory performance because it is quite difficult to acquire and manage reasonable preference functions. Wang [9] tried to associate the syntactic preference function, first described in [10], with the semantic preference functions. This attempt apparently failed to achieve a practical performance for open domain applications [11], [12]. One remarkable study by Alshawi et al. on integration of various preference functions [13] encouraged the development of a more practical analysis system. In particular, that study proved that the notion of *mean distance* for the evaluation of lexical collocation preference functions, which considers frequencies in badly parsed trees, was very effective.

Extending the mean distance method to a syntactic preference function, we propose Parser's Ambiguity Type Information (PATI) as a new way of coping with ambiguity in rule-based natural language analysis. PATI is a weighted, directed graph that represents the differences in applied grammar rules among candidate trees. In PATI, the directions of edges represent priority relations among rule sets and the weights represent the frequencies of those relations. It can identify the target of syntactic disambiguation more definitely and provide helpful information for designing and implementing a strategy for disambiguation. E-TRAN 2001 uses a general chart parser with a grammar formalism based on the Generalized Phrase Structure Grammar [14]. PATI is automatically constructed using information extracted from candidate trees, one of which is marked as the correct one with its constituent structure. PATI guides the hand tuning of the initial grammar to reduce the amount of ambiguity, which could considerably save the human efforts in the tuning by providing clues about the essential knowledge to be encoded into the rules. PATI is then used to calculate the rule preference scores that are based on the frequency information of the rules. The scoring function is different from those in previous studies in that it uses the rule frequencies from all the candidate trees produced by the system, not only from the best tree. Experimental results show that PATI is useful for developing a large-scale grammar and for identifying various kinds of ambiguity types. It also maintains the accuracy of the ambiguity resolution.

The rest of the paper is organized as follows:

Section II: Definition and construction of PATI
Section III: The grammar tuning process
Section IV: The rule preference function
Section V: Experimental results
Section VI: Conclusion and future work

II. Overview of PATI

1. Definition of PATI

We start with preliminary definitions for comparing candidate trees of a sentence.

Definition 1. Let t_1, t_2, Λ, t_n be n candidate trees produced by analyzing a sentence s; let R_k be a multiset of rules applied for building $t_k (1 \le k \le n)$; and let $t_c (1 \le c \le n)$ be the correctly parsed tree. *Rule set difference* D_j^i is defined as $D_j^i = (R_i - R_j)$ [1)] and *priority pair* P_i^c is defined as $P_i^c = (D_c^i, D_i^c)$, where $i \ne j, c \ne i$ and $1 \le i, j \le n$. Finally, the *priority pair set* of s, $PS(s)$, is defined as the set of $n-1$ priority pairs and the *difference set* $DS(s)$ as the set of $2(n-1)$ rule set differences.

For example, let us examine a famous sentence with ambiguities.

s_1: Time flies like an arrow.

Figure 1 shows two candidate trees generated from analyzing the above sentence.

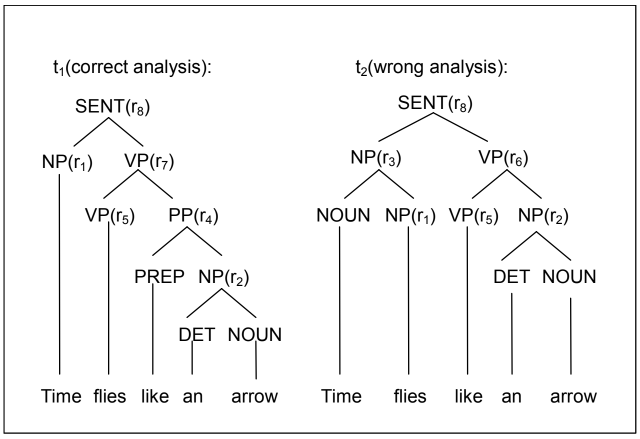

Fig. 1. Candidate trees of s_1.

The rules applied to analyze a sentence s_1 are as follows:

r_1: NP $\rightarrow$ NOUN	r_2: NP $\rightarrow$ DET NOUN
r_3: NP $\rightarrow$ NOUN NP	r_4: PP $\rightarrow$ PREP NP
r_5: VP $\rightarrow$ VERB	r_6: VP $\rightarrow$ VP NP
r_7: VP $\rightarrow$ VP PP	r_8: SENT $\rightarrow$ NP VP

R_1 is the rule set for the candidate tree t_1 and R_2 is for t_2, so $R_1=\{r_1, r_2, r_4, r_5, r_7, r_8\}$ and $R_2=\{r_1, r_2, r_3, r_5, r_6, r_8\}$. The rule set differences are $D_2^1 = \{r_4, r_7\}$ and $D_1^2 = \{r_3, r_6\}$. The priority pair is $P_2^1 = (\{r_3, r_6\}, \{r_4, r_7\})$, the priority pair set is

1) In this paper, the symbol '-' denotes the difference set of two multisets. The difference set A-B contains elements of A whose multiplicity in A is larger than that in B. The multiplicity of matching elements is the difference between the multiplicities in A and B.

$PS(s_1) = \{P_2^1\}$, and the difference set is $DS(s_1) = \{D_2^1, D_1^2\}$.

Definition 2. Suppose we analyze a corpus *C* using a rule set *R*. The *priority relation graph* is a directed, weighted graph $G = (V, E)$, where $V = \mathrm{Y}_{s\in C} DS(s)$, $E = \mathrm{Y}_{s\in C} PS(s)$, and the weight w of an edge is the frequency at which the edge appears in the analyses of *C*.

Figure 2 shows a priority relation graph of s_1 when the frequency of the priority pair is one.

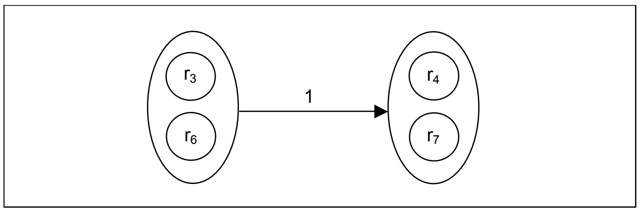

Fig. 2. Priority relation graph of s_1.

Though a priority relation graph can represent types of ambiguity, it includes some redundant information. For example, let's consider the following sentences.

$s_{2,1}$: I know that you are happy.
$s_{2,2}$: He sees sleeping babies.
$s_{2,3}$: I ate a fish with bones.
$s_{2,4}$: I know that it contains operating systems for my PC.

Figure 3 shows the priority pairs of the above sentences. The rules applied to analyze sentences from $s_{2,1}$ to $s_{2,4}$ are as follows:

r_1: RLCL $\rightarrow$ SENT	r_2: NP $\rightarrow$ NP RLCL
r_3: VP $\rightarrow$ VP NP	r_4: RLCL $\rightarrow$ PRON SENT
r_5: VP $\rightarrow$ VP RLCL	r_6: VP $\rightarrow$ VP PRESP
r_7: NP $\rightarrow$ PRESP NP	r_8: VP $\rightarrow$ VP PP
r_9: NP $\rightarrow$ NP PP	

In Fig. 3, four priority pairs from (a) to (d) result from the analyses of the sentences from $s_{2,1}$ to $s_{2,4}$, respectively. The difference sets of priority pairs in (a) to (c) also appear in (d). The priority pair (d) can be regarded as the combination of the three priority pairs (a) to (c). To get a more compact representation of ambiguity types, it is desirable to remove edges and vertices like (d). For this, we need some more definitions.

Definition 3. Let $e_i = (v_1^i, v_2^i)$ and $e_j = (v_1^j, v_2^j)$ be two distinct edges of a priority relation graph. If $v_1^i \subseteq v_1^j$ [2)] and $v_2^i \subseteq v_2^j$, then e_i is defined to *subsume* e_j, which is denoted as $e_i \ \pi \ e_j$.

[2)] In this paper, the symbol '$\subseteq$' denotes the subset relation between multisets. Multiset A is a subset of multiset B if the multiplicity of matching elements in B is greater than or equal to their multiplicity in A.

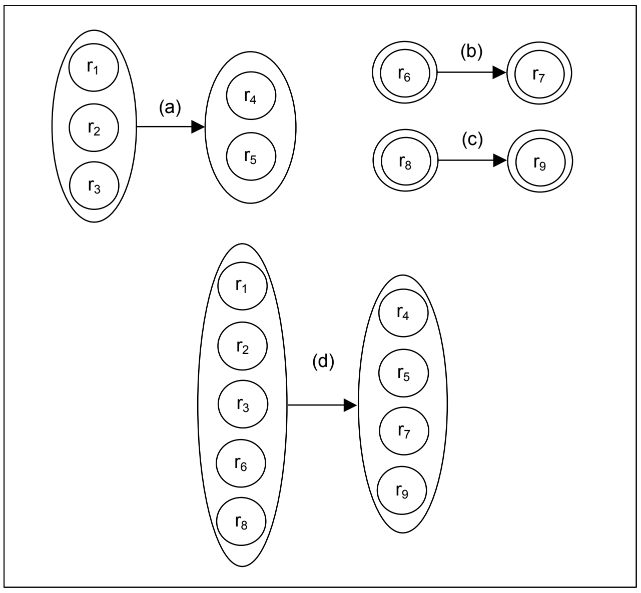

Fig. 3. Priority pairs from $s_{2,1}$ to $s_{2,4}$.

In Fig. 3, the edges of (a), (b), and (c) subsume the edge of (d). It is a generalization of the priority relation that regards one edge as a specialized form of the other edges.

Definition 4. For an edge $e = (v_1, v_2) \in E$, if there is no $e' \in E$ such that $e' \ \pi \ e$, then *e* is a *minimal edge* and v_1, v_2 are *minimal vertices*.

In Fig. 3, the edges and vertices of (a), (b), and (c) are minimal edges and minimal vertices, but the edge and vertices of (d) are not. Finally, the definition of PATI is as follows.

Definition 5. Given a priority relation graph $G = (V, E)$, PATI is $\hat{G} = (\hat{V}, \hat{E})$ where,

$$\hat{V} = \{v \mid v \in V \text{ and } v \text{ is a minimal vertex}\},$$
$$\hat{E} = \{e \mid e \in E \text{ and } e \text{ is a minimal edge}\},$$
$$\hat{w}(e) = w(e) + \sum_{e' \phi e, e' \in E} w(e'),$$

and the *ambiguity type* is a pair of vertices connected with at least one edge.

2. Construction of PATI

Figure 4 shows the construction process of PATI. The English sentence parser analyzes sentences of a corpus and generates a parsed corpus. Human experts build a marked corpus by marking a correct one among candidate trees in the parsed corpus. A priority relation graph is generated from the marked corpus by comparing the correct trees with other candidates. Finally, PATI is constructed using the priority relation graph.

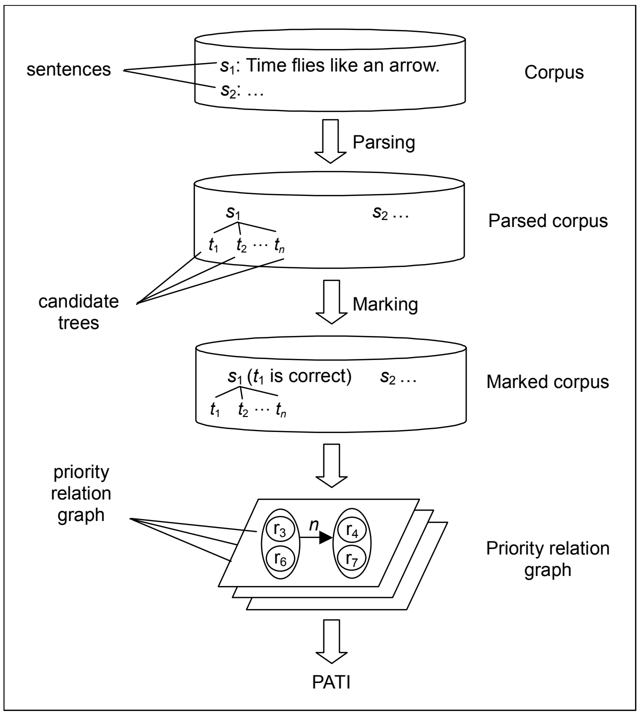

Fig. 4. Construction process of PATI.

Figures 5 and 6 show the algorithms for constructing the priority relation graph and PATI.

We constructed a parsed corpus by analyzing 3,500 English sentences and used a manually built context-free grammar containing about 300 rules. We extracted 133 ambiguity types using the above algorithms. The appendix presents four groups of example ambiguity types. The notation format explaining each ambiguity type is as follows:

Type Number	Difference set 1	Main Causes
	Difference set 2	
	Example Sentences	

III. Grammar Tuning

The appropriateness of linguistic knowledge encoded into grammar rules is a major factor affecting performance of the rule-based approach for ambiguity resolution, but it is quite difficult to determine what is the essential knowledge to be encoded for a grammar under development. The frequency information of PATI provides an efficient way for refining grammar rules. We present two representative methods, *constraint strengthening* and *rule splitting*.

The purpose of constraint strengthening is to reduce the occurrences of ungrammatical candidate trees. Consider the following example.

```
procedure make_priority_relation_graph
/* $C$ is an input corpus, $S$ is a sentence, $T$ is a set of candidate trees,
$t_c$ and $t_i$ are candidate trees, $D_i^c$ and $D_c^i$ are rule set differences,
and $G = (V, E)$ is the resulting priority relation graph. */
begin
    $V \leftarrow \varnothing, E \leftarrow \varnothing$
  for all $S$ in $C$ do
    get $T$ by parsing $S$
    if there is more than one candidate tree then
      $c \leftarrow$ index of the correctly parsed candidate tree
      for all $t_i \in T - \{t_c\}$ do
        get $D_i^c$ and $D_c^i$ by comparing $t_c$ with $t_i$
        $V \leftarrow V \cup \{D_i^c, D_c^i\}$
        if $(D_i^c, D_c^i) \notin E$ then
          $E \leftarrow E \cup \{(D_i^c, D_c^i)\}$, $w((D_i^c, D_c^i)) \leftarrow 1$
        else $w((D_i^c, D_c^i)) \leftarrow w((D_i^c, D_c^i)) + 1$
        endif
      endfor
    endif
  endfor
  return $G = (V, E)$
end
```

Fig. 5. Algorithm for constructing priority relation graph.

```
procedure make_PATI
/* $G = (V, E)$ is the input priority relation graph and $\hat{G} = (\hat{V}, \hat{E})$ is
the resulting PATI. */
begin
  $\hat{V} \leftarrow \varnothing$, $\hat{E} \leftarrow \varnothing$
  for all $(v_i, v_j) \in E$ do
    subsumed $\leftarrow$ 0
    for all $(v_k, v_l) \in E - (v_i, v_j)$ do
      if $v_i \subseteq v_k$ and $v_j \subseteq v_l$ then
        subsumed $\leftarrow$ 1
      endif
    endfor
    if subsumed = 0 then
      $\hat{V} \leftarrow \hat{V}$ Y $\{v_i, v_j\}$, $\hat{E} \leftarrow \hat{E}$ Y $(v_i, v_j)$
      $\hat{w}((v_i, v_j)) \leftarrow w((v_i, v_j))$
    endif
  endfor
  for all $(v_i, v_j) \in \hat{E}$ do
    for all $(v_k, v_l) \in E - \hat{E}$ do
      if $v_i \subseteq v_k$ and $v_j \subseteq v_l$ then
        $\hat{w}((v_i, v_j)) \leftarrow \hat{w}((v_i, v_j)) + w((v_k, v_l))$
      endif
    endfor
  endfor
  return $\hat{G} = (\hat{V}, \hat{E})$
end
```

Fig. 6. Algorithm for constructing PATI.

[**sent** [**pp** Out of the subjects she is taking at] [**sent** [**np** school], [**sent** two are required and three are elective]].]

This analysis can be produced by the rule SENT → NP PUNC SENT and SENT → PP SENT. The former rule is for analyzing sentences that contain vocatives. In order to prevent the above ungrammatical analysis, the latter rule is modified as SENT → PP SENT[–VOCAT] by the method of constraint strengthening. The strengthened constraint '–VOCAT' may contribute to reducing the total number of candidate analyses. PATI automatically provides such candidates using the frequency ratio of two vertices, *FR*, which is defined as:

$$FR(v_i, v_j) = \begin{cases} \dfrac{min(\hat{w}(v_j, v_i), \hat{w}(v_i, v_j))}{max(\hat{w}(v_j, v_i), \hat{w}(v_i, v_j))}, & \text{if } (v_j, v_i) \in \hat{E} \text{ and } (v_i, v_j) \in \hat{E}, \\ 0, & \text{otherwise.} \end{cases}$$

We extract ambiguity types with an *FR* value of 0 and investigate the sentences related to those types for constraint strengthening.

If the *FR* of two vertices is not 0, two edges exist between the two vertices. An *FR* near 1.0 implies that the corresponding ambiguity type cannot be effectively resolved by any syntactic preference function. For example, a prepositional phrase attachment problem is represented by the following two vertices using PATI.

$$v_1 = \{\text{NP} \rightarrow \text{NP PP}\}$$
$$v_2 = \{\text{VP} \rightarrow \text{VP PP}\}.$$

Intuitively we can guess that $FR(v_1, v_2)$ may be near 1.0 and that other kinds of preference functions, such as the lexical collocation function, are needed to resolve this ambiguity type. In the rest of this paper, we refer to this kind of ambiguity type as a *high FR* (*HFR*) type. On the other hand, an *FR* value near 0 means that syntactic information can play an important role in resolving that ambiguity type. Rule preference functions can be very effective for disambiguation in this case. Constraint strengthening is a more active method in the sense that it can prevent ungrammatical trees from being produced.

Rule splitting can make grammar rules more suitable for efficient ambiguity resolution by reducing the overall portion of *HFR* types in PATI. As explained above, if *HFR* types are reduced, syntactic preference functions work better in integration with other kinds of preference functions. Let's consider again the PP attachment problem. The rule in v_2 attaches PP to VP[3]. By adding subcategorization information of the predicate of VP into the constraints of the rule, we can expect *HFR* to decrease for ambiguity types related to the PP attachment. More generally, for a current rule (a) shown below, a new constraint c_{i+1} is considered in addition for splitting, and the resulting rules (b) and (c) will have c_{i+1} and $\neg\ c_{i+1}$, respectively, as their new constraints. Ambiguity types with *HFR* greater than a certain threshold can be extracted from PATI and rule splitting is considered.

[3] Here, for simplicity, the current content of constraints on the non-terminals is not presented.

(a) $A[c_0, c_1, \Lambda\ , c_i]$
(b) $A'[c_0, c_1, \Lambda\ , c_i, c_{i+1}]$
(c) $A''[c_0, c_1, \Lambda\ , c_i, \neg\ c_{i+1}]$

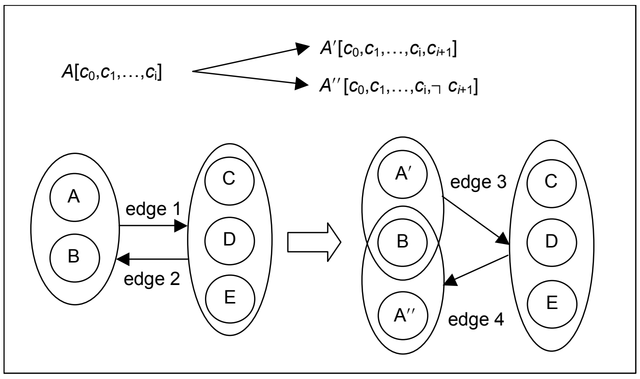

Fig. 7. Concept and consequence of rule splitting.

Figure 7 shows the concept and consequence of rule splitting. For example, let's consider following sentences and rules.

$s_{3,1}$: The bus driver **made John stop**.
$s_{3,2}$: She **made holiday plans**.

r_1: INFCL → VP
r_2: VP[+OCOMP[4]] → VP[+OBJ] INFCL
r_3: NP → NOUN[–PLURAL] NP
r_4: NP → NOUN[–PLURAL, +HUMAN] NP
r_5: NP → NOUN[–PLURAL, –HUMAN] NP

Two edges of different directions in the original ambiguity type come from the parsed results of sentences $s_{3,1}$ and $s_{3,2}$. By splitting r_3 with the additional constraint *HUMAN*, we get r_4 and r_5 and the resulting ambiguity types. This rule splitting process is shown in Fig. 8.

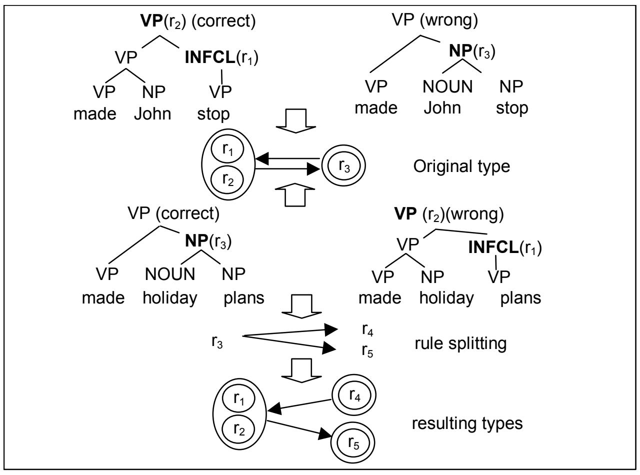

Fig. 8. Example of rule splitting.

[4] Object complement.

As we can see in the above examples, PATI indicates candidate rules to be refined, and this alleviates the human efforts of grammar tuning. Constraint strengthening and rule splitting are the same in spirit, that is, they give a description of rules in more detail. The difference is that the aim of the former is to prevent ungrammatical structures already found from occurring, whereas the latter makes the syntactic preference function (explained in section IV) more effective.

IV. Rule Preference Function and Overall Scoring Scheme

In large-scale rule-based analysis systems, various kinds of preference functions are chosen and combined to produce a score for selecting the best parsed candidate tree. Some functions are based on lexical or semantic collocations while others are based on syntactic information.

In this paper, we focus on the latter though we also have functions based on lexical probabilities or collocations. Syntactic preference functions may simply count particular constructs, such as adjunct and attachment, or estimate probabilities of rules. Assuming that various aspects of syntactic structures are already reflected in PATI, this paper adopts a syntactic preference function that is only based on the rule preference function, *RP*(*r*), defined as follows:

$$RP(r) = \ln f_h(r) - \ln f_l(r),$$

$$f_h(r) = \sum_{r \in v_i, (v_j, v_i) \in \hat{E}} \hat{w}((v_j, v_i)),$$

$$f_l(r) = \sum_{r \in v_i, (v_i, v_j) \in \hat{E}} \hat{w}((v_i, v_j)),$$

where *r* is a rule, $f_h(r)$ is the sum of the weights of incoming edges into the vertices containing *r*, and $f_l(r)$ is the sum of the weights of outgoing edges. This function is different from conventional rule probability functions in two ways. First, it considers only the frequencies from PATI, not the total frequencies. Second, it also incorporates the term 'ln $f_l(r)$' representing the frequencies from badly parsed trees, that is, negative examples.

Figure 9 shows a sample PATI for illustrating the calculation of rule preference scores. Using this PATI, *RP*(*r*) is calculated as follows:

$f_h(r_1) = 1230 + 922 = 2152,\ f_l(r_1) = 507 + 678 = 1185,$
$RP(r_1) = \ln 2152 - \ln 1185 = 7.67 - 7.08 = 0.59$

$f_h(r_2) = 1230,\ f_l(r_2) = 507,$
$RP(r_2) = \ln 1230 - \ln 507 = 7.11 - 6.23 = 0.88$

$f_h(r_3) = 507 + 922 = 1429,\ f_l(r_3) = 1230 + 678 = 1908,$
$RP(r_1) = \ln 1429 - \ln 1908 = 7.67 - 7.08 = -0.29$

$f_h(r_4) = 1230 + 678 = 1908,\ f_l(r_4) = 507 + 922 = 1429,$
$RP(r_4) = \ln 1908 - \ln 1429 = 7.08 - 7.67 = 0.29$

$f_h(r_5) = 507, f_l(r_5) = 1230,$
$RP(r_5) = \ln 507 - \ln 1230 = 6.23 - 7.11 = -0.88$

$f_h(r_6) = 1321, f_l(r_6) = 1020,$
$RP(r_6) = \ln 1321 - \ln 1020 = 7.19 - 6.93 = 0.26$

$f_h(r_7) = 1020 + 922 = 1942,\ f_l(r_7) = 1321 + 678 = 1999,$
$RP(r_7) = \ln 1942 - \ln 1999 = 7.57 - 7.60 = -0.03$

$f_h(r_8) = 466 + 678 = 1144,\ f_l(r_8) = 874 + 922 = 1796,$
$RP(r_8) = \ln 1144 - \ln 1796 = 7.04 - 7.49 = -0.45.$

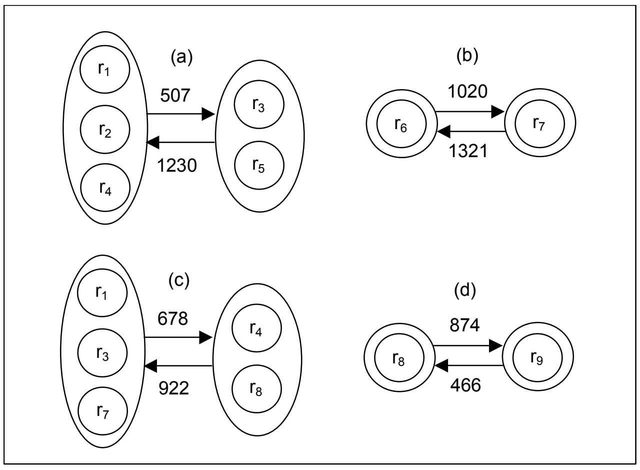

Fig. 9. Sample PATI.

The syntactic preference function, *SP*(*t*), is defined as follows:

$$SP(t) = \sum_{r \in PR(t)} RP(r),$$

where *t* is a candidate tree and *PR*(*t*) is the set of rules participating in building the tree. For example, s_1 in Fig. 1 has two candidate trees. Thus, *SP*(*t*) is calculated as follows:

$$SP(t_1) = RP(r_1) + RP(r_2) + RP(r_4) + RP(r_5) + RP(r_7) + RP(r_8)$$
$$= 0.59 + 0.88 + 0.29 - 0.88 - 0.03 - 0.45 = 0.40$$

$$SP(t_2) = RP(r_1) + RP(r_2) + RP(r_3) + RP(r_5) + RP(r_6) + RP(r_8)$$
$$= 0.59 + 0.88 - 0.29 - 0.88 + 0.26 - 0.45 = 0.11.$$

In the above calculation, $SP(t_1)$ is greater than $SP(t_2)$. Therefore, the candidate tree t_1 is selected as the correct one in view of the syntactic preference function.

The syntactic preference function is combined with other preference functions to produce evaluating scores for candidate trees. We use a lexical preference function that is based on part-of-speech probabilities and two semantic collocation functions [15], [16]. All the preference functions are combined by the method proposed in [13].

V. Experiments

In this section, we present two types of experimental results. One supports the usefulness of PATI for grammar development in a large-scale rule-based natural language analysis system. The other shows that PATI can increase the accuracy of ambiguity resolution.

We developed a general purpose parser implemented by C language on a Unix machine. The coverage of the parser, the percentage of the test sentences for which a correct parse was found, was 97.1%. For broad coverage of the analysis, the initial grammar rules were constructed with minimal constraints. PATI was constructed using information extracted from the initial grammar and the corpus in Table 1.

Table 2 shows the statistics of the initial PATI. In the table, “Sum of Frequencies” represents the sum of weights of edges corresponding to an ambiguity type. The *ambiguity complexity* (AC) represents the amount of ambiguity in the sentence analysis and is defined as follows:

Table 1. Corpus for constructing PATI.

Sentence Length	Area-1	Area-2	Area-3	Total
1–10	542	411	340	1,293
11–20	410	457	417	1,284
21–30	248	282	393	923
Total	1,200	1,150	1,150	3,500

Area-1: High School English Textbook
Area-2: IBM Manual ‘SQL/DS Concepts and Facilities’
Area-3: USA Today

$$AC = \frac{\sum \text{weights in PATI}}{|\text{sentences in a corpus}|}$$

Using the initial PATI, the grammar is tuned as described in section III. A new PATI is constructed after constraint strengthening and rule splitting. Table 3 gives the statistics of PATI using the tuned grammar.

The increase in the number of ambiguity types is due to the increase in the number of rules by the rule splitting process, but the ratio of ambiguity types with *FR* values under 0.2 becomes larger. This implies that a larger portion of all the ambiguity types could be effectively resolved by the syntactic preference

Table 2. Statistics of ambiguity types from the initial grammar.

Area	$FR \le 0.2$		$FR > 0.2$		Total		Ambiguity Complexity
	Number of Types	Sum of Frequencies	Number of Types	Sum of Frequencies	Number of Types	Sum of Frequencies	
Area-1	38	4,203	95	11,820	133	16,023	13.35
Area-2	43	4,808	90	13,028	133	17,836	15.51
Area-3	41	5,560	92	15,893	133	21,453	18.65

Table 3. Statistics of ambiguity types from the tuned grammar.

Area	$FR \le 0.2$		$FR > 0.2$		Total		Ambiguity Complexity
	Number of Types	Sum of Frequencies	Number of Types	Sum of Frequencies	Number of Types	Sum of Frequencies	
Area-1	114	3,478	177	5,745	291	9,223	7.69
Area-2	108	3,810	173	6,317	291	10,127	8.81
Area-3	119	4,011	172	7,091	291	11,102	9.65

Table 4. Test corpus.

Sentence Length	Area-1	Area-2	Area-3	Total
1–10	134	120	148	402
11–20	181	185	245	611
21–30	180	194	213	487

function. In addition, because many ungrammatical candidates are prevented from being built, the sum of frequencies decreases while the number of ambiguity types increases. This is important because it can contribute to reducing the total amount of ambiguity.

We also constructed a test corpus using sentences from the three areas used in constructing PATI. Table 4 shows the statistics of the test corpus. Table 5 shows the accuracies of ambiguity resolution using the initial grammar and the tuned grammar. The results using the initial grammar demonstrate that the performance of our syntactic preference function is superior to that of simple rule probabilities.

The rule probability is calculated for each non-terminal (NP, VP, SENT, …). In constructing the parsed corpus described in section II, the rule count is summed respectively for each non-terminal in the correct parse trees. The probability of each rule is calculated as:

$$p(r_{N^i}^k) = \frac{|r_{N^i}^k|}{\sum_j^{n_k} |r_{N^i}^j|},$$

where $r_{N^i}^k$ is the k-th rule that has N^i as a left-hand side non-terminal, and n_k is the number of occurrences of the k-th rule. On the other hand, PATI contains information from all the candidate trees, including the badly parsed trees. This may give rise to a better performance of *SP*. As expected, we obtained more enhanced disambiguation accuracies using the tuned grammar. In this case, the accuracies using *SP* are also higher than those using rule probabilities.

In the table, the column 'Combine' shows the accuracies using the overall scoring scheme combining *SP* and other kinds of preference functions described in section IV with a sentence segmentation technique [17], [18]. Long sentences are analyzed in a segment-by-segment parsing method. They are segmented into several segments before parsing, and then each segment is parsed. The parse tree is built by combining the analysis results of each segment. With the help of the above method, the parsing complexity can be reduced.

VI. Conclusion

We proposed PATI as an efficient way of developing grammar rules for large-scale applications and providing a syntactic preference function for ambiguity resolution. An initial PATI was constructed from an initial grammar and a parsed corpus. The grammar was enhanced with the help of PATI and a new PATI was constructed to get a syntactic preference function.

The PATI contains information about more ambiguity types with reduced ambiguity complexity of the analysis. We achieved a very high accuracy of ambiguity resolution for an open domain test corpus. We also verified that the syntactic preference function based on PATI contributes significantly to this problem.

All kinds of ambiguous situations, not only the well known cases, such as the PP attachment problem, but also cases that have never been treated with formal linguistic descriptions, could be identified by PATI. Furthermore, PATI can be obtained directly from a comparatively small parsed corpus and at a low cost of human effort.

Table 5. Performance comparison of preference functions.

Sentence Length	Number of Sentences	Average Number of Candidates	Disambiguation Accuracy (%) (using the initial grammar)		Disambiguation Accuracy (%) (using the tuned grammar)		
			Rule Prob.	SP	Rule Prob.	SP	Combine
1–10	402	3.71	48.73	69.65	52.42	86.72	92.25
11–20	611	5.33	31.17	59.92	33.70	74.50	89.68
21–30	487	13.58	14.22	28.90	15.91	43.24	82.83
Total	1,500	7.57	30.37	52.46	32.94	67.63	88.14

For future work, we plan two kinds of studies. We will develop tools supporting grammar tuning to reduce human efforts. Machine learning techniques will be adopted for more effective integration of the syntactic preference function with other kinds of preference functions. We expect this will improve the accuracy of ambiguity resolution.

Appendix

Group A: Grammatical function change of verb phrases

1	$PP \rightarrow PREP^3\ NP^4 \quad VP \rightarrow VP^1\ PP^2$	Part of speech ambiguity
	$NP \rightarrow NOUN^1\ NP^2 \quad VP \rightarrow VP^3\ NP^4$	
	Time^1 flies^2 like^3 an arrow^4.	

2	$NP \rightarrow NOUN^4\ NP^5$ $NP \rightarrow NP^2\ CONJ^3\ NP^5$	Part of speech ambiguity, conjunction (and)
	$SENT \rightarrow NP^4\ VP^5$ $SENT \rightarrow SENT^1\ CONJ^3\ SENT^5$	
	It provides^1 utilities^2 and^3 communication^4 files^5.	

3	$VP \rightarrow VP^1\ CONJ^3\ VP^4$	Part of speech ambiguity, conjunction (or)
	$NP \rightarrow NP^2\ CONJ^3\ NP^4$	
	Light cannot curve^1 around the earth^2 or^3 travel^4.	

4	$SENT \rightarrow NP^3\ VP^4 \quad SUBCL \rightarrow CONJ^2\ SENT^4$ $VP \rightarrow VP^1\ SUBCL^4$	Part of speech ambiguity, conjunction (as)
	$NP \rightarrow NOUN^3\ NP^4 \quad PP \rightarrow PREP^1\ NP^4$ $VP \rightarrow VP^1\ PP^4$	
	I desire^1 money as^2 people^3 desire^4.	

5	$SENT \rightarrow NP^2\ VP^3 \quad RLCL \rightarrow SENT^3$ $VP \rightarrow VP^1\ RLCL^3$	Part of speech ambiguity, verb phrase (show)
	$NP \rightarrow NOUN^2\ NP^3 \quad VP \rightarrow VP^1\ NP^3$	
	A survey shows^1 the rate^2 fall^3 to 7.86 percent.	

6	$NP \rightarrow NOUN^2\ NP^3$	Part of speech ambiguity, verb phrase (make)
	$INFCL \rightarrow VP^3$ $VP[+OCOMP] \rightarrow VP[+OBJ]^1\ INFCL^3$	
	He thanked Clinton for making^1 the three-hour^2 stop^3 at Kigali.	

7	$VP \rightarrow VP^4\ NP^5 \quad INFCL \rightarrow PREP^3\ VP^4$ $NP \rightarrow NP^2\ INFCL^4$	Part of speech ambiguity, *to* infinitive phrase (make)
	$NP \rightarrow AJP^4\ NP^5 \quad PP \rightarrow PREP^3\ NP^5$ $VP \rightarrow VP^1\ PP^5$	
	They guaranteed^1 the right of slave owners^2 to^3 own^4 slaves^5.	

8	$VP \rightarrow VP^2\ NP^3$	Part of speech ambiguity, past participle phrase
	$NP \rightarrow NP^1\ PASTP^2$	
	The boy^1 called^2 names^3.	

9	$SENT \rightarrow NP^4\ VP^5$ $SENT \rightarrow SENT^1\ CONJ^3\ SENT^5$	Conjunction (and), past participle phrase
	$NP \rightarrow NP^2\ CONJ^3\ NP^4 \quad NP \rightarrow NP^2\ PASTP^5$	
	Pierre fell^1 in love with this bright girl^2 and^3 they^4 got^5 married.	

10	$SENT \rightarrow NP^3\ VP^4 \quad RLCL \rightarrow SENT^4$ $VP \rightarrow VP^2\ RLCL^4$	Past participle phrase
	$VP \rightarrow VP^2\ NP^3 \quad NP \rightarrow NP^1\ PASTP^2$	
	The boy^1 said^2 the girl^3 played^4.	

11	$VP \rightarrow VERB^2\ VP^3 \quad SENT \rightarrow PP^1\ PUNC^4\ SENT^5$	Comma, present participle phrase
	$SENT \rightarrow PRESP^3\ PUNC^4\ SENT^5$ $SENT \rightarrow PP^1\ SENT^5$	
	Out of the subjects^1 she is^2 taking^3 at school,^{4} two are required and^5 three are elective.	

12	$VP \rightarrow VERB^1\ VP^2$	Present participle phrase
	$VP \rightarrow VP^1\ PRESP^2$	
	He is^1 working^2.	

13	$NP \rightarrow PRESP^2\ NP^3$	Present participle phrase
	$VP \rightarrow VP^1\ PRESP^2$	
	It contains^1 operating^2 systems^3.	

14	$VP \rightarrow VP^2\ NP^3 \quad PP \rightarrow PREP^1\ PRESP^2$	Present participle phrase
	$NP \rightarrow PRESP^2\ NP^3 \quad PP \rightarrow PRESP^1\ NP^3$	
	This bill is not primarily about^1 fixing^2 America's infrastructure^3.	

15	$VP \rightarrow VP^1\ NP^2 \quad SENT \rightarrow INFCL^1\ VP^3$	*to* infinitive phrase
	$SENT \rightarrow NP^2\ VP^3 \quad SENT \rightarrow INFCL^1\ SENT^3$	
	To desire^1 food or^2 drink is^3 lust.	

16	$\mathrm{RLCL}\rightarrow\mathrm{SENT}^2\ \ \mathrm{NP}\rightarrow\mathrm{NP}^1\,\mathrm{RLCL}^2$ $\mathrm{SENT}\rightarrow\mathrm{PP}^1\,\mathrm{PUNC}^3\,\mathrm{SENT}^4$	Comma, relative clause
	$\mathrm{SENT}\rightarrow\mathrm{SENT}^2\,\mathrm{PUNC}^3\,\mathrm{SENT}^4$ $\mathrm{SENT}\rightarrow\mathrm{PP}^1\,\mathrm{SENT}^3$	
	Out of the subjects^1 she is taking^2 at school,^{3} two are required and^4 three are elective.	

17	$(\mathrm{VP}\rightarrow\mathrm{VP}^2\,\mathrm{CONJ}^3\,\mathrm{VP}^4)$	Conjunction (and), relative pronoun (who)
	$(\mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{CONJ}^3\,\mathrm{VP}^4)$	
	Her children showed^1 their gratitude to her who raised^2 and^3 educated^4 them. I saw^1 him steal^2 a pound of butter and^3 put^4 it in his hat.	

Group B: Nucleus change of verb phrases

1	$\mathrm{PP}\rightarrow\mathrm{PREP}^2\,\mathrm{NP}^3\ \ \mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{PP}^3$	Part of speech ambiguity
	$\mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{AVP}^2\ \ \mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{NP}^3$	
	Light cannot curve^1 around^2 the earth or^3 travel.	

2	$\mathrm{SENT}\rightarrow\mathrm{AVP}^2\,\mathrm{SENT}^3\ \ \mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{SENT}^3$	Part of speech ambiguity
	$\mathrm{SUBCL}\rightarrow\mathrm{CONJ}^2\,\mathrm{SENT}^3\ \ \mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{SUBCL}^3$	
	I know^1 where^2 your book is^3.	

3	$\mathrm{RLCL}\rightarrow\mathrm{PRON}^3\,\mathrm{SENT}^4\ \ \mathrm{NP}\rightarrow\mathrm{NP}^2\,\mathrm{RLCL}^4$	Part of speech ambiguity
	$\mathrm{SUBCL}\rightarrow\mathrm{CONJ}^3\,\mathrm{SENT}^4\ \ \mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{SUBCL}^4$	
	I know^1 the place^2 where^3 your book is^4.	

4	$\mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{AJP}^2$	Part of speech ambiguity
	$\mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{NP}^2$	
	I got^1 red^2.	

5	$\mathrm{NP}\rightarrow\mathrm{AJP}^1\,\mathrm{NP}^2$	Part of speech ambiguity
	$\mathrm{VP}\rightarrow\mathrm{AVP}^2\,\mathrm{VP}^3$	
	This^1 light^2 cannot curve^3.	

6	$\mathrm{NP}\rightarrow\mathrm{NOUN}^1\,\mathrm{NP}^2$	Part of speech ambiguity
	$\mathrm{VP}\rightarrow\mathrm{AVP}^2\,\mathrm{VP}^3$	
	Sun^1 light^2 cannot curve^3.	

7	$\mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{AVP}^2$	Part of speech ambiguity
	$\mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{NP}^2$	
	I said^1 Tuesday^2.	

8	$\mathrm{RLCL}\rightarrow\mathrm{PRON}^1\,\mathrm{SENT}^3$	Part of speech ambiguity
	$\mathrm{NP}\rightarrow\mathrm{AJP}^1\,\mathrm{NP}^2\ \ \mathrm{RLCL}\rightarrow\mathrm{SENT}^3$	
	One reason is that^1 light^2 cannot curve^3.	

9	$\mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{AJP}^2$	Part of speech ambiguity
	$\mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{AVP}^2$	
	The market was^1 lower^2.	

10	$\mathrm{RLCL}\rightarrow\mathrm{PRON}^2\,\mathrm{SENT}^3\ \ \mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{RLCL}^2$	Part of speech ambiguity
	$\mathrm{RLCL}\rightarrow\mathrm{SENT}^3\ \ \mathrm{NP}\rightarrow\mathrm{NP}^2\,\mathrm{RLCL}^3$ $\mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{NP}^2$	
	One reason is^1 that^2 light cannot curve^3.	

11	$\mathrm{VP[+IOBJ]}\rightarrow\mathrm{VP}^1\,\mathrm{NP}^2$	Verb phrase
	$\mathrm{VP[+OBJ]}\rightarrow\mathrm{VP}^1\,\mathrm{NP}^2$	
	He told^1 Clinton^2.	

12	$(\mathrm{NP}\rightarrow\mathrm{NOUN}^1\,\mathrm{NP}^2)$	Noun phrase
	$(\mathrm{NP}\rightarrow\mathrm{NOUN}^2\,\mathrm{NP}^3)$	
	I showed a book^1 name^2 people^3 know.	

13	$(\mathrm{VP}\rightarrow\mathrm{VP}^2\,\mathrm{NP}^3)$	Conjunction, verb phrase
	$(\mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{NP}^3)$	
	He saluted and^1 held^2 the book^3.	

14	$\mathrm{NP}\rightarrow\mathrm{NOUN}^2\,\mathrm{NP}^3$	Preposition phrase
	$\mathrm{VP}\rightarrow\mathrm{VP}^1\,\mathrm{NP}^3$	
	They mate^1 for family^2 groups^3.	

Group C: Additional word change of verb phrases

1	$\mathrm{NP}\rightarrow\mathrm{PRESP}^1\,\mathrm{NP}^2$	Part of speech ambiguity
	$\mathrm{VP}\rightarrow\mathrm{NOUN}^1\,\mathrm{NP}^2$	
	I heard beating^1 drums^2.	

2	$\mathrm{NP}\rightarrow\mathrm{AJP}^1\,\mathrm{NP}^2$	Part of speech ambiguity

	$VP \rightarrow NOUN^1\ NP^2$	
	One^1 reason^2 is that you cannot go.	

3	$NP \rightarrow NOUN^2\ NP^3$	Part of speech ambiguity
	$VP \rightarrow VP^1\ AVP^2$	
	I like^1 Saturday^2 parties^3.	

4	$SENT \rightarrow AVP^1\ SENT^3$	Adverbial phrase
	$NP \rightarrow AVP^1\ NP^2$	
	Also^1 the companies^2 grow^3.	

5	$AJP \rightarrow AVP^2\ AJP^3$	Adverbial phrase
	$VP \rightarrow VP^1\ AVP^2$	
	She is^1 so^2 beautiful^3.	

6	$VP \rightarrow VP^1\ PP^3$	Preposition phrase
	$NP \rightarrow NP^2\ PP^3$	
	I eat^1 a fish^2 with a fork^3.	

7	$VP \rightarrow VP^1\ INFCL^3$	*to* infinitive phrase
	$NP \rightarrow NP^2\ INFCL^3$	
	He put^1 off his hat^2 to sleep^3.	

8	$VP \rightarrow VP^1\ PRESP^3$	Present participle phrase
	$NP \rightarrow NP^2\ PRESP^3$	
	He saw^1 the flowers^2 walking^3 there.	

Group D: Rule application scope change

1	$(VP \rightarrow VP^1\ AVP^3)$	Conjunction, adverbial phrase
	$(VP \rightarrow VP^2\ AVP^3)$	
	They study or^1 work^2 abroad^3.	

2	$(PP \rightarrow PREP^1\ NP^2)$	Conjunction, preposition phrase
	$(PP \rightarrow PREP^1\ NP^3)$	
	He used songs of^1 birds^2 and^3 others	

3	$(SENT \rightarrow PP^1\ PUNC^2\ SENT^4)$	Conjunction, preposition phrase
	$(SENT \rightarrow PP^1\ PUNC^2\ SENT^3)$	
	At home^1,^{2} she slept^3,^{4} he worked.	

4	$(NP \rightarrow NP^1\ PUNC^2\ NP^3)$	Conjunction, noun phrase
	$(NP \rightarrow NP^1\ PUNC^2\ NP^4)$	
	I like music^1,^{2} art^3, and^4 dance.	

5	$(VP \rightarrow VP^1\ PUNC^2\ VP^3)$	Conjunction, verb phrase
	$(VP \rightarrow VP^1\ PUNC^2\ VP^4)$	
	I wake^1 up,^{2} eat^3, and^4 sleep.	

6	$(NP \rightarrow PRESP^1\ NP^2)$	Conjunction, present participle phrase
	$(NP \rightarrow PRESP^1\ NP^3)$	
	We used sleeping^1 bags^2 or^3 boots.	

References

[1] http://www.easytran.com.

[2] K.L. Baker, A.M. Franz, and P.W. Jordan, "Coping with Ambiguity in Knowledge-based Natural Language Analysis," *Proc. of COLING*-94, 1994, pp. 90-94.

[3] S. Kwasny and N.K. Sondheimer, "Relaxation Theories for Parsing Ill-Formed Input," *American Journal of Computational Linguistics*, vol. 7, no. 2, 1981, pp. 99-108.

[4] Ho-Young Jung, Mansoo Park, Hoi-Rin Kim, and Minsoo Hahn, "Speaker Adaptation Using ICA-Based Feature Transformation," *ETRI J.*, vol. 24, no. 6, Dec. 2002, pp. 469-472.

[5] E. Charniak, "Statistical Parsing with a Context-Free Grammar and Word Statistics," *Proc. of the Fourteenth Nat'l Conf. on Artificial Intelligence* (AAAI97), 1997, pp. 598-603.

[6] M. Erasn and E. Charniak, "A Statistical Syntactic Disambiguation Program and What It Learns," *Symbolic, Connectionist, and Statistical Approaches to Learning for Natural Language Processing*, 1996, pp. 146-159.

[7] M. Johnson, S. Geman, S. Canon, Z. Chi, and S. Riezler, "Estimators for Stochastic Unification-Based Grammars," *Proc. of the 37th Annual Meeting of the Association for Computational Linguistics* (ACL'99), 1999.

[8] S. Riezler, T. King, R. Kaplan, R. Crouch, J. Maxwell, and M. Johnson, "Parsing the Wall Street Journal Using a Lexical-Functional Grammar and Discriminative Estimation Techniques," *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

[9] J. Wang, "Syntactic Preferences for Robust Parsing with Semantic Preferences," *Proc. of COLING*-92, 1992, pp. 239-245.

[10] J. Kimball, "Seven Principles of Surface Structure Parsing in Natural Language," *Cognition*, vol. 2, 1973, pp. 15-47.

[11] M.G. Dyer, "Symbolic Neuro Engineering and Natural Language Processing: A Multilevel Research Approach," *Advances in Connectionist and Neural Computation Theory*, vol. 1, Ablex

Publishing Corp., 1991, pp. 32-68.

[12] D.L. Waltz and J.B. Pollack, "Massive Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation," *Cognitive Science*, vol. 9, 1985, pp. 51-74.

[13] H. Alshawi and D. Carter, "Training and Scaling Preference Functions for Disambiguation," *Computational Linguistics*, vol. 20, no. 4, 1994, pp. 635-648.

[14] G. Gazdar, E. Klein, G. Pullum, and I. Sag, *Generalized Phrase Structure Grammar*, Blackwell, 1985.

[15] K.S. Shim, *Structural Disambiguation of to-infinitives Using Augmented Collocations*, Ph.D. thesis, Department of Computer Engineering, Seoul National University, 1994.

[16] S.J. Chun, *A Study on Prepositional Phrase Attachment and the Transfer of the Preposition Using Semantic Hierarchy*, Master thesis, Department of Computer Engineering, Seoul National University, 1994.

[17] S.D. Kim, "Reducing Parsing Complexity by Intra-Sentence Segmentation Based on Maximum Entropy Model," *Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, 2000.

[18] S.D. Kim, B.T. Zhang, and Y.T. Kim, "Learning-Based Intrasentence Segmentation for Efficient Translation of Long Sentences," *Machine Translation*, vol. 16. no. 3, 2001, pp. 151-174.

Jae Won Lee has been a full-time Instructor of the School of Computer Science and Information at Sungshin Women's University in Seoul, Korea since 1999. He received his BS, MS, and PhD degrees in computer engineering from Seoul National University in 1990, 1992, and 1998. He received the best paper award at the 7th Pacific Rim International Conference on Artificial Intelligence (PRICAI-02). His current research interests include computational finance, artificial intelligence, machine learning, natural language processing, and computer music.

Sung-Dong Kim has been an Assistant Professor of the Department of Computer System Engineering at Hansung University in Seoul, Korea since 2001. He received his BS, MS, and PhD degrees in computer engineering from Seoul National University in 1991, 1993, and 1999. He received the best paper award at the 7th Pacific Rim International Conference on Artificial Intelligence (PRICAI-02). His current research interests include machine translation, natural language processing, computational finance, machine learning, and data mining.

Jinseok Chae is an Assistant Professor of Department of Computer Science and Engineering at the University of Incheon, Korea. He received BS, MS, and PhD degrees in computer engineering from Seoul National University in 1990, 1992, and 1998. Formerly, he was an Assistant Staff of the Engineering Laboratory at Seoul National University from 1992 to 1997 and Senior Researcher of the Korea Research Information Center from 1997 to 1998. His research interests include internet software, markup languages and digital library.

Jongwoo Lee is an Assistant Professor of Computer Engineering at Kwangwoon University in Seoul, Korea. He received his BS, MS, and PhD degrees in Computer Engineering from Seoul National University in 1990, 1992, and 1996. From 1996 to 1999, he worked for Hyundai Electronics Industries, Co. He was an Assistant Professor of Division of Information and Telecommunication Engineering at Hallym University in Chooncheon, Korea from 1999 to 2002. His research interests include computational finance, cluster computing, parallel and distributed systems, and system software.

Do-Hyung Kim received his BE in computer engineering in 1985 from Seoul National University and MS and PhD in computer science in 1987 from Korea Advanced Institute of Science and Technology (KAIST). After short period (from Mar. 1992 to Aug. 1992) as a Member of Research Staff at the Information and Electronics Research Institute (IERI) in KAIST, he joined the faculty of the School of Computer Science and Engineering, Sungshin Women's University in September 1992. He primarily teaches courses on programming languages and compiler construction. His research interests include: broadly, programming language design, compiler construction, logic programming; more specifically, parallel execution of logic programs. He also has some interests in algorithm analyses.