PARSING WITHOUT GRAMMAR

Shinsuke Mori and Makoto Nagao
Section of Electronics and Communication, Kyoto University
Yoshida-honmachi, Sakyo, Kyoto, 606-01 Japan
{mori,nagao}@kuee.kyoto-u.ac.jp

Abstract

We describe and evaluate experimentally a method to parse a tagged corpus without grammar modeling a natural language on context-free language. This method is based on the following three hypotheses. 1) Part-of-speech sequences on the right-hand side of a rewriting rule are less constrained as to what part-of-speech precedes and follows them than non-constituent sequences. 2) Part-of-speech sequences directly derived from the same non-terminal symbol have similar environments. 3) The most suitable set of rewriting rules makes the greatest reduction of the corpus size. Based on these hypotheses, the system finds a set of constituent-like part-of-speech sequences and replaces them with a new symbol. The repetition of these processes brings us a set of rewriting rules, a grammar, and the bracketed corpus.

1 Introduction

A standard approach to a natural language analysis is to characterize it with a set of rules, a grammar. Given the difficulty in developing a grammar manually, it is necessary to build a method for automatic grammar induction. One of the most promising results of grammar inference is based on the inside-outside algorithm, which can be used to train a stochastic context-free grammar. It is an extension of the forward-backward algorithm. [Pereira and Schabes, 1992] and [Schabes *et al.*, 1993] proposed a method to infer the parameters of a stochastic context-free grammar from a partially parsed corpus and evaluated the results. [Brill, 1993] describes another technique for grammar induction: "the system learns a set of ordered transformations and applies it to a new sentence."

These two methods profit from corpora annotated with syntactic structure, the Penn Treebank [Marcus and Santorini, 1993]. Corpora in the Penn Treebank have two sorts of additional data: the part-of-speech of each word and the syntactic structure of each sentence. The first stage of building a corpus is an automatic tagging or parsing and the second the manual correction of errors. Since the accuracy of current parsers is not satisfactory, the manual correction of parsing results is an arduous task. As for tagging, however, the method of [Church, 1988], applied to build the Penn Treebank, is reported as "95-99% correct, depending on the definition of correct" and another tagger developed by [Brill, 1992] marks almost the same accuracy.

It follows that it is worth trying to induce a grammar from corpora without syntactic structure, that is to say, to parse corpora using only part-of-speech information. Only a few attempts, however, have been made so far at grammar induction from an unbracketed corpus. [Magerman and Marcus, 1990] proposes a parsing system using mutual information statistics and a manually written "distituent grammar," a list of tag pairs which cannot be adjacent within a constituent, such as (*prep noun*). [Brill and Marcus, 1992] also proposes a technique

for grammar induction.

The operations of extracting a grammar from a tagged corpus and applying it to the very same corpus is equal to the process of parsing the corpus without a grammar. In this paper, we propose a new method to parse a tagged corpus without a grammar based on the following three hypotheses:

1. Part-of-speech sequences on the right-hand side of a rewriting rule are less constrained as to what part-of-speech precedes and follows them than non-constituent sequences.

2. Part-of-speech sequences directly derived from the same non-terminal symbol have similar environments.

3. The most suitable set of rewriting rules makes the greatest reduction of the corpus size.

The initial state of the corpus is part-of-speech sequences. The algorithm finds a group of constituent-like part-of-speech sequences in the corpus and produces rewriting rules which have the part-of-speech sequences on the right-hand side and the same non-terminal symbol on the left-hand side. Applying these rewriting rules to the corpus, the algorithm makes the corpus shorter in terms of number of symbolic units, more parsed. The system repeats these processes until it cannot find any more part-of-speech constituent sequences. Then the system stops, with the corpus parsed and a set of rewriting rules, i.e. a grammar, created.

In subsequent sections, first we discuss the three hypotheses, secondly describe the algorithm, thirdly present results and compare them to other recent results in automatic phrase structure induction, and finally conclude this research and discuss future works.

2 Three Hypotheses

In the theory of formal language, the rewriting rules determine whether a sequence of alphabets is a sentential form or not. Therefore the language, a set of sentential forms, reflects the characteristics of the rewriting rules. Since our method models natural language on context-free language, a corpus is regarded as a set of sentential forms. It follows that a corpus reflects the characteristics of the rewriting rules of the natural language. In the following parts of this section, we present an example of the corpus we used, define some symbols for explanation and discuss the three hypotheses on the relation between a corpus and the rewriting rules derived from this point of view.

2.1 Corpus and Symbol Definitions

For our main experiment, we used part-of-speech (POS) sequences from the Wall Street Journal (WSJ) in the Penn Treebank. Table 1 shows its POS tagset and Figure 1 is an example

```
Nekoosa would n't be a  diversification .
NNP     MD    RB  VB DT NN              .
( ( ( Nekoosa ) would n't ( be ( a diversification ) ) ) . )
```

Figure 1: An example sentence in WSJ

Table 1: Penn Treebank POS tagset

1.	CC	Coordinating conjunction	21.	RBR	Adverb, comparative
2.	CD	Cardinal number	22.	RBS	Adverb, superlative
3.	DT	Determiner	23.	RP	Particle
4.	EX	Existential *there*	24.	SYM	Symbol
5.	FW	Foreign word	25.	TO	*to*
6.	IN	Preposition or subordinating conjunction	26.	UH	Interjection
			27.	VB	Verb, base form
7.	JJ	Adjective	28.	VBD	Verb, past tense
8.	JJR	Adjective, comparative	29.	VBG	Verb, gerund or present participle
9.	JJS	Adjective, superlative			
10.	LS	List item marker	30.	VBN	Verb, past participle
11.	MD	Modal	31.	VBP	Verb, non-3rd person singular present
12.	NN	Noun, singular or mass			
13.	NNS	Noun, plural	32.	VBZ	Verb, 3rd person singular present
14.	NNP	Proper noun, singular			
15.	NNPS	Proper noun, plural	33.	WDT	Wh-determiner
16.	PDT	Predeterminer	34.	WP	Wh-pronoun
17.	POS	Possessive ending	35.	WP$	Possessive wh-pronoun
18.	PRP	Personal pronoun	36.	WRB	Wh-adverb
19.	PRP$	Possessive pronoun	37.	,	Comma
20.	RB	Adverb	38.	.	Sentence-final punctuation

sentence from WSJ. We use the POS information in the second line in this figure for grammar induction and the syntactic structure in the third line for evaluation.

Our method models natural language on context-free language, which is described by a grammar $G = (N, T, P, S)$, where N is the set of non-terminal symbols, T is the set of terminal symbols, P is the set of rewriting rules and S is the start symbol. Since our method regards the corpus as a set of POS sequences, the set of terminal symbols T is equal to the part-of-speech tagset of the Penn Treebank. Non-terminal symbols are, however, introduced by the system and they aren't elements of the syntactic tagset of the Penn Treebank. For the subsequent explanation, we make the following definitions:

$$pos \in T, \quad syn \in N, \quad tag \in N \cup T$$

$$\boldsymbol{syn} \in N^+, \quad \boldsymbol{pos} \in T^+, \quad \boldsymbol{tag} \in (N \cup T)^+$$

where $X^+ = X^* - \{\varepsilon\}$ generally.

2.2 Constituent-like Part-of-speech Sequence

The first step of this method is to extract sequences of one ore more POS from the corpus to form the right-hand side of rewriting rules. The following is the first hypothesis according to which the system extracts constituent-like POS sequences.

- POS sequences on the right-hand side of a rewriting rule are less constrained as to what POS precedes and follows them than non-constituent sequences.

To explain this hypothesis concretely, let us consider the following two different POS sequences: $\boldsymbol{pos}_a$ which appears on the right-hand side of a rewriting rule and $\boldsymbol{pos}_x = \boldsymbol{pos}_{x1} \cdot \boldsymbol{pos}_{x2}$ which

doesn't appear in any rewriting rule. In this case, let us say that G may contain the following rewriting rules:

1. $syn_a \rightarrow \boldsymbol{pos}_a$
2. $syn_b \rightarrow syn_c \cdot syn_d$
3. $syn_c \rightarrow \boldsymbol{pos}_{c'} \cdot \boldsymbol{pos}_{x1}$
4. $syn_d \rightarrow \boldsymbol{pos}_{x2} \cdot \boldsymbol{pos}_{d'}$

These rules tell us that the POSs which precede or follow $\boldsymbol{pos}_a$ are more loosely restricted than those of $\boldsymbol{pos}_x$. In fact, $\boldsymbol{pos}_a$ can appear where syn_a appears, while $\boldsymbol{pos}_x$ requires the last three rewriting rules to be produced, in the following way:

$$syn_b \Rightarrow syn_c \cdot syn_d \Rightarrow \boldsymbol{pos}_{c'} \cdot \underbrace{\boldsymbol{pos}_{x1} \cdot \boldsymbol{pos}_{x2}}_{\boldsymbol{pos}_x} \cdot \boldsymbol{pos}_{d'}$$

In this case, the POS which can precede $\boldsymbol{pos}_x$ is restricted to the last POS of $\boldsymbol{pos}_{c'}$ and the POS which follows $\boldsymbol{pos}_x$ is restricted to the first POS of $\boldsymbol{pos}_{d'}$. This is the foundation of the first hypothesis.

As a measure of constituency of a particular POS sequence, we use the entropies of the probability distributions of the POSs which precede or follow it. These entropies are calculated using the following equations:

$$H_l(\boldsymbol{pos}) = -\sum_i P(pos_i \cdot \boldsymbol{pos} \mid \boldsymbol{pos}) \log P(pos_i \cdot \boldsymbol{pos} \mid \boldsymbol{pos})$$

$$H_r(\boldsymbol{pos}) = -\sum_i P(\boldsymbol{pos} \cdot pos_i \mid \boldsymbol{pos}) \log P(\boldsymbol{pos} \cdot pos_i \mid \boldsymbol{pos})$$

We call H_l and H_r the left-side entropy and the right-side entropy respectively. The conditional probabilities $P(pos_i \cdot \boldsymbol{pos} \mid \boldsymbol{pos})$ and $P(\boldsymbol{pos} \cdot pos_i \mid \boldsymbol{pos})$ are computed from the frequencies of $\boldsymbol{pos}$, $pos_i \cdot \boldsymbol{pos}$ and $\boldsymbol{pos} \cdot pos_i$ in the corpus using the following equations.

$$P(pos_i \cdot \boldsymbol{pos} \mid \boldsymbol{pos}) = \frac{P(pos_i \cdot \boldsymbol{pos})}{P(\boldsymbol{pos})} = \frac{f(pos_i \cdot \boldsymbol{pos})}{f(\boldsymbol{pos})}$$

$$P(\boldsymbol{pos} \cdot pos_i \mid \boldsymbol{pos}) = \frac{P(\boldsymbol{pos} \cdot pos_i)}{P(\boldsymbol{pos})} = \frac{f(\boldsymbol{pos} \cdot pos_i)}{f(\boldsymbol{pos})}$$

Therefore, n-gram statistics, the frequencies of all the symbol sequences appearing in the corpus, are applicable to compute the entropies. To calculate n-gram statistics for arbitrary n, we adopted the algorithm proposed in [Nagao and Mori, 1994]. Notice that n is more than one and does not exceed the length of the longest sentence in the corpus, because n-grams containing a symbol for final punctuation mark never appear in rewriting rules, except for ones with S on the left-hand side.

In addition to entropy, we propose another measure of constituency: the ratio of delimiters which precede or follow the POS sequence in question. In our method the delimiters are sentence-final punctuation mark and comma.

To extract constituent-like POS sequences from the corpus, we set threshold values for the entropies (H_{min}) and for the delimiter ratios (Pd_{min}). From the discussion above, the following four inequalities are the conditions for a constituent-like POS sequence:

1. $H_l(\boldsymbol{pos}) \geq H_{min}$
2. $H_r(\boldsymbol{pos}) \geq H_{min}$
3. $Pd_l(\boldsymbol{pos}) = P(\text{``.''} \cdot \boldsymbol{pos} \mid \boldsymbol{pos}) + P(\text{``,''} \cdot \boldsymbol{pos} \mid \boldsymbol{pos}) \geq Pd_{min}$
4. $Pd_r(\boldsymbol{pos}) = P(\boldsymbol{pos} \cdot \text{``.''} \mid \boldsymbol{pos}) + P(\boldsymbol{pos} \cdot \text{``,''} \mid \boldsymbol{pos}) \geq Pd_{min}$

Under the conditions $H_{min} = 3$ and $Pd_{min} = 0.05$, 429 POS sequences are extracted. Table 2 shows 20 of them in order of frequency.

Table 2: Samples of extracted part-of-speech sequences

f	H_l	H_r	Pd_l	Pd_r	$\boldsymbol{pos}$	f	H_l	H_r	Pd_l	Pd_r	$\boldsymbol{pos}$
54724	3.1	3.3	0.16	0.19	NNP	13327	3.6	3.4	0.06	0.08	VBN
39375	3.6	3.6	0.05	0.27	NNS	9519	3.8	3.5	0.08	0.34	JJ·NNS
23758	3.2	3.5	0.20	0.17	DT·NN	9385	3.2	3.2	0.07	0.22	IN·NNP
20088	3.3	3.4	0.24	0.25	NNP·NNP	9278	3.4	3.5	0.09	0.24	IN·DT·NN
19055	3.1	3.8	0.09	0.06	VBD	7753	3.5	3.0	0.33	0.05	PRP
18460	4.2	4.0	0.17	0.16	RB	7487	3.0	3.3	0.15	0.25	DT·JJ·NN
14550	3.7	3.1	0.06	0.17	CD	7296	3.8	3.5	0.09	0.33	NN·NNS

2.3 Similarity between Constituents

The second step is to cluster the POS sequences extracted in the first step. The following is our second hypothesis, which gives a measure of this clustering.

- POS sequences directly derived from the same non-terminal symbol have similar environments.

To explain the background of this hypothesis, let us consider the case in which the grammar contains the following rewriting rules.

$$syn_e \rightarrow \boldsymbol{pos}_{e1}, \quad syn_e \rightarrow \boldsymbol{pos}_{e2}$$

This means that POS sequences which can appear to the left or right of the two POS sequences $\boldsymbol{pos}_{e1}$ and $\boldsymbol{pos}_{e2}$ are those which can also be found to the left or right of syn_e. It must reasonably be expected that two constituent-like POS sequences which are derived from a single non-terminal symbol will have similar probability distributions for the POSs which precede and follow them. We choose, as a measure of similarity to cluster extracted POS sequences, the Euclidean distance between two probability distributions. The clustering is composed of the following three processes (see Figure 2).

1. Produce a graph whose nodes correspond to POS sequences and whose arcs correspond to the Euclidean distance between two POS sequences.
2. Delete arcs whose value is greater than a threshold value (D_{min}).
3. Decompose the graph into connected components by examining connectivity.

Each connected component corresponds to a cluster and the nodes to their elements. Table 3 shows the result of clustering under the conditions $H_{min} = 3$, $Pd_{min} = 0.05$ *and* $D_{min} = 0.25$. In this table, "area" is a value given to each cluster which is calculated by summing the product of the length and the frequency of each element [1].

2.4 Selecting Rewriting Rules

After clustering, a new set of rewriting rules can be obtained from any cluster by putting each of the POS sequences on the right-hand side of a rule and introducing a new single non-terminal symbol on the left-hand sides. Next, the system applies them to reduce the corpus for

[1] More precisely, the frequency of shorter sequences is reduced by considering their overlap with longer ones.

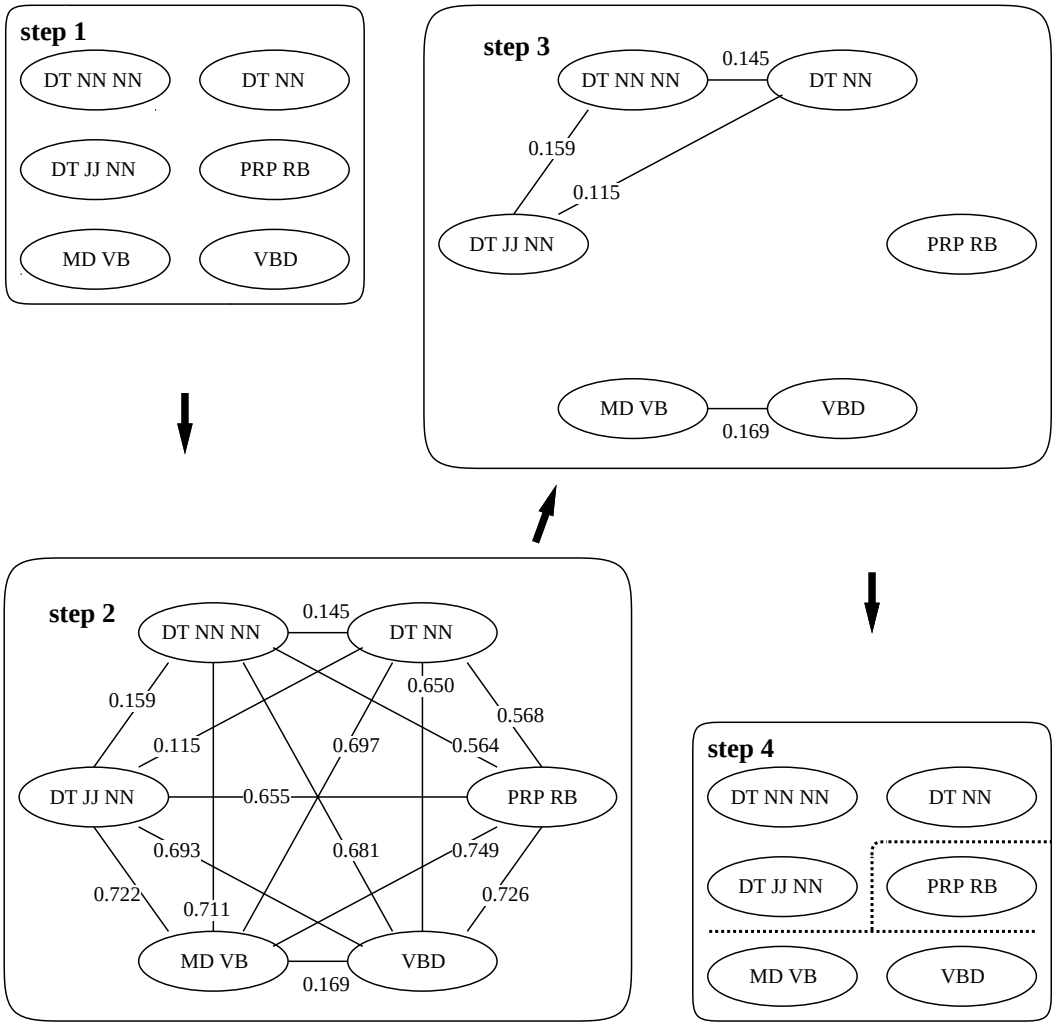

Figure 2: Clustering part-of-speech sequences ($D_{min} = 0.25$)

further extraction of rewriting rules which may contain non-terminal symbols on their right-hand side. It is possible, however, that two or more rewriting rules will conflict with each other. Let us consider the following two rewriting rules made from the first element of the second cluster in Table 3 and the third element of the third cluster, where syn_2 and syn_3 are non-terminal symbols introduced for the second cluster and the third cluster respectively.

$$syn_2 \rightarrow DT \cdot NN \tag{1}$$

$$syn_3 \rightarrow IN \cdot DT \cdot NN \tag{2}$$

In this case, rule (2) can be represented using rule (1), as follows.

$$syn_3 \rightarrow IN \cdot syn_2$$

Considering cases like this, we can conclude that changing all the clusters into rewriting rules and applying them to the corpus would disturb the appropriate extraction of rewriting rules. For this reason, the system selects only one cluster from the output of the second step. We

Table 3: An example of result of clustering

area	f	H_l	H_r	***pos***	area	f	H_l	H_r	***pos***
20260	19055	3.1	3.8	VBD	80317	6553	3.2	3.4	IN·NN
	1205	3.2	3.8	RB·VBD		3722	3.1	3.6	IN·NNS
86473	23758	3.2	3.5	DT·NN		9278	3.4	3.5	IN·DT·NN
	3905	3.0	3.7	DT·NNS		1499	3.6	3.7	IN·DT·NNS
	2137	3.1	3.5	PRP$·NN		1771	3.1	3.3	IN·JJ·NN
	7487	3.0	3.3	DT·JJ·NN		2555	3.2	3.3	IN·JJ·NNS
	1102	3.1	3.7	DT·JJ·NNS		1326	3.4	3.3	IN·DT·NN·NN
	1106	3.2	3.5	DT·NN·IN·DT·NN		3283	3.0	3.5	IN·DT·JJ·NN

"area" is a value given to each cluster which is calculated by summing the product of the length and the frequency of each element.

chose the "area" of the cluster as the measure for selecting the best cluster. This is based on the third hypothesis presented below.

- The most suitable set of rewriting rules makes the greatest reduction of the corpus size.

One may expect that a large value for area means that the rewriting rules of the cluster are basic symbol sequences which appear regardless of their global environment. And the larger their area is, the more their application will reduce the number of symbols in the corpus.

3 Algorithm

We developed a system, based on the hypotheses discussed in the previous section, which extracts a set of rewriting rules from a corpus and applies them to the corpus. The system executes the following four processes repeatedly:

1. Compute n-gram statistics on the corpus,
2. Extract part-of-speech sequences whose frequency is more than f_{min} and which meet the conditions $H_{min} = 3$ and $Pd_{min} = 0.05$,
3. Cluster the part-of-speech sequences and select a cluster to produce a set of rewriting rules,
4. Rewrite the corpus, applying the rewriting rules.

In the experiment, the initial value of f_{min} is equal to 2,000. If no POS sequence is extracted in the second process, f_{min} is multiplied by 0.9 and the second process is repeated with the new f_{min}. If f_{min} becomes less than 50, the system stops. Below, we describe more precisely the last two processes.

3.1 Produce Rewriting Rules

As we described above, the system extracts constituent-like POS sequences and clusters them depending on their distributional similarity. Next it selects the cluster which has the largest "area" in the corpus to produce rewriting rules. The last process is to select a non-terminal to put on the left side of rewriting rules. There are two cases, depending on the number of

POS sequences in the cluster which consist of a single non-terminal symbol. If there is exactly one, that non-terminal symbol is put on the left-hand side to produce rewriting rules, and the rewriting rule which would have that non-terminal symbol on both sides is erased. In the other case, i.e. the number of POS sequences composed of a single non-terminal symbol is zero or more than one, the system introduces a new non-terminal symbol and puts it on the left-hand side to produce rewriting rules.

3.2 Rewrite Corpus

After having produced a set of rewriting rules, the system applies them to every sentence in the corpus. At this point, there are two problematic situations. The first one is that a rewriting rule is applicable in more than one way. For example, suppose that the rewriting rule is $syn_1 \rightarrow tag_1 \cdot tag_1$ and the following symbol sequence exists in the corpus:

$$\ldots tag_1 \cdot tag_1 \cdot tag_1 \ldots$$

The rule is applicable to both the first and the last $tag_1 \cdot tag_1$ at the same time. To handle cases like this, the system applies each rewriting rule simply from left to right.

The second case is that two or more rewriting rules conflict with each other. For example, suppose that there are two rewriting rules such as

$$syn_1 \rightarrow tag_1 \cdot tag_2 \tag{3}$$

$$syn_1 \rightarrow tag_1 \cdot tag_2 \cdot tag_3 \tag{4}$$

and the following symbol sequence exists in the corpus:

$$\ldots tag_1 \cdot tag_2 \cdot tag_3 \ldots$$

In this case both of the rules are applicable. To avoid this conflict, the system applies rewriting rules in the order of the length of their right-hand side. In this example, only rule (4) is applied.

4 Results

We conducted experiments on the sentences in WSJ, which are composed of the POS tags in Table 1. The corpus contains 24,678 sentences and 549,247 tags. The average sentence length is, therefore, 22.3 tags. Figure 3 shows the distribution by length of sentences in the corpus.

We ran two experiments with different threshold values for the distance between two distributions (Exp. 1, $D_{min} = 0.20$ and Exp. 2, $D_{min} = 0.25$). The output of each experiment is the final state of the corpus and the extracted rewriting rules.

4.1 Accuracy of Parsing

First we discuss the final state of the corpus, which can be regarded as the parsing results. Figure 4 shows an example of a parsed sentence, where the symbols starting with "NT" are the non-terminal symbols introduced by the system.

For the evaluation of our experiments we have chosen the measure called "crossing parenthesis accuracy" which arose out of a parser evaluation workshop [Black *et al.*, 1991]. The measure is the percentage of POS constituents as output by our system which do not cross any

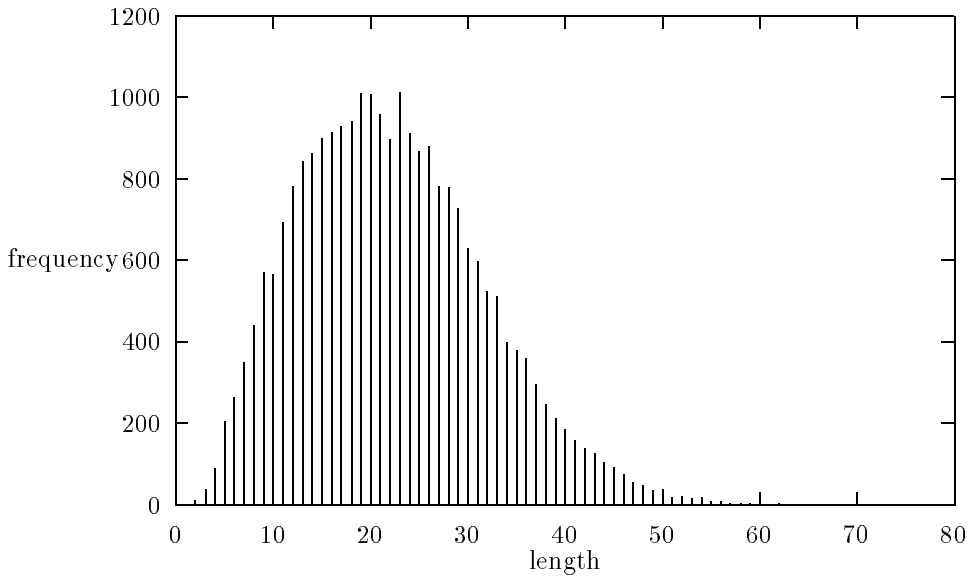

Figure 3: Sentence length distribution

constituents as parsed in the Penn Treebank. Table 4 shows the crossing parenthesis accuracy of our method. In this table, "right linear" means the right binary branching structure and "left linear" means the left binary branching structure.

Table 4: Result

	# rules	# N	C.P.A.
right linear	—	—	56.3%
left linear	—	—	24.3%
Exp. 1	956	19	73.7%
Exp. 2	594	25	74.8%

C.P.A. = Crossing Parenthesis Accuracy

Since this method parses the corpus by applying rewriting rules which are extracted only from the POS sequence information, it is quite natural that the accuracy is less than that of grammar inference methods profiting from syntactic structure information [Pereira and Schabes, 1992] [Schabes *et al.*, 1993] [Brill, 1993]. In addition to the difference in source information, it must be noted that those methods were tested only on relatively short sentences, while in our experiments sentence length is not limited. From this viewpoint one may say that the accuracy of our method is sufficiently promising.

We must draw attention to another difference between our method and the others. They simply bracket sentences and are not able to introduce non-terminal symbols, while our method is able to infer non-terminal symbols without any information but a corpus.

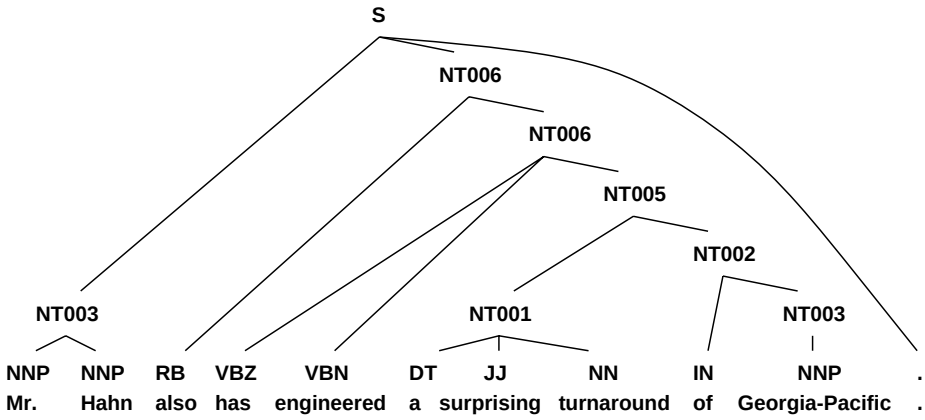

Figure 4: An example of a parsed sentence (Exp. 2)

4.2 Evaluation of Rewriting Rules

The other output of our system is a set of rewriting rules. Its quantitative evaluation is so difficult that we compare them with a general English grammar.

Table 5 shows some rewriting rules extracted in Experiment 2. Almost all the rules here are comprehensible from the viewpoint of general English grammar: $NT001$ represents noun phrase, $NT002$ prepositional phrase and so on. The interesting thing in this table is that the rewriting rules whose left-hand side is $NT006$ have various combinations of noun phrases and/or prepositional phrases on their right-hand side. They must be considered as verbal case frames. This indicates that it is possible to extract types of verbal case frames, which is defined a priori in various attempts at automatic case frame acquisition [Brent and Berwick, 1991] [Brent, 1991] [Manning, 1993].

Elsewhere, however, the grammar contains some inappropriate rewriting rules, as follows:

$$NT002 \rightarrow IN \cdot JJR \cdot NNS \cdot NNS$$

It would be better to represent this by the following rules instead.

$$NT002 \rightarrow IN \cdot NT001 \quad NT001 \rightarrow JJR \cdot NNS \cdot NNS$$

It is true that there are some rewriting rules like this in the grammar but, as we mentioned above, almost all the non-terminal symbols are comprehensible. Since our method assumes simple context-free language as the model and extracts a grammar based on hypotheses deduced from its characteristics, these results lead to the conclusion that structural features of language can be extracted using simple statistical methods.

It should also be added that it is difficult to evaluate a grammar because the evaluation depends on the language model of the parsed corpus to be compared.

5 Conclusions

In this paper, we have described a new method to parse a tagged corpus without grammar, modeling a natural language on context-free language. We proposed the following three hypotheses.

Table 5: Some rewriting rules extracted in Experiment 2

$NT001$	$\rightarrow$	$PRP\$ \cdot NNS \cdot NNS$	$NT005$	$\rightarrow$	$VBN \cdot NT002$
$NT001$	$\rightarrow$	$PDT \cdot DT \cdot NNS \cdot NN$	$NT005$	$\rightarrow$	$VBG \cdot NT001$
$NT001$	$\rightarrow$	$NNP \cdot NNP \cdot POS \cdot NN$	$NT005$	$\rightarrow$	$PRP \cdot MD \cdot VB \cdot NT002$
$NT001$	$\rightarrow$	$DT \cdot VBG \cdot NNS \cdot NNS$	$NT006$	$\rightarrow$	$VBZ \cdot NT005 \cdot NT005$
$NT001$	$\rightarrow$	$DT \cdot JJ \cdot NN$	$NT006$	$\rightarrow$	$VBZ \cdot NT005 \cdot NT002$
$NT002$	$\rightarrow$	$TO \cdot NT001$	$NT006$	$\rightarrow$	$VBZ \cdot NT005 \cdot NT001$
$NT002$	$\rightarrow$	$TO \cdot VB \cdot NT001$	$NT006$	$\rightarrow$	$VBZ \cdot NT005$
$NT002$	$\rightarrow$	$TO \cdot VB \cdot NNS$	$NT006$	$\rightarrow$	$VBZ \cdot NT002 \cdot NT005$
$NT002$	$\rightarrow$	$RB \cdot TO \cdot NT001$	$NT006$	$\rightarrow$	$VBZ \cdot NT002$
$NT002$	$\rightarrow$	$IN \cdot NT001$	$NT006$	$\rightarrow$	$VBZ \cdot NT001 \cdot NT001$
$NT002$	$\rightarrow$	$IN \cdot RB \cdot NT001$	$NT006$	$\rightarrow$	$VBZ \cdot NT001$
$NT002$	$\rightarrow$	$IN \cdot NNS \cdot NN$	$NT006$	$\rightarrow$	$VBD \cdot NT005 \cdot NT005$
$NT002$	$\rightarrow$	$TO \cdot NT003$	$NT006$	$\rightarrow$	$VBD \cdot NT002$
$NT002$	$\rightarrow$	$RB \cdot NT002$	$NT006$	$\rightarrow$	$MD \cdot VB \cdot NT001$
$NT002$	$\rightarrow$	$RB \cdot IN \cdot NT003$	$NT006$	$\rightarrow$	$MD \cdot VB \cdot NT001$
$NT002$	$\rightarrow$	$IN \cdot NT003$	$NT007$	$\rightarrow$	$VBZ \cdot VBN \cdot VBN$
$NT003$	$\rightarrow$	$NNP \cdot NNP \cdot NNP$	$NT007$	$\rightarrow$	$MD \cdot VB$
$NT003$	$\rightarrow$	$NNP \cdot NNP$	$NT008$	$\rightarrow$	$NNS \cdot NT006$
$NT003$	$\rightarrow$	$NT003 \cdot CC \cdot NT003$	$NT008$	$\rightarrow$	$NNS \cdot VBP \cdot NT005$
$NT004$	$\rightarrow$	$NN \cdot NN$	$NT008$	$\rightarrow$	$NNS \cdot VBP \cdot NT001$
$NT004$	$\rightarrow$	$JJ \cdot NN \cdot NN$	$NT009$	$\rightarrow$	$NT004 \cdot NT008$
$NT005$	$\rightarrow$	$NT002 \cdot VBN \cdot NT002$	$NT009$	$\rightarrow$	$NT003 \cdot JJ \cdot NT008$
$NT005$	$\rightarrow$	$NT001 \cdot NT002$	$NT010$	$\rightarrow$	$WRB \cdot PRP \cdot NT006$
$NT005$	$\rightarrow$	$NT001 \cdot VBN \cdot NT002$	$NT010$	$\rightarrow$	$VBG \cdot NT004 \cdot NT006$
$NT005$	$\rightarrow$	$NT001 \cdot VBG \cdot NT002$	$NT010$	$\rightarrow$	$VBG \cdot NT004$

1. POS sequences on the right-hand side of a rewriting rule are less constrained as to what POS precedes and follows them than non-constituent sequences.

2. POS sequences directly derived from the same non-terminal symbol have similar environments.

3. The most suitable set of rewriting rules makes the greatest reduction of the corpus size.

Then, we developed an algorithm based on these hypotheses which extracts rewriting rules and parses the sentences at the same time. The correctness of these hypotheses has been experimentally attested by the evaluation of the extracted grammar and parsing accuracy.

The method we have proposed in this paper is a bottom-up approach to grammar inference. On the other hand, a top-down approach such as [Magerman and Marcus, 1990] may also be important for grammar induction. Clustering techniques, e.g. [Hindle, 1990] and subcategorization techniques, e.g. [Brent, 1991] may bridge the gap between these two approaches. Combining these techniques including tagging techniques, larger amounts of syntactic information can be retrieved from unbracketed corpora or even from untagged corpora.

References

[Black *et al.*, 1991] E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and

T. Strzalkowski. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of the DARPA Speech and Natural Language Workshop*, 1991.

[Brent and Berwick, 1991] Michael R. Brent and Robert C. Berwick. Automatic acquisition of subcategorization frames from tagged text. In *Proceedings of the DARPA Speech and Natural Language Workshop*, 1991.

[Brent, 1991] Michael R. Brent. Automatic acquisition of subcategorization frames from untagged text. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, 1991.

[Brill and Marcus, 1992] Eric Brill and Mitchell Marcus. Automatically acquiring phrase structure using distributional analysis. In *Proceedings of the DARPA Speech and Natural Language Workshop*, 1992.

[Brill, 1992] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, 1992.

[Brill, 1993] Eric Brill. Automatic grammar induction and parsing free text: A transformation-based approach. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 1993.

[Church, 1988] Kenneth Ward Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, 1988.

[Hindle, 1990] Donald Hindle. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, 1990.

[Magerman and Marcus, 1990] David M. Magerman and Mitchell P. Marcus. Parsing a natural language using mutual information statistics. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, 1990.

[Manning, 1993] Chiristopher D. Manning. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 1993.

[Marcus and Santorini, 1993] Mitchell P. Marcus and Beatrice Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), 1993.

[Nagao and Mori, 1994] Makoto Nagao and Shinsuke Mori. A new method of n-gram statistics for large number of n and automatic extraction of words and phrases from large text data of Japanese. In *Proceedings of the 15th International Conference on Computational Linguistics*, 1994.

[Pereira and Schabes, 1992] Fernando Pereira and Yves Schabes. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, 1992.

[Schabes *et al.*, 1993] Yves Schabes, Michal Roth, and Randy Osborne. Parsing the Wall Street Journal with the inside-outside algorithm. In *Proceedings of the Sixth European Chapter of the Association for Computational Linguistics*, 1993.