Incremental CFG Parsing with Statistical Lexical Dependencies

Takahisa Murase†, **Shigeki Matsubara‡**, **Yoshihide Kato†** and **Yasuyoshi Inagaki†**
Graduate School of Engineering, Nagoya University †
Faculty of Language and Culture, Nagoya University ‡
Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan
murase,matu,yosihide,inagaki @inagaki.nuie.nagoya-u.ac.jp

Abstract

Incremental parsing with a context free grammar produces partial syntactic structures for an initial fragment on the word-by-word basis. Owing to the syntactic ambiguity, however, too many structures are produced, and therefore its parsing speed becomes very slow. This paper describes a technique for efficient incremental parsing using lexical information. The probability concerning dependencies between words, as the lexical information, is automatically acquired from a large-scale corpus with syntactic structures. A process for discarding syntactic structures which will not be likely has been integrated into the incremental chart parsing. That is, partial syntactic structures whose dependency probabilities are not high will be removed from the chart. Our technique proposed in this paper can also be considered as a kind of practical methods of incremental disambiguation. An experiment using Penn Treebank has shown our technique to be feasible and efficient.

1 Introduction

With the advances of speech processing technologies in recent years, it has been tried to develop systems which understand speech simultaneously with it (Ehsani et al., 1994; Matsubara et al., 1999; Nakano et al., 1999). An incremental parsing is one of essential techniques for such the systems, because it is useful for the system to get syntax-relations of input sentences at an early stage of the input. Several techniques for incremental parsing have been proposed so far (Akiba and Tanaka, 1992; Matsubara et al., 1997; Milward, 1995; Steedman, 1987). However, producing the syntactic structures word-by-word makes the parsing process inefficient in comparing with the usual parsing method which has only to produce the structures for one sentence.

In order to overcome this problem, several works have been done. Akiba et al. proposed a technique referring obligatory case information (Akiba et al., 1993), and Haddock studied a technique examining consistency between the conversational situation and the input content (Haddock, 1987). We can consider that the former uses a semantic information and the latter a contextual one to solve the syntactic ambiguity incrementally. However, it is not easy to acquire such the advanced linguistic knowledge.

On the other hand, many researches have used the statistical information to increase the efficiency of parsing (Charniak, 1997; Collins, 1996; Magerman, 1995; Ratnaparkhi, 1997; Stolcke, 1995). The common idea of (Charniak, 1997; Collins, 1996; Magerman, 1995) is to assign a probability to every syntactic structure for one sentence, and to find the most likely one. In contrast, Imaichi et al., and Watanabe have proposed a technique to avoid producing unlikely syntactic structures by using statistical lexical dependency information (Imaichi et al., 1998; Watanabe, 1999). What needs to be noted is that all of these techniques mentioned here are intended for parsing of a whole sentence, not a partial one.

This paper proposes an efficient incremental parsing technique by using statistical lexical information. Our technique is based on the incremental chart parsing method, and it determines dependency probabilities of partial syntactic structures which will be produced on the way of incremental parsing, and it statistically evaluates the semantical likelihood of them. The statistical information of the lexical dependency relation for initial fragments is useful to attain a more efficient incremental parsing. We have used the Penn Treebank (Marcus et al., 1993) to make an experiment. This has shown our technique to be feasible and efficient.

This paper is organized as follows: Section 2 explains the incremental chart parsing. Section 3 describes the parsing method using statistical dependency information. Section 4 reports the experimental results.

2 Incremental Parsing

2.1 Incremental chart parsing

The incremental chart parsing (Matsubara et al., 1997), which is one of the syntax analysis techniques using a context free grammar, expresses the result in progress as the graph called *chart* (Kay, 1980). The *node* labeled the number exists between words in an input sentence, and the *edge* stretched between nodes is labeled a syntactic structure called *term*. The term $[?]_X$ is called an *undecided term* and represents that the part has not been produced yet. The leftmost occurrence of an undecided term in a term is called a *leftmost undecided term*. If an edge is labeled a term which includes an undecided term, it is called *active*, otherwise *inactive*.

The incremental chart parsing starts on the chart in which the edge labeled term $[?]_S$ was stretched from the node 0 to itself. When i-th word w_i is produced, the following procedures are performed in order:

a) Consultation of a dictionary: If the category of a word w_i is A, the edge labeled term $[w_i]_A$ is stretched between nodes $i-1$ and i in a chart.

b) Application of grammar rules: If the term of the edge stretched between nodes $i-1$ and i in a chart is $[\cdots]_{A_1}$ and a grammar rule $A \to A_1\ A_2 \cdots A_n$ exists, the edge labeled term $[[\cdots]_{A_1}[?]_{A_2}\cdots[?]_{A_n}]_A$ is added between nodes $i-1$ and i in a chart. As much as possible, this operation is repeated.

c) Replacement of terms: Let $[?]_X$ be the leftmost undecided term of the term σ of the edge stretched between nodes 0 and $i-1$ in a chart. If the category of the term τ of the edge stretched between $i-1$ and i is X, the edge labeled the term produced by replacing the leftmost undecided term of σ with τ is added between nodes 0 and i in a chart.

That is, the incremental chart parsing introduces two new operations into the standard bottom-up chart parsing. One is the application of grammar rules to an active edge and the other is the replacement of the leftmost undecided term with the term of an active edge. The operations enable the parsing to necessarily construct syntactic structures for initial fragments of grammatical input sentences. But, the amount of calculation increases in comparing with the usual parsing method. Therefore, it takes much time for parsing.

2.2 The process of incremental parsing

In the incremental chart parsing, new structures are produced word-by-word through the replacement operation. As an example, let us consider incremental chart parsing for an English sentence:

$$I\ need\ a\ flight\ from\ Atlanta\ to\ Charlotte. \quad (1)$$

using the grammar and dictionary in Table 1. Its parsing process is shown in Table 2. Each line corresponds to the edge in a chart, the number of the columns of # corresponds to the production order of the edge, the *loc* expresses the position of the edge, and the *term* expresses the syntactic structure. For example, by replacing the leftmost undecided term of a term $[[[I]_{prp}]_{np}[?]_{vp}]_s$ for "I" with a term $[[need]_{vbp}[?]_{np}]_{vp}$ for "need", a term $[[[I]_{prp}]_{np}[[need]_{vbp}[?]_{np}]_{vp}]_s$ for a word sequence "I need" is produced. Figure 1 expresses the relation between the produced terms as a graph. This graph can be regarded as a kind of the search trees in which a node corresponds to a term (Kato et al., 2000). The node in the search tree is derived from its parent node by adding the information about the input word to the syntactic structure of the parent node. That is, incremental chart parsing makes the tree grow up word-by-word, and pruning the tree makes the parsing process efficient.

Table 1: Grammar rules and dictionary

⟨grammar rules⟩	⟨ dictionary⟩
$s \to np\ vp$	$prp \to I$
$np \to prp$	$vbp \to need$
$vp \to vbp\ np'\ pp$	$dt \to a$
$np' \to dt\ nn$	$nn \to flight$
$pp \to p\ np$	$p \to from$
$np \to nnp\ pp$	$nnp \to Atlanta$
$np \to np'\ pp$	$p \to to$
$np \to nnp$	$nnp \to Charlotte$
$vp \to vbp\ np$	

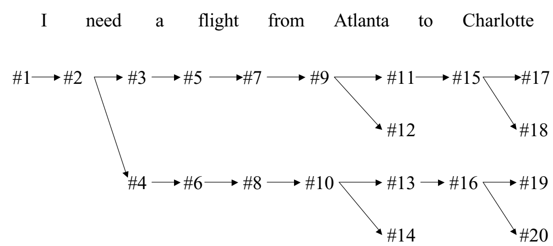

Figure 1: Process of incremental parsing

3 Incremental Parsing with Lexical Dependency Information

In this section, we propose a technique for pruning the search tree discussed in the previous section, that is, for avoiding the production of unlikely syntactic structures by using the dependency information between words in incremental

Table 2: parsing process for "I need a flight from Atlanta to Charlotte"

word	#	*loc*	*term*
	1	0-0	$[?]_s$
I	2	0-1	$[[[I]_{prp}]_{np}[?]_{vp}]_s$
need	3	0-2	$[[[I]_{prp}]_{np}[[need]_{vbp}[?]_{np'}[?]_{pp}]_{vp}]_s$
	4	0-2	$[[[I]_{prp}]_{np}[[need]_{vbp}[?]_{np}]_{vp}]_s$
a	5	0-3	$[[[I]_{prp}]_{np}[[need]_{vbp}[[a]_{dt}[?]_{nn}]_{np'}[?]_{pp}]_{vp}]_s$
	6	0-3	$[[[I]_{prp}]_{np}[[need]_{vbp}[[[a]_{dt}[?]_{nn}]_{np'}[?]_{pp}]_{np}]_{vp}]_s$
ight	7	0-4	$[[[I]_{prp}]_{np}[[need]_{vbp}[[a]_{dt}[flight]_{nn}]_{np'}[?]_{pp}]_{vp}]_s$
	8	0-4	$[[[I]_{prp}]_{np}[[need]_{vbp}[[[a]_{dt}[flight]_{nn}]_{np'}[?]_{pp}]_{np}]_{vp}]_s$
from	9	0-5	$[[[I]_{prp}]_{np}[[need]_{vbp}[[a]_{dt}[flight]_{nn}]_{np'}[[from]_p[?]_{np}]_{pp}]_{vp}]_s$
	10	0-5	$[[[I]_{prp}]_{np}[[need]_{vbp}[[[a]_{dt}[flight]_{nn}]_{np'}[[from]_p[?]_{np}]_{pp}]_{np}]_{vp}]_s$
Atlanta	11	0-6	$[[[I]_{prp}]_{np}[[need]_{vbp}[[a]_{dt}[flight]_{nn}]_{np'}[[from]_p[[Atlanta]_{nnp}[?]_{pp}]_{np}]_{pp}]_{vp}]_s$
	12	0-6	$[[[I]_{prp}]_{np}[[need]_{vbp}[[a]_{dt}[flight]_{nn}]_{np'}[[from]_p[[Atlanta]_{nnp}]_{np}]_{pp}]_{vp}]_s$
	13	0-6	$[[[I]_{prp}]_{np}[[need]_{vbp}[[[a]_{dt}[flight]_{nn}]_{np'}[[from]_p[[Atlanta]_{nnp}[?]_{pp}]_{np}]_{pp}]_{np}]_{vp}]_s$
	14	0-6	$[[[I]_{prp}]_{np}[[need]_{vbp}[[[a]_{dt}[flight]_{nn}]_{np'}[[from]_p[[Atlanta]_{nnp}]_{np}]_{pp}]_{np}]_{vp}]_s$
to	15	0-7	$[[[I]_{prp}]_{np}[[need]_{vbp}[[a]_{dt}[flight]_{nn}]_{np'}[[from]_p[[Atlanta]_{nnp}[[to]_p[?]_{np}]_{pp}]_{np}]_{pp}]_{vp}]_s$
	16	0-7	$[[[I]_{prp}]_{np}[[need]_{vbp}[[[a]_{dt}[flight]_{nn}]_{np'}[[from]_p[[Atlanta]_{nnp}[[to]_p[?]_{np}]_{pp}]_{np}]_{pp}]_{np}]_{vp}]_s$
Charlotte	17	0-8	$[[[I]_{prp}]_{np}[[need]_{vbp}[[a]_{dt}[flight]_{nn}]_{np'}[[from]_p[[Atlanta]_{nnp}[[to]_p[[Charlotte]_{nnp}[?]_{pp}]_{np}]_{pp}]_{np}]_{pp}]_{vp}]_s$
	18	0-8	$[[[I]_{prp}]_{np}[[need]_{vbp}[[a]_{dt}[flight]_{nn}]_{np'}[[from]_p[[Atlanta]_{nnp}[[to]_p[[Charlotte]_{nnp}]_{np}]_{pp}]_{np}]_{pp}]_{vp}]_s$
	19	0-8	$[[[I]_{prp}]_{np}[[need]_{vbp}[[[a]_{dt}[flight]_{nn}]_{np'}[[from]_p[[Atlanta]_{nnp}[[to]_p[[Charlotte]_{nnp}[?]_{pp}]_{np}]_{pp}]_{np}]_{pp}]_{np}]_{vp}]_s$
	20	0-8	$[[[I]_{prp}]_{np}[[need]_{vbp}[[[a]_{dt}[flight]_{nn}]_{np'}[[from]_p[[Atlanta]_{nnp}[[to]_p[[Charlotte]_{nnp}]_{np}]_{pp}]_{np}]_{pp}]_{np}]_{vp}]_s$

chart parsing.

3.1 The Mapping from a Syntactic Structure to Dependencies

This section describes Collins' method of mapping from a syntactic structure to the word dependencies shortly (Collins, 1996).

In the method, each rule has one special category called *head*, which is annotated by the sign '*'. The other categories in right-hand side are called *complement*. The dependency relation that a complement is the dependent on the head can be recognized in each syntactic structure. For a term:

$$[[\cdots]_{Y_1}\cdots[\cdots]_{Z^*}\cdots[\cdots]_{Y_n}]_X \qquad (2)$$

if a grammar rule "$X \rightarrow Y_1 \cdots Z^* \cdots Y_n$" exists, $[\cdots]_Z$ is called *head term* of the term (2). Each term has a *head-word* which is defined as follows:

1. Let w_i be a word and X be a category, then the head-word of the term $[w_i]_X$ is w_i.
2. The head-word of an undecided term $[?]_X$ is "?".
3. Let α_i be $[\cdots]_{A_i}(i = 1, \ldots, n)$, and α be a term $[\alpha_1 \cdots \alpha_i \cdots \alpha_n]_A$. If a grammar rule $A \rightarrow A_1 \cdots A_i^* \cdots A_n$ exists, then the head-word of α is the head-word of α_i.

The head-word holds the main meaning of the term. The head-word propagates up through the syntactic structure, each term receiving its head-word from its head term. The dependency between the constituents can be extracted from the syntactic structure produced using the grammar rule with the head and complements. According to it, the dependency relation between words can also be obtained. That is, in the term (2) produced by applying a grammar rule "$X \rightarrow Y_1 \cdots Z^* \cdots Y_n$", the head-word of $[\cdots]_{Y_j}$ $(1 \leq j \leq n)$ is the dependent on the head-word of $[\cdots]_Z$.

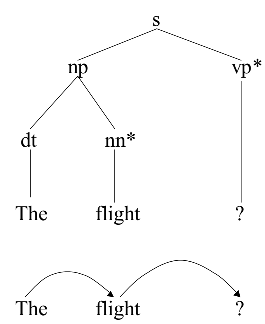

Figure 2: The dependency containing "?"

3.2 Dependency probability

Collins has proposed the probabilistic model based on the lexical dependency probability (Collins, 1996). The probabilistic model assigns a probability to every syntactic structure. Formally, given a sentence $S(= w_1 \cdots w_n)$ and a syntactic structure T, the model estimates the conditional probability $P(T|S)$ as follows:

$$P(T|S) \;=\; \prod_{j=1}^{n} P(w_j \stackrel{rel}{\rightarrow} w_{h_j}|S) \qquad (3)$$

$P(w_j \stackrel{rel}{\rightarrow} w_{h_j}|S)$ is the probability that w_j depends on w_{h_j}. It is defined as follows:

$$P(w_j \stackrel{rel}{\rightarrow} w_{h_j}|S) \stackrel{\mathrm{def}}{=} \frac{C(w_j \stackrel{rel}{\rightarrow} w_{h_j}, t_j, t_{h_j}, d_{jh_j}, c_{jh_j})}{C(w_j, w_{h_j}, t_j, t_{h_j}, d_{jh_j}, c_{jh_j})} \qquad (4)$$

Here, C is the coincidence frequency function. And t_j is the part-of-speech of the word w_j, t_{h_j} is

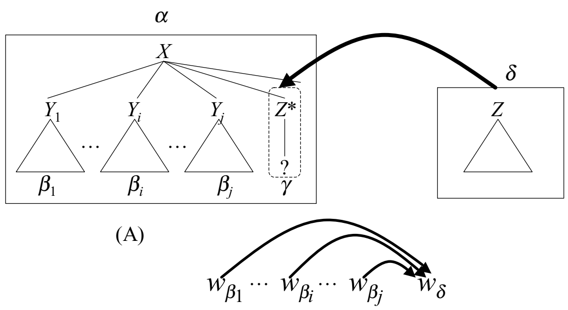

Figure 3: Case 1: leftmost undecided term γ is the head term of α

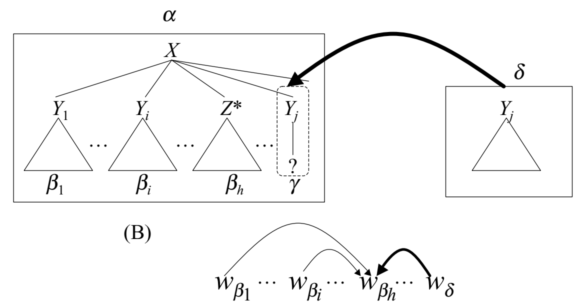

Figure 4: Case 2: leftmost undecided term γ is located on the right of the head term of α

that of w_{h_j}. d_{jh_j} is the distance between w_j and w_{h_j}. c_{jh_j} is the number of the commas between w_j and w_{h_j}.

3.3 Dependency probability for partial syntactic structure

Collins' probabilistic model should be modified to be used for incremental parsing, because the model cannot estimate the dependency probability in the case where the head word or the dependent is "?" as shown in Figure 2. In this paper, we regard the probability of such a dependency as 1. That is, for an initial fragment $W(= w_1 \cdots w_i)$ and a partial syntactic structure T, we define the conditional probability $P(T|W)$ as follows:

$$P(T|W) = \prod_{j=1}^{i} \hat{P}(w_j \stackrel{rel}{\rightarrow} w_{h_j}|W) \qquad (5)$$

$$\hat{P}(w_j \stackrel{rel}{\rightarrow} w_{h_j}|W) = \begin{cases} 1 & (if \quad h_j > i) \\ P(w_j \stackrel{rel}{\rightarrow} w_{h_j}|W) & (otherwise) \end{cases} \qquad (6)$$

When the input is completed, the probability $P(T|W)$ is the same as the one in Collins' model.

3.4 Pruning

We take the method of finding the syntactic structures which have a low cost rather than that of comparing the score between syntactic structures.

In our technique, we use the following threshold Θ which is dynamically varied dependently on the number of input words, i.e. i.

$$\Theta = \theta^i \quad (0 \leq \theta \leq 1) \qquad (7)$$

Here, θ is the threshold for a dependency probability of a pair of words. If $P(T|W) \leq \Theta$, the syntactic structure is judged to be unlikely.

3.5 Utilizing dependency probability

In this section, we describe the technique for controlling the replacement operation of terms in

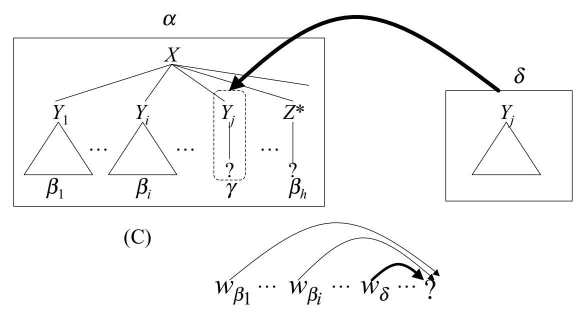

Figure 5: Case 3: leftmost undecided term γ is located on the left of the head term of α

incremental chart parsing by using the probabilistic model mentioned in section 3.3 and avoiding the production of unlikely syntactic structures.

In incremental chart parsing, replacing terms is equivalent to forming a new dependency relation between words. If the dependency formed newly by replacement operation can be gotten, the likelihood of the syntactic structure can be examined.

The replacement operation of terms can be classified into three cases based on the formation of the dependency. Figure 3 shows the first case. When the leftmost undecided term γ of a term α is replaced with a term δ, the dependency (A) will be formed (w_{β_i} is the head-word of a term β_i, and w_δ is that of δ.). In Figure 3, since γ is the head term, it is newly formed that w_{β_i} $(1 \leq i \leq j)$ is the dependent on w_δ. Figure 4 shows the second case. When γ is replaced with δ, the dependency (B) will be formed. In Figure 4, since γ is located on the right of the head term β_h, it is newly formed that w_δ is the dependent on w_{β_h}. Figure 5 shows the third case. The dependency (C) will be formed in this case. Since γ is located on the left of β_h, the dependency containing "?" is newly formed. Moreover, not concerned with these cases, when the head-word of δ is "?", the relation that δ depends on "?" is formed. By clas-

Table 3: Grammar rules and dictionary

⟨grammar rules⟩	⟨ dictionary⟩
$s \quad \rightarrow np\ vp^*$	$prp \rightarrow I$
$np \rightarrow prp^*$	$vbp \rightarrow need$
$vp \rightarrow vbp^*\ np'\ pp$	$dt \rightarrow a$
$np' \rightarrow dt\ nn^*$	$nn \rightarrow flight$
$pp \rightarrow p^*\ np$	$p \rightarrow from$
$np \rightarrow nnp^*\ pp$	$nnp \rightarrow Atlanta$
$np \rightarrow np'^*\ pp$	$p \rightarrow to$
$np \rightarrow nnp^*$	$nnp \rightarrow Charlotte$
$vp \rightarrow vbp^*\ np$	

sifying replacement operation into three cases, it can be turn out what dependencies will be formed.

Let us consider replacing the leftmost undecided term of σ with the term τ. Let the probability for σ be P_σ. The probability for the new term is calculated by the product of P_σ and the probability of the dependency which will be newly formed. If the probability for the new term is lower than the threshold, the replacement of terms is prevented.

3.6 Example

The parsing of an English sentence (1) is as follows: The grammar and dictionary shown in Table 3 are used.

Let us consider the replacement shown in Figure 6. The dependency between the head-words "need" of the term $[need]_{vbp*}$ and "from" of $[[from]_{in*}[?]_{np}]_{pp}$ will be formed in case 2 mentioned above. If let the dependency probability $P(from \stackrel{rel}{\rightarrow} need \mid I\ need\ a\ flight\ from)$ be 3.60×10^{-6}, and the probability for σ be 0.98, then the probability for the new term σ' is calculated like $0.98 \times 3.60 \times 10^{-6} = 3.52 \times 10^{-6}$. If the threshold is $\Theta = 0.2^5$, then $0.2^5 > 3.52 \times 10^{-6}$. By preventing the replacement of terms, the syntactic structure #9 in Table 2 is not produced.

4 Experiment

In order to evaluate the effectiveness of the technique proposed in this paper, we have made an experiment using Penn Treebank. We implemented it in GNU Common Lisp 2.3.6 on a Linux workstation (CPU: Intel Pentium Xeon III 1GHz, Memory: 1024MB).

4.1 Outline of the experiment

We used 52 sentences of a spoken language corpus, ATIS corpus as the open test data. The average length is 8.4 words. Figure 7 shows the relationship between the sentence length and the number of sentences. The grammar and lexical rules has been acquired from the sentences with the syntactic structures in ATIS corpus. For the purpose of the parsing speed, we used the grammar

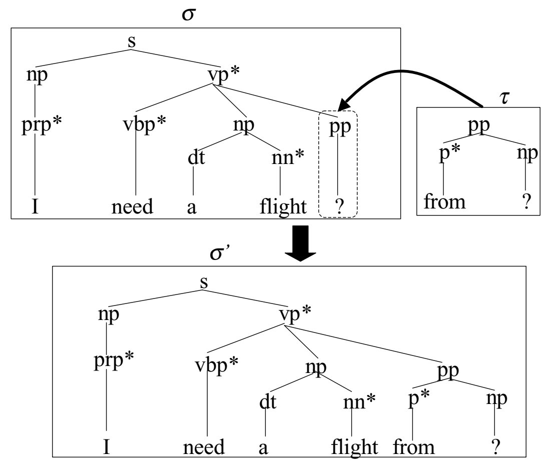

Figure 6: Example of calculating the probability of a syntactic structure

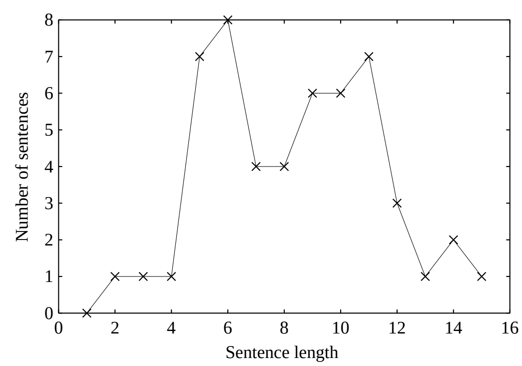

Figure 7: Sentence length distribution of our test data from ATIS corpus

rules which are transformed according to a grammar transfer technique (Johnson, 1998). Every non-terminal symbol included in syntactic structures in the corpus is copied the parent node onto. Thus, we used 508 grammar rules and 363 lexical rules. We have measured the number of the syntactic structures and the parsing time word-by-word in both the conventional incremental chart parsing and our technique. We set $0, \ldots, 0.3$ as θ of the threshold (7).

As the training data, we used both 520 sentences of ATIS corpus, and in addition, 38,438 sentences of WSJ corpus to compensate the sparse data.

4.2 Experimental result

Figure 8 shows the relation between the sentence length and the average number of syntactic structures. The longer the sentences are, the more syntactic structures are produced in the incremental chart parsing. As compared with it, it is in-

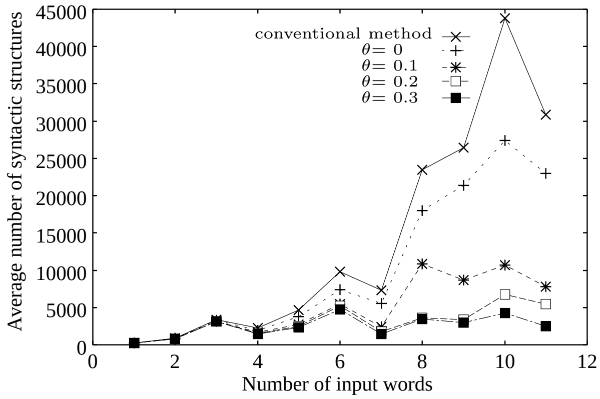

Figure 8: Average number of syntactic structures

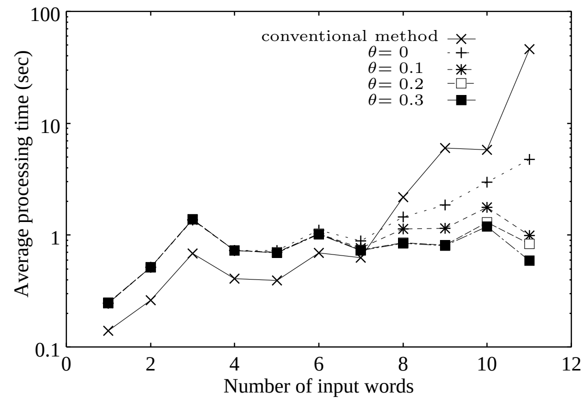

Figure 9: Average processing time

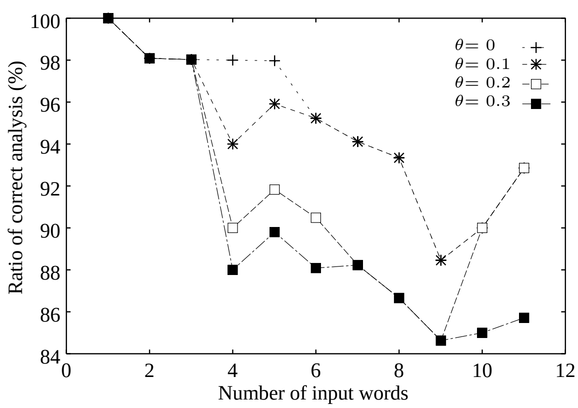

Figure 10: Recall of syntactic structures

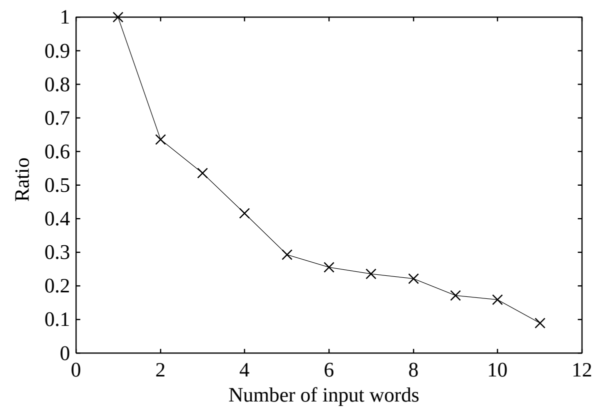

Figure 11: Ratio of the average number of dependencies containing "?" to the number of all possible dependencies

dicated that our technique is useful for reducing the syntactic structures. This fact brings an effect on the processing speed. Figure 9 shows the relation between the sentence length and the average processing time of parsing one word. As Figure 9 shows, even if it takes much time to parsing a word in the incremental parsing, it may not be so in our technique. Moreover, the recall of the correct syntactic structures is shown in Figure 10. This result indicates that many correct structures were judged to be likely. From the experimental result, we have confirmed the effectiveness of our technique.

In our technique, however, there is a problem to be pointed out here. Figure 9 indicates that the technique using the lexical dependency may not be efficient for the short initial fragment. In an early step of the input, many dependencies mapped from syntactic structures contain the words which has not been produced yet. We defined the probability of such the dependencies as 1. So many syntactic structures are judged to be likely semantically. Figure 11 shows the ratio of the average number of dependencies which contain a "?" to the number of input words. The longer the fragments are, the smaller the ratio becomes.

Moreover, as the evaluation criteria, we used the PARSEVAL measures (Black et al., 1991) for complete syntactic structures which were produced for an input sentence by our technique. Table 4 shows the results. We give results using six measures: the percentage of sentences for which a syntactic structure was produced, **LR** (labeled recall), **LP** (labeled precision), **CBs** (the average number of crossing brackets per sentence), **0CBs** (the percentage of sentences with 0 crossing brackets), **≤2CBs** (the percentage of sentences with ≤2 crossing brackets). We can then additionally evaluate the parser's performance to the maximum likelihood syntactic structure (MLS). As we can see, our technique is a viable broad-coverage statistical one.

5 Conclusion

In this paper, we have proposed the technique for efficient incremental parsing using the statistical

Table 4: Evaluation on ATIS corpus

θ	Per. of Sen.	All Produced Structures					MLS				
		LR	LP	CBs	0CBs	≤2CBs	LR	LP	CBs	0CBs	≤2CBs
0	96.2%	59.5%	49.9%	1.84	37.2%	71.7%	92.9%	90.8%	0.26	94.0%	96.0%
0.1	92.3%	71.2%	63.7%	1.16	54.4%	82.0%	93.2%	90.8%	0.27	93.8%	95.8%
0.2	90.4%	76.4%	69.5%	0.75	65.6%	90.2%	93.1%	90.6%	0.28	93.6%	95.7%
0.3	82.7%	79.2%	72.4%	0.62	74.4%	91.1%	95.1%	92.6%	0.14	97.7%	97.7%

lexical dependency information. Utilizing the dependency between words enables the technique to get out of producing semantically unlikely syntactic structures. An experimental result has demonstrated that the parsing time becomes shorter because of the reduction of the number of syntactic structures.

Acknowledgment

The authors would like to thank all members of Nagoya University Corpus Project for their helpful suggestions. This research has been supported in part by Kayamori Foundation of Information Science Advancement.

References

T. Akiba and H. Tanaka. 1992. A Method for Incremental Parsing using Extended Partial Trees. *The 45th National Convention of Information Processing Society of Japan*, 3:175–176. (in Japanese)

T. Akiba, K. Itou, M. Okumura, and H. Tanaka. 1993. A Japanese Analysis based on Incremental Disambiguation Model. *Computer Software*, 10(1):29–40. (in Japanese)

E. Black et al. 1991. A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars. *Proceedings of the February 1991 DARPA Speech and Natural Language Workshop*, pages 306–311.

E. Charniak. 1997. Statistical Parsing with a Context-free Grammar and Word Statistics. *Proceedings of the 14th National Conference on Artificial Intelligence*, pages 598–603.

M. J. Collins. 1996. A New Statistical Parser Based on Bigram Lexical Dependencies. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 184–191.

F. Ehsani, K. Hatazaki, J. Noguchi, and T. Watanabe. 1994. Interactive Speech Dialogue System Using Simultaneous Understanding. *Proceedings of the 3rd International Conference on Spoken Language Processing*, 2:879–882.

J. Haddock. 1987. Incremental Interpretation and Combinatory Categorial Grammar. *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 661–663.

O. Imaichi, Y. Matsumoto, and M. Fujio. 1998. An Integrated Parsing Method using Stochastic Information and Grammatical Constraints. *Journal of Natural Language Processing*, 5(3):67–83. (in Japanese)

M. Johnson. 1998. PCFG Models of Linguistic Tree Representations. *Computational Linguistics*, 24:617–636.

Y. Kato, S. Matsubara, K. Toyama, and Y. Inagaki. 2000. Spoken Language Parsing based on Incremental Disambiguation. *Proceedings of the 6th International Conference on Spoken Language Processing*, 2:999–1002.

M. Kay. 1980. Algorithm Schemata and Data Structures in Syntactic Processing. *Technical Report CSL-80-12*.

D. Magerman. 1995. Statistical Decision-Tree Models for Parsing. *Proceedings of the 33rd Annual Meetings of the Association for Computational Linguistics*, pages 276–283.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

S. Matsubara, S. Asai, K. Toyama, and Y. Inagaki. 1997. Chart-based Parsing and Transfer in Incremental Spoken Language Translation. *Proceedings of the 4th Natural Language Processing Pacific Rim Symposium*, pages 521–524.

S. Matsubara, K. Toyama, and Y. Inagaki. 1999. Sync/Trans: Simultaneous Machine Interpretation between English and Japanese. *N. Foo (Ed.): Advanced Topics in Artificial Intelligence, Lecture Note in Artificial Intelligence 1747*:134–143.

D. Milward. 1995. Incremental Interpretation of Categorial Grammar. *Proceedings of the 7th European Association for Computational Linguistics*, pages 119–126.

M. Nakano, N. Miyazaki, and J. Hirasawa. 1999. Understanding Unsegmented User Utterances in Real-Time Spoken Dialogue Systems. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 200–207.

A. Ratnaparkhi. 1997. A Linear Observed Time Statistical Parser Based on Maximum Entropy Models. *In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing.*

J. Steedman. 1987. Combinatory Grammar and Human Sentence Processing. *Modurality in Knowledge Representation and Natural language Understanding*, pages 187–205. MIT Press.

A. Stolcke. 1995. An Efficient Probabilistic Contex-Free Parsing Algorithm that Computes Prefix Probabilities. *Computational Linguistics*, 21(2):165–201.

H. Watanabe. 1999. A Pruning Method of CFG-Parsing Using Dependency Information. *The 59th National Convention of Information Processing Society of Japan*, 2:345–346. (in Japanese)