Spoken Language Parsing with Robustness and Incrementality

Yoshihide Kato , Shigeki Matsubara , Katsuhiko Toyama and Yasuyoshi Inagaki†
Graduate School of Engineering, Nagoya University †
Faculty of Language and Culture, Nagoya University ‡
Center for Integrated Acoustic Information Research, Nagoya University §
Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan
yosihide@inagaki.nuie.nagoya-u.ac.jp

Abstract

This paper describes a method of incremental parsing for grammatically ill-formed sentences. The method consists of incremental chart parsing and error correcting, which are executed synchronously. The error correction at an early stage enables the parser to construct a syntactic structure of an initial fragment of a grammatically ill-formed sentence whenever a word is inputed. The parser has two features: One is to utilize the syntactic constraints such as reachability and connectability, and the other is to assign a cost, which represents degrees of error corrections, to every syntactic structure. Therefore, the parser, at any time, adopts the syntactic structure of the well-formed sentence which is most similar to the input sentence as a parsing result.

1 Introduction

It is important for a speech dialogue system to interact with a user naturally and smoothly. Such system is required to possess the capability of immediate response to the user, so that the user can recognize the degree of the system's understanding at any time. To satisfy this requirement, the spontaneous speech dialogue system should be able to interpret spoken language at least incrementally (Milward and Cooper, 1994; Inagaki and Matsubara, 1995). Incremental interpretation means to process an input sentence from left to right and extract semantic information from the initial fragment of it.

Another problem on the spontaneous speech dialogue system is whether it can deal with grammatical ill-formedness since grammatically ill-formed expressions appear frequently in spoken language. Most of the systems which have been developed so far, however, assume that input sentences are grammatically well-formed. To solve the problem, some techniques for processing grammatically ill-formed sentences have been proposed. For example, unification-based framework (Douglas and Dale, 1992; Imaichi and Matsumoto, 1995) deals with the ill-formedness such as constraint violations. Partial parsing (Jensen et al., 1983; McDonald, 1992) extracts the information from the partial structures constructed in the parsing process. Extended parsing algorithms (Saito and Tomita, 1988; Mellish, 1989; Kato, 1994; Lee et al., 1995) deal with the syntactic ill-formedness such as deletion, insertion and mutation of words. Little attention, however, has been given to the viewpoint of incrementality in these literatures.

This paper describes a method of incremental parsing for grammatically ill-formed sentences. The method consists of incremental chart parsing (Matsubara et al., 1997) and error correcting, which are executed synchronously. The error correcting, at any time, processes syntactic errors such as deletion, insertion and mutation of words. The error correction at an early stage enables the parser to construct a syntactic structure of an initial fragment of a grammatically ill-formed sentence whenever a word is inputed. The parser has two features: the first one is to utilize the syntactic constraints such as reachability and connectability so that it can correct the errors efficiently. The second one is to assign a cost, which represents degrees of error corrections, to every syntactic structure. The parser adopts the syntactic structure of the well-formed sentence which is most similar to the input sentence as a parsing result. In addition, in order to increase efficiency of parsing, the parser deletes the structures which do not contribute to the construction of the complete structure with a minimum cost.

This paper is organized as follows: The next section is a summary of incremental chart parsing. In section 3, we describe the error correcting method. Reports of an efficiency evaluation of our parsing system are given in section 4. In section 5, we make a comparison with related works.

2 Incremental Chart Parsing

In this section, we summarize incremental chart parsing (Matsubara et al., 1997).

The incremental chart parsing records parsing results in a graph called a *chart*. The edge is labeled a syntactic structure called a *term* which is denoted by an expression $[\alpha]_x$, where x is a category and α is a list of terms, a word or a special symbol '?'. The term $[?]_x$ is called an *undecided term*. The leftmost

input: $w_1 w_2 \cdots w_n$
output: *chart*
$chart := \{(0, 0, [?]_s)\}$;
for $i = 1$ **to** n **begin**
  1)consulting dictionary:
    if category of w_i is x **then** add $(i-1, i, [w_i]_x)$ to *chart*
  2)applying grammar rule:
    for each $(i-1, i, [\cdots]_{x_1}) \in chart$ and
      each grammar rule $a \rightarrow x_2 y \cdots z$
      if $x_1 = x_2$
        then add $(i-1, i, [[\cdots]_{x_1}[?]_y \cdots [?]_z]_a)$ to *chart*
  3)replacing term:
    for each $(0, i-1, \sigma), (i-1, i, [\cdots]_x) \in chart$
      if the category of $lut(\sigma)$ is x
        then add $(0, i, rep(\sigma, [\cdots]_x))$ to *chart*
end

Figure 1: Algorithm of incremental chart parsing

s → np vp	vp → vt s	pron → I	vt → think
np → pron	vp → vi pp	det → the	vi → think
np → det n	vp → be adj	n → train	be → is
np → gi pp	pp → p np	gi → going	adj → best
np → n		p → by	

Figure 2: Example of Grammar

occurrence of an undecided term in a term σ is called a *leftmost undecided term* in σ, which is denoted by $lut(\sigma)$. If an edge is labeled the term which includes an undecided term, it is said to be *active*, otherwise *inactive*.

The incremental parsing algorithm is shown in Figure 1. Here, the edge with label σ between the nodes j and k is denoted by (j, k, σ). The result of replacing the leftmost undecided term in σ with τ is denoted by $rep(\sigma, \tau)$. The incremental chart parsing introduces two new operations to the standard bottom-up chart parsing (Kay, 1980). 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 parser to construct a syntactic structure of the initial fragment of an input sentence.

As an example, let us consider the parse of the following sentence by using the grammar shown in Figure 2:

I think going by train is best. (2.1)

When "think" is inputed after the input of "I", the standard bottom-up chart parser constructs terms $[[[I]_{pron}]_{np}[?]_{vp}]_s$, $[[think]_{vi}[?]_{pp}]_{vp}$ and $[[think]_{vt}[?]_s]_{vp}$. On the other hand, the incremental chart parser also constructs terms:

$$[[[I]_{pron}]_{np}[[think]_{vt}[?]_s]_{vp}]_s \quad (2.2)$$

$$[[[I]_{pron}]_{np}[[think]_{vi}[?]_{pp}]_{vp}]_s \quad (2.3)$$

These terms, at least, represent the relation that "I" is the subject of the verb "think" which can be acquired from the initial fragment " I think". This shows that the incremental chart parsing has capability of constructing a syntactic structure of an initial fragment at an early stage.

3 Incremental Parsing of Ill-formed Sentences

This study deals with three kinds of syntactic errors such as deletion, insertion and mutation of words. In this section, we extend the incremental chart parsing to process grammatically ill-formed sentences which include errors with keeping high degree of incrementality.

3.1 Structure Construction for Ill-formed Sentences

Our purpose is to develop a method of constructing a syntactic structure of the initial fragment of the sentence including errors at any time. This is accomplished easily by the following operations: inserting a word (for a deletion error), skipping a word (for an insertion error) and replacing a word with another one (for a mutation error).

Let us consider the following sentence:

I think by train is best. (3.1)

It might be suspected that (3.1) includes one of the following errors:

1. deletion of "going" in the sentence "I think going by train is best."
2. insertion of "by" in the sentence "I think train is best."
3. mutation of "the" into "by" in the sentence "I think the train is best."

In the first case, the error can be corrected by inserting a gerund "going". Actually, if the parser adds the edge labeled the term $[going]_{gi}$ to the chart and applies grammar rules to it, the parser constructs a term (3.2).

$$[[[going]_{gi}[?]_{pp}]_{np}[?]_{vp}]_s \quad (3.2)$$

The parser replaces the leftmost undecided term $[?]_s$ in (2.2) with (3.2) so that constructs a term

$$[[[I]_{pron}]_{np}[[think]_{vt}[[[going]_{gi}[?]_{pp}]_{np}[?]_{vp}]_s]_{vp}]_s$$

for the initial fragment "I think".

In the second case, the error can be corrected by skipping "by" so that the parser constructs a term

$$[[[I]_{pron}]_{np}[[think]_{vt}[[[train]_n]_{np}[?]_{vp}]_s]_{vp}]_s$$

for the initial fragment "I think by train".

Finally, in the third case, the error can be corrected by replacing "by" with a determiner "the". Then, the parser constructs a term:

$$[[[I]_{pron}]_{np}[[think]_{vt}[[[the]_{det}[?]_n]_{np}[?]_{vp}]_s]_{vp}]_s$$

The above examples show that the incremental chart parser can construct syntactic structures of the initial fragments of ill-formed sentences by the above three operations at an early stage.

3.2 Error Detection

The previous section has illustrated the structure construction for a grammatically ill-formed sentence. The operations, however, do not always enable the parser to construct a syntactic structure. For example, if "train" is skipped in (3.1), no syntactic structure can be constructed.

From the viewpoint of efficiency, the parser should not execute such operations. Thus, error detecting technique is required. What is important is that error detecting should be executed as soon as possible. For this purpose, the parser detects errors by utilizing the information included in the parsing result and a lookahead word. Since such information can be extracted easily from the initial fragment which has been already inputed, high degree of incrementality can be achieved.

3.2.1 Reachability and Connectablity

To formalize the information which is used for the error detection, we define two relations between syntactic categories: *reachability* and *connectability*. Let α and β be any sequences of categories and let a, x, y and z be any categories.

Definition(Reachability) x is *reachable* to y if x and y satisfy one of the following conditions:

1. $x = y$.
2. A grammar rule $y \rightarrow x\alpha$ exists.
3. For some z, x is reachable to z and z is reachable to y. □

The relation that x is reachable to y means that a structure whose root is y has the leftmost descendant x. We write $x \rightsquigarrow y$ if x is reachable to y.

Definition(connectability) x is *connectable* to y if x and y satisfy the following condition 1. or 2.:

1. A grammar rule $a \rightarrow \alpha xz\beta$ exists and $y \rightsquigarrow z$.
2. A grammar rule $a \rightarrow \alpha x$ exists and a is connectable to y. □

The relation that x is connectable to y means that a category sequence xy is allowed by the grammar. We write $x^\frown y$ if x is connectable to y.

3.2.2 Error Correction Procedures

In our method, the parser corrects errors whenever a word is inputed. The procedures for error corrections are shown in Figure 3. Here, we use a special symbol '$*$' for the result of insertion or replacement of a word by error correction. The reachability relation is utilized for the error detecting using the information included in the parsing results. The connectability relation is utilized for the error detecting using a lookahead word.

deletion correction:
for each $(0, i-1, \sigma), (i-1, i, \tau) \in chart$ and
  each category x
  if $x \rightsquigarrow \sigma$ and $x^\frown \tau$ **then** add $(i-1, i-1, [*]_x)$ to $chart$
insertion correction:
for each $(0, i-2, \sigma) \in chart$ and **each** category x of w_i
  if $x \rightsquigarrow \sigma$ **then** add $(i-2, i-1, \varepsilon)$ to $chart$
mutation correction:
for each $(0, i-2, \sigma), (i-1, i, \tau) \in chart$ and
  each category x
  if $x \rightsquigarrow \sigma$ and $x^\frown \tau$ **then** add $(i-2, i-1, [*]_x)$ to $chart$

Figure 3: Procedures for error correction

For example, as Figure 4 (a), let us consider deletion correction for the sentence (3.1) when "by"is inputed. The categories of the leftmost undecided terms in the terms (2.2) and (2.3) are s and pp, respectively. The categories which are reachable to s or pp are gi, n, p and $pron$. Moreover, the only gi in these categories is connectable to the category p of "by". Therefore, the parser inserts a word whose category is gi. The parser corrects the other errors in the same way as Figure 4 (b) and (c) show.

3.3 Cost of Error Correction

In order to keep high degree of incrementality, the parser executes the error correcting at any time. Thus, it may execute the error correcting where errors do not occur actually and construct syntactic structures of the sentence which is not similar to the input sentence. Such structures should be eliminated from the parsing results. For this purpose, the cost which represents degrees of error corrections is assigned to every syntactic structure and the parser adopts the structures with a minimum cost as a parsing result. The calculation of the cost value is as follows:

1. Assigning a cost to every error correction.
2. If the parser constructs a structure by error correction, add the cost of the error correction to the cost of the structure.

For example, when "I" is inputed, a term

$$[[[I]_{pron}]_{np}[?]_{vp}]_s \tag{3.3}$$

is constructed in the parsing phase. Next, when "think" is inputed, a term

$$[[[*]_n]_{np}[?]_{vp}]_s \tag{3.4}$$

is constructed in the error correcting phase. The cost value of the term (3.3) is less than that of (3.4) since (3.3) is with no error correction and (3.4) is with mutation error correction. Consequently, the parser adopts (3.3) as a parsing result.

From the viewpoint of efficiency, the error correcting may increase the number of edges in the chart so that the parsing becomes inefficient. To improve efficiency of the parsing, the parser should delete the

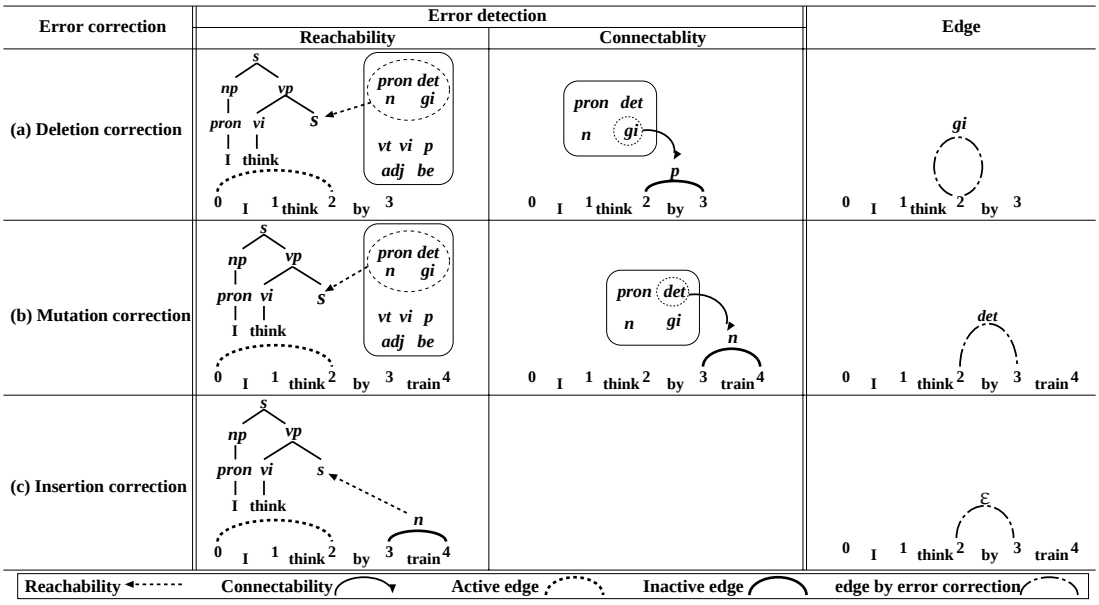

Figure 4: Examples of error corrections

edge deletion:
for each $(0, i, \sigma), (0, i, \tau) \in chart$
  if $cs(\sigma) = cs(\tau)$ and $cost(\sigma) < cost(\tau)$
    then delete $(0, i, \tau)$ from $chart$

Figure 5: Procedure of edge deletion

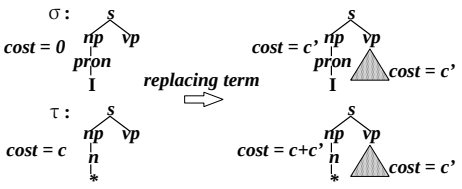

Figure 6: Comparison between the costs of syntactic structures

structures which do not contribute to the construction of the complete structure with a minimum cost. Therefore, the parser executes edge deletion shown in Figure 5. Here, the cost value of a term σ is denoted by $cost(\sigma)$ and a sequence of categories of the undecided terms which occur in a term σ is denoted by $cs(\sigma)$.

Let us consider an example of edge deletion. The term σ in Figure 6 is constructed when "I" is inputed. The term τ in Figure 6 is constructed by mutation correction when "think" is inputed. $cs(\sigma)$ and $cs(\tau)$ are vp. We know $cost(\sigma) < cost(\tau)$ since σ is with no error correction and τ is with mutation correction. Thus, the edge $(0, 1, \tau)$ is deleted.

3.4 An Example of Parsing

This section gives an example of incremental parsing of grammatically ill-formed sentences. Table 1 shows the chart construction process for the sentence (3.1). We assume that the cost value of every error correction is 1.

When "by" is inputed, a gerund is inserted by deletion correction and the edge #6 is constructed. When "train" is inputed, "by" is skipped by insertion correction and the edge #11 is constructed. Next, "by" is replaced with a determiner by mutation correction and the edge #12 is constructed. The edges #3, #10, #15, #18 and #19, which are constructed in error correcting phase, are deleted by edge deletion. The example illustrates that the parser can construct syntactic structures of initial fragments of the grammatically ill-formed sentence (3.1) whenever a word is inputed. The edges which are constructed finally are #23, #24 and #25, and each edge is constructed by deletion, insertion or mutation correction, respectively.

4 Evaluation of Efficiency

We have made an experiment to evaluate efficiency of the method proposed in this paper. The experiment has been made under the following conditions:

- The parser has been implemented in GNU Common Lisp on UltraSPARC-II (Main memory 512MB, CPU 248MHz)
- The grammar consists of 74 rules and 54 categories.
- The cost value of any error correction is 1 regardless of the kind of an error.
- 60 grammatically ill-formed sentences are generated at random in the following way: First, 20 well-formed sentences are generated at random

Table 1: Incremental parsing process of "I think by train is best."

	chart				edge
word	#	*loc*	*term*	cost	deletion
	1	0-0	$[?]_s$	0	
I	2	0-1	$[[[I]_{pron}]_{np}[?]_{vp}]_s$	0	
think	3	0-1	$[[[*]_n]_{np}[?]_{vp}]_s$	1	deleted
	4	0-2	$[[[I]_{pron}]_{np}[[think]_{vt}[?]_s]_{vp}]_s$	0	
	5	0-2	$[[[I]_{pron}]_{np}[[think]_{vi}[?]_{pp}]_{vp}]_s$	0	
by	6	0-2	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[?]_{pp}]_{np}[?]_{vp}]_s]_{vp}]_s$	1	
	7	0-3	$[[[I]_{pron}]_{np}[[think]_{vi}[[by]_p[?]_{np}]_{pp}]_{vp}]_s$	0	
	8	0-3	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[by]_p[?]_{np}]_{pp}]_{np}[?]_{vp}]_s]_{vp}]_s$	1	
train	9	0-3	$[[[I]_{pron}]_{np}[[think]_{vi}[[by]_p[[*]_{det}[?]_n]_{np}]_{pp}]_{vp}]_s$	1	
	10	0-3	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[by]_p[[*]_{det}[?]_n]_{np}]_{pp}]_{np}[?]_{vp}]_s]_{vp}]_s$	2	deleted
	11	0-3	$[[[I]_{pron}]_{np}[[think]_{vt}[?]_s]_{vp}]_s$	1	
	12	0-3	$[[[I]_{pron}]_{np}[[think]_{vt}[[*]_{det}[?]_n]_{np}[?]_{vp}]_s]_{vp}]_s$	1	
	13	0-4	$[[[I]_{pron}]_{np}[[think]_{vi}[[by]_p[[train]_n]_{np}]_{pp}]_{vp}]_s$	0	
	14	0-4	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[by]_p[[train]_n]_{np}]_{pp}]_{np}[?]_{vp}]_s]_{vp}]_s$	1	
	15	0-4	$[[[I]_{pron}]_{np}[[think]_{vi}[[by]_p[[*]_{det}[train]_n]_{np}]_{pp}]_{vp}]_s$	1	deleted
	16	0-4	$[[[I]_{pron}]_{np}[[think]_{vt}[[[train]_n]_{np}[?]_{vp}]_s]_{vp}]_s$	1	
	17	0-4	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{det}[train]_n]_{np}[?]_{vp}]_s]]_{vp}]_s$	1	
is	18	0-4	$[[[I]_{pron}]_{np}[[think]_{vi}[[by]_p[[[*]_{pron}]_{np}]_{pp}]_{vp}]_s$	1	deleted
	19	0-4	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[by]_p[[*]_{pron}]_{np}]_{pp}]_{np}[?]_{vp}]_s]]_{vp}]_s$	2	deleted
	20	0-5	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[by]_p[[train]_n]_{np}]_{pp}]_{np}[[is]_{be}[?]_{adj}]_{vp}]_s]]_{vp}]_s$	1	
	21	0-5	$[[[I]_{pron}]_{np}[[think]_{vt}[[[train]_n]_{np}[[is]_{be}[?]_{adj}]_{vp}]_s]]_{vp}]_s$	1	
	22	0-5	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{det}[train]_n]_{np}[[is]_{be}[?]_{adj}]_{vp}]_s]_{vp}]_s$	1	
best	23	0-6	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[by]_p[[train]_n]_{np}]_{pp}]_{np}[[is]_{be}[best]_{adj}]_{vp}]_s]_{vp}]_s$	1	
	24	0-6	$[[[I]_{pron}]_{np}[[think]_{vt}[[[train]_n]_{np}[[is]_{be}[best]_{adj}]_{vp}]_s]_{vp}]_s$	1	
	25	0-6	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{det}[train]_n]_{np}[[is]_{be}[best]_{adj}]_{vp}]_s]_{vp}]_s$	1	

from the grammar (the length of the sentence is 4, 6, 8 or 10 and the number of the sentences is 5 for each length). Second, random occurrences of specific types of errors are introduced into these sentences.

We measured the worst processing time per word in the following cases:

- the parsing with both error detection and edge deletion
- the parsing without error detection and with edge deletion
- the parsing with error detection and without edge deletion

Table 2 shows averages of the worst processing time per word. The time in the parsing without edge deletion was a great deal longer than that with edge deletion. The time in the parsing without error detection was longer than that with error detection. This result demonstrates that our proposed method is useful for increasing efficiency of the parsing.

5 Related Works

First, we compares our work with related works from the viewpoint of incrementality, i.e., the timing of error correcting and the construction of syntactic structures on the way.

Mellish (Mellish, 1989) and Kato (Kato, 1994) have proposed a method based on chart parsing. The method is divided into the bottom-up phase and the top-down phase. The bottom-up phase constructs structures of an input sentence in the bottom-up chart parsing manner. If it failed to construct a complete structure, the top-down phase is started to detect errors. The bottom-up chart parsing cannot construct a structure of the initial fragment of a sentence and, consequently, Mellish's and Kato's parser also cannot construct it. In addition, it is not until the bottom-up phase is finished that the error detection process is invoked. Moreover, error correcting needs all of the partial syntactic structures constructed in the bottom-up chart parsing. Thereby, the method cannot execute parsing and error correcting synchronously. Saito's method is on basis of Generalized LR parsing (Saito and Tomita, 1988). The method executes parsing and error correcting synchronously as ours. Saito's parser, however, cannot construct a structure of the initial fragment of a sentence, because the method is based on Generalized LR parser. Lee extended Earley Algorithm to deal with grammatically ill-formed sentences (Lee et al., 1995). Their parser performs parsing and error correcting synchronously. The algorithm cannot construct a syntactic structure while parsing is performed.

On the other hand, our proposed parser can process grammatically ill-formed sentences incrementally since it executes parsing and error correcting synchronously. In addition, the parser can construct

Table 2: Worst processing time per word

Kind of Error	Length of Sentence	with both error detection and edge deletion	without error detection and with edge deletion	with error detection and without edge deletion
insertion of a word	5	0.028 sec	0.048 sec	0.044 sec
	7	0.158 sec	0.372 sec	0.548 sec
	9	0.390 sec	0.600 sec	2.964 sec
	11	1.014 sec	1.282 sec	61.948 sec
mutation of a word	4	0.064 sec	0.038 sec	0.060 sec
	6	0.026 sec	0.034 sec	0.030 sec
	8	0.600 sec	0.830 sec	4.514 sec
	10	1.390 sec	2.030 sec	56.000 sec
deletion of a word	3	0.018 sec	0.088 sec	0.024 sec
	5	0.070 sec	0.142 sec	0.146 sec
	7	0.334 sec	0.420 sec	1.160 sec
	9	0.888 sec	1.304 sec	6.926 sec

structures of the initial fragments of grammatically ill-formed sentences since it is based on the incremental chart parsing.

Next, our works is compared with Saito's and Lee's from the viewpoint of efficiency of the error detecting.

Saito's parser utilizes LR parsing table which predicts what word is the next input in order to detect errors. In Lee's method, the parser predicts the next constituent by utilizing the parsing results. Saito's and Lee's error detecting are identical to the error detecting which utilizes the reachability relation.

On the other hand, the error detecting which have been proposed in this paper is more efficient since the error detecting utilizes not only the reachability relation but also the connectability relation.

6 Conclusion

This paper has proposed an incremental and robust parsing method for grammatically ill-formed sentences. The method executes parsing and error correcting synchronously. Correcting errors at any time enables the parser to construct syntactic structures of the initial fragments of grammatically ill-formed sentences whenever a word is inputed. Using our method, the number of the structures may increase explosively because the parser constructs them not only in parsing phase but also in error correcting phase. In order to decrease it, a cost has been assigned to every error correction.

In future works, we intend to apply the method proposed in this paper to the incremental spoken language translation system (Matsubara and Inagaki, 1997).

References

S. Douglas and R. Dale. 1992. Towards Robust PATR. In *Proc. of COLING-92*, pages 468–474.

O. Imaichi and Y. Matsumoto. 1995. Integration of Syntactic, Semantic and Contextual Information in Processing Grammatically Ill-Formed Inputs. In *Proc. of IJCAI-95*, pages 1435–1440.

Y. Inagaki and S. Matsubara. 1995. Models for Incremental Interpretation of Natural Language. In *Proc. of SNLP-95*, pages 51–60.

K. Jensen, G. E. Heidorn, L. A. Miller, and Y. Ravin. 1983. Parse Fitting and Prose Fixing: Getting a Hold on Ill-Formedness. *Computational Linguistics*, 9(3–4):147–160.

T. Kato. 1994. Yet Another Chart-Based Technique for Parsing Ill-Formed Input. In *Proc. of ANLP-94*, pages 107–112.

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

K.J. Lee, J.K. Kweon, J. Seo, and G.C. Kim. 1995. A Robust Parser Based on Syntactic Information. In *Proc. of EACL-95*, pages 223–228.

S. Matsubara and Y. Inagaki. 1997. Incremental Transfer in English-Japanease Machine Translation. *IEICE Trans. on Inf. & Sys*, E80-D(11):1122–1129.

S. Matsubara, S. Asai, K. Toyama, and Y. Inagaki. 1997. Chart-based Parsing and Transfer in Incremental Spoken Language Translation. In *Proc. of NLPRS-97*, pages 521–524.

D.D. McDonald. 1992. An Efficient Chart-based Algorithm for Partial-Parsing of Unrestricted Texts. In *Proc. of ANLP-92*, pages 193–200.

C.S. Mellish. 1989. Some Chart-Based Techniques for Parsing Ill-Formed Input. In *Proc. of ACL-89*, pages 102–109.

D. Milward and R. Cooper. 1994. Incremental Interpretation : Applications, Theory, and Relationship to Dynamic Semantics. In *Proc. of COLING-94*, pages 748–754.

H. Saito and M. Tomita. 1988. Parsing Noizy Sentences. In *Proc. of COLING-88*, pages 561–566.