EFFICIENT INCREMENTAL DEPENDENCY PARSING

Yoshihide Kato[†]
Shigeki Matsubara[‡]
Katsuhiko Toyama[†§]
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,matu,toyama,inagaki}@inagaki.nuie.nagoya-u.ac.jp

Abstract

This paper describes an efficient method of incremental dependency parsing based on phrase structure grammar with the dependency relation. The reachability relation between syntactic categories is utilized for connecting a head word with a dependent word simultaneously with the inputs. The method does not need to construct the whole parse tree of an initial fragment on the word-by-word basis, and thus can be expected to be usable for simultaneous spoken language processing. An experiment on the ATIS corpus has shown the technique of utilizing the reachability to be effective for reducing processing time of the incremental dependency parsing.

1 Introduction

Dependency grammar has a long tradition in natural language theories. In recent years, great attention has been paid to the concept of dependency, as the importance of relations between words is recognized in the field of language parsing. Dependency parsing techniques of written language has been studied so far, on the other hand, few investigations has been done for that of spoken language. In particular, such spoken language processing systems as real-time dialogue system, simultaneous machine interpreting system, etc. are being studied, and thus dependency parsing needs to be discussed from the viewpoint of incrementality.

An approach based on context-free phrase structure grammars is known as one of the dependency parsing techniques. In this approach, dependency grammar is described as a set of phrase structure rules with head-dependent relations between syntactic categories, and the dependencies are extracted from parse trees according to Collins' structures mapping procedure [1]. This means that we can not acquire the dependency structures before getting parse tree completely. If such the technique is used for incremental spoken language processing, it is strongly required not to damage the incrementality.

This paper describes an efficient method of incremental dependency parsing based on CFG with the dependency relation. In order to connect a head word with a dependent word simultaneously with the inputs, the reachability relation between syntactic categories is utilized. Our method, which is different from the standard one of incremental parsing such as incremental chart parsing [4], does not necessarily construct the whole parse tree for an initial fragment, and can be therefore expected to be suitable for incremental spoken language processing. To evaluate the performance of our method, an experiment on the ATIS corpus has been made. The experimental result has shows the technique of utilizing the reachability to be effective for reducing parsing time.

$$\begin{array}{ll} S & \to NP\ VP^* \\ VP \to VP1^*\ ADV\ /\ VP1^*\ PP \end{array} \qquad \begin{array}{l} NP\ \to DET\ N^*\ /\ DET\ ADJ\ N^* \\ VP1 \to V^*\ NP\ /\ V^*\ NP\ NP \end{array}$$

Figure 1: Grammar

---

procedure *dependency*
input: $[[\cdots]_{C_1}\cdots[\cdots]_{C_i}\cdots[\cdots]_{C_m}]_X$
output: D
$D := \cup_{C_j(1\leq j\leq m)} dependency([\cdots]_{C_j})$
if a rule $X \to C_1\cdots C_i^* \cdots C_m$ exists **then** add $\langle h([\cdots]_{C_j}), h([\cdots]_{C_i})\rangle$ to D $(1\leq j\leq m, j\neq i)$

Figure 2: Dependency extraction procedure

2   Incremental Extraction of Word Dependencies

An orthodox and typical method of incremental dependency parsing based on context free grammar with the dependency relation could be as follows: When an input word is produced, (i) the partial parse trees of the initial fragment are constructed by incremental CFG parsing such as incremental chart parsing [4], (ii) the dependencies between words are extracted from the parse trees according to the structure mapping procedure [1]. This section describes such the method shortly.

Dependency is a kind of relations between words, and represents that a word modifies another word. The modified word is called a *head* and the modifying word is called a *dependent*. We write the dependency, in which w_d is dependent and w_h is head, as $\langle w_d, w_h\rangle$. Dependencies can be extracted from a parse tree by identifying the head-child in each rule [1]. Just one category in the right-hand side of each rule is called *head-child*, which is annotated with a special symbol * as shown in Fig. 1. A parse tree has a *head-word*, which represents a main meaning of the parse tree. We denote the head-word of a term σ by $h(\sigma)$. Head-words propagate up through the tree, each parent receiving its head-word from its head-child. The dependency extraction procedure is defined in Fig. 2.

Incremental chart parsing produces the partial parse trees for initial fragments [4]. The partial parse tree is called a *term*, which is denoted by $[\alpha]_X$ where X is a category and α is a word, a special symbol ? or a list of terms. We denote the category of $[\alpha]_X$ by $cat([\alpha]_X)$. The term $[?]_X$ is called *undecided term* and corresponds to the part not inputed yet. The leftmost occurrence of an undecided term in a term σ is called *leftmost undecided term* in σ, which denoted by $lut(\sigma)$. The incremental chart parsing introduces two new operations to the standard bottom-up chart parsing [2]. One is the application of rules to the term including undecided terms and the other is the replacement of the leftmost undecided term with the term including undecided terms. As an example, let us consider the parse of the fragment "The boy saw". The standard chart parsing produces terms: $(1)[[[the]_{DET}[boy]_N]_{NP}[?]_{VP}]_S$, $(2)[saw]_V$, $(3)[[saw]_V[?]_{NP}]_{VP1}$ and so on. However, it produces no term for "The boy saw". On the other hand, the incremental chart parsing applies the rule $VP \to VP1\ ADV$ for (3) to produce a term $(4)[[[saw]_V[?]_{NP}]_{VP1}[?]_{ADV}]_{VP}$ and the leftmost undecided term of (1) is replaced with (4) so that it produces the term $(5)[[[the]_{DET}[boy]_N]_{NP}[[[saw]_V[?]_{NP}]_{VP1}[?]_{ADV}]_{VP}]_S$ for "The boy saw". Dependencies for initial fragments can be extracted from the terms produced in incremental chart parsing by applying the dependency extraction procedure. For example, the dependencies for "The boy saw" is computed from the term (5) as shown in Fig. 3.

3   Incremental Dependency Parsing

The method described in Sect.2 can compute the dependencies for initial fragments. However, we can not say the efficiency is enough, because it may produce a lot of terms which have the same dependencies. If it is possible to extract the dependencies correctly, all the terms are not necessary to
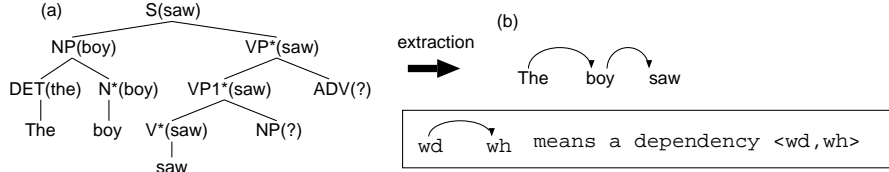
Figure 3: An example of computing dependencies for initial fragments

```
procedure dependency2
input: σ₁ ··· σₘ
output: D
D := dependency(σₘ)
for i := 0 to m − 2
    if cat(σ_{m−i}) ⇝^{h *} cat(lut(σ_{m−i−1})) then h(lut(σ_{m−i−1})) := h(σ_{m−i})
    D := D ∪ dependency(σ_{m−i−1})
```

Figure 4: Algorithm of extracting dependencies from connected terms

produce so that the efficiency is increased. We accomplish this by utilizing the reachability relation.

Before describing our proposed method, we give a definition of reachability. Let X and Y be categories. We write $X \rightsquigarrow Y$, if a rule $Y \rightarrow X\alpha$ exists for some categories sequence α. Let $\rightsquigarrow^*$ be the reflexive transitive closure of $\rightsquigarrow$. We say that Y is reachable from X, if $X \rightsquigarrow^* Y$. The relation that Y is reachable from X means that a parse tree whose root is Y may have the leftmost descendant X.

To compute dependencies, we classify the reachability relation.

Definition 1 *Let X and Y be categories. We write $X \overset{h}{\rightsquigarrow} Y$ if a rule $Y \rightarrow X^*\alpha$ exists for some categories sequence α. We write $X \overset{d}{\rightsquigarrow} Y$ if a rule $Y \rightarrow X\alpha Z^*\beta$ exists for some categories sequences α and β and some category Z.* □

When Y is reachable from X, X and Y satisfy $X \overset{h}{\rightsquigarrow}^* Y$ or $X \rightsquigarrow^* \overset{d}{\rightsquigarrow} \rightsquigarrow^* Y$. $X \overset{h}{\rightsquigarrow}^* Y$ means that the head-word of leftmost descendant X propagates up to the ancestor Y. In contrast, $X \rightsquigarrow^* \overset{d}{\rightsquigarrow} \rightsquigarrow^* Y$ means that the head-word of X does not propagate up to Y.

For initial fragments, the proposed parsing (i) produces terms according to the standard chart parsing, (ii) connects terms by utilizing reachability and (iii) extracts dependencies from the connected terms. The same dependencies as the one described in Sect. 2 can be obtained by the method.

The method connects a term σ with another term τ, if $cat(lut(\sigma))$ is reachable from $cat(\tau)$. We represent connected terms by a list of terms. The adjacent terms σ and τ in the list mean that they are connected by reachability. Fig. 4 illustrates the procedure of extracting dependencies from connected terms. By utilizing the relation $\overset{h}{\rightsquigarrow}^*$, the procedure propagates a head-word of a term to another term. If the relation $\rightsquigarrow^* \overset{d}{\rightsquigarrow} \rightsquigarrow^*$ is satisfied, the head-word is not propagated. This enables the parsing to compute the dependencies for fragments without applying rules to terms including undecided terms.

As an example, let us consider the incremental parsing process of the fragment "The boy saw". When "saw" is processed, the parsing produces the terms (1) and (2) according to the standard chart parsing fashion. The category of (2) is V and the one of the leftmost undecided term of (1) is VP. Since $V \rightsquigarrow^* VP$, (1) and (2) are connected. Moreover, since $V \overset{h}{\rightsquigarrow}^* VP$ and the head-word of (2) is "saw", the head-word of the leftmost undecided term of (1) becomes "saw" (see Fig. 5.). For "The boy saw", the dependencies $\langle the, boy \rangle$ and $\langle boy, saw \rangle$ are obtained by the procedure in Fig. 4. The dependencies are the same as the ones extracted from the term (5).

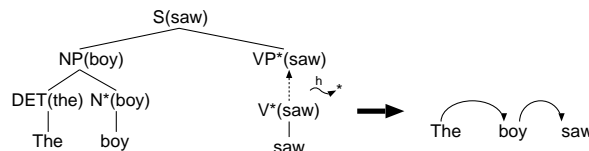Figure 5: An example of propagation of head-word
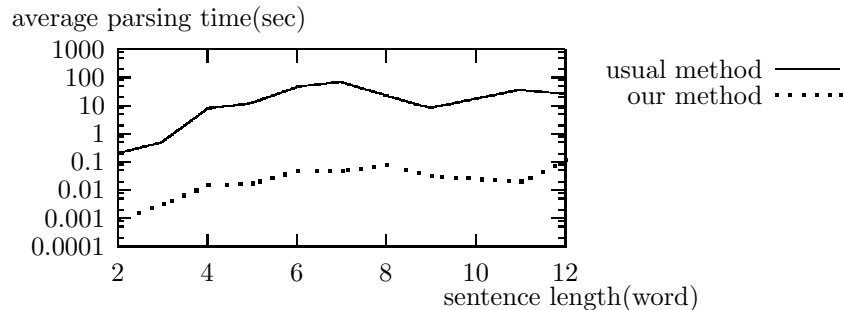
average parsing time(sec)

Figure 6: Parsing time per sentence

4 Experimental Result

In order to evaluate the performance of our technique, an parsing experiment has been made. The parsing was implemented in GNU Common Lisp on a Linux PC with Pentium III 1GHz processor and 1GB main memory. We used the ATIS corpus of Penn Treebank [3]. The corpus contains 578 sentences with CFG parse tree tags, and 509 phrase structure rules in them are used for the test.

The parsing time has been compared between the technique explained in Sect. 2 (i.e. usual method) and the one utilizing the reachability relation in Sect. 3 (i.e. our method). Fig. 6 shows the relation between the sentence length and the processing time[1]. The average time of our method is 0.03 sec., while that of the usual method is 23.49 sec. This result shows the technique of utilizing the reachability to be effective for reducing processing time.

5 Concluding Remarks

This paper has described an efficient method of incrementally computing the dependencies between words. We have pointed out that the reachability relation between syntactic categories is related to the head-dependent relation between words. We provided a simple algorithm for acquiring the lexical dependencies based on the reachability relation. An experimental result has demonstrated that our method is effective for reducing processing time. Therefore, the method can be expected to be applicable to a simultaneous speech interpreting system which we are studying.

In this method, the lexical dependencies of the natural language sentence have been extracted from the CFG parse tree. From the point of view of only the efficiency, however, the parsing with only syntactic constraints on dependency grammar might also be available. For the comparison with such the technique, further experiments will be made in the future.

References

[1] Collins, M. J.: A New Statistical Parser Based on Bigram Lexical Dependencies, *Proc. of the 34th Annual Meeting of the Assoc. for Computational Linguistics*, pp.184-191 (1996).

[2] Kay, M.: Algorithm Schemata and Data Structures in Syntactic Processing, *Technical Report CSL-80-12*, Xerox PARC (1980).

[3] Marcus, M. P., Santorini, B. and Marcinkiewicz, M. A. : Building a Large Annotated Corpus of English: the Penn Treebank, *Computational Linguistics*, Vol.19, No.2, pp.310-330 (1993).

[4] Matsubara, S., Asai, S., Toyama, K. and Inagaki, Y.: Chart-based Parsing and Transfer in Incremental Spoken Language Translation, *Proc. of NLPRS-97*, pp.521-524 (1997).

---

[1]We terminated the parsing when the processing time per word was over 60 sec. The result is for 154 sentences for which the usual method did not terminate the parsing.