SOLUTIONS FOR PROBLEMS OF MT PARSER
- METHODS USED IN MU-MACHINE TRANSLATION PROJECT -

Jun-ichi NAKAMURA, Jun-ichi TSUJII and Makoto NAGAO
Department of Electrical Engineering
Kyoto University
Sakyo, Kyoto 606, JAPAN

1. Introduction

A parser is a key component of a machine translation (MT) system. If it fails in parsing an input sentence, the MT system cannot output a complete translation. A parser of a practical MT system must solve many problems caused by the varieties of characteristics of natural languages. Some problems are caused by the incompleteness of grammatical rules and dictionary information, and some by the ambiguity of natural languages. Others are caused by various types of sentence constructions, such as itemization, insertion by parentheses and other typographical conventions that cannot be naturally captured by ordinary linguistic rules.

The authors of this paper have been developing MT systems between Japanese and English (in both directions) under the Mu-machine translation project [NAGAO 85]. In the system's development, several methods have been implemented with grammar writing language GRADE [NAKAMURA 84] to solve the problems of the MT parser . In this paper, first the characteristics of GRADE and the Mu-MT parser are briefly described. Then, methods to solve the MT parsing problems that are caused by the varieties of sentence constructions and the ambiguities of natural languages are discussed from the viewpoint of efficiency and maintainability.

2. Characteristics of GRADE and Mu-MT parser

Mu-project has been developing two MT systems based on the transfer approach: Japanese-to-English MT system [TSUJII 84] [NAGAO 84, 85, 86], and English-to-Japanese MT system [TSUJII 85]. All grammars in the project, including the Japanese and English analysis grammars are written in the same grammar writing language, GRADE [NAKAMURA 84], which has the following features:

1) Each grammatical rule is expressed by a Tree-to-Tree transformation rewriting rule based on a flexible pattern matching algorithm.
2) A 'subgrammar' method allows for the division of the whole grammar into several parts, and for the detailed control of the translation process. A subgrammar is a set of rewriting rules and the whole grammar is expressed as a network of such subgrammars (subgrammar-network).
3) The parallel execution of a subgrammar's rewriting rules makes it possible to collect all possible interpretations of a specified part of an input sentence, and to compare them in order to select the one most feasible.

Making use of these features, the English analysis grammar of Mu-Project is roughly divided into the following three subgrammars (SG) [TSUJII 85]:

1) pre-analysis SG, which analyzes constructions such as itemized forms and insertion by parentheses which cannot be treated by ordinary grammatical rules, and divides the input sentence into several fragments.
2) main-analysis SG, which performs syntactic and semantic analysis in the usual sense.
3) post-analysis SG, which combines the fragments of sentences divided by the pre-analysis SG.

Solutions to the efficiency and maintainability problems of an MT parser are described in the following sections with examples from the English analysis grammar.

3. Problems caused by Constructions of Sentences

Constructions of sentences, such as, itemized forms cause serious problem of MT parsers. There are many such exceptional constructions in written texts, especially in the abstracts of scientific and technological papers which the Mu-project aims to translate. (The current corpus of the project consists of the abstracts extracted from the INSPEC database without any pre-editing.) The following is a typical example:

> Four major factors affect the cost of ownership: 1) purchase price, 2) investment tax credits, 3) cost of maintenance and repairs and 4) efficiency costs. --- (1)

This type of construction can be handled by the rules of the context free grammar (CFG) shown in figure 1.

```
S   →  S' ':' IF '.'      IF  →  IFL 'and' IFE
IFL →  IFE                IFL →  IFE ',' IFL
IFE →  IN NP              IN  →  NUMBER ')'
NP  →  DET N          :rules for itemized forms
...
S   →  S' '.'         : rules for
S'  →  NP VP            structural analysis
...
```

S: Sentence, IF: Itemized Form, IFL: List of Itemized Element, IFE: Element of Itemized Form, IN: Item Number, NP: Noun Phrase

Figure 1 Context Free Rules for processing Itemized Forms

However, if rewriting rules to handle such sentence constructions are added to the analysis grammar, the following problems would arise:

a) Deterioration of analysis efficiency: Rewriting rules which need not be referred to in a structural analysis will increase. Since CFG parsers cannot distinguish the rewriting rules for the structural analysis from the rewriting rules for the typographical sentence construction analysis, the increase in the total number of rules reduces the efficiency of the analysis.
b) The loss of useful heuristics for correct analysis: For example, that each item in an itemized form can be analyzed independently is the heuristics useful for the analysis grammar. To utilize such heuristics, the recognition of global sentence structures should precede the detailed structural analysis. It, however, cannot be utilized effectively in an analysis based on CFG.
c) Deterioration of the maintainability of the analysis grammar: It becomes difficult to dis-

tinguish rules concerned with particular text types (i.e., abstracts of scientific papers, articles of newspapers, etc.) from rules concerned with more general linguistic phenomena.

In constrast to such an approach, Tree-to-Tree-type rewriting rules and using subgrammar-networks to control the parsing, which are features of GRADE, allow the MT parser to analyze such sentential forms without the deterioration of efficiency and maintainability. The analysis procedure of an itemized form, for example, is the following:

1) First, the fragments of an input sentence are separated from each other by a tree structure pattern such as that shown in figure 2. In this example, one fragment is the core part of the sentence and the others are itemized parts.
2) Each fragment is analyzed independently by the main-analysis subgrammar.
3) Finally, fragmental results are integrated into the whole analysis result by the post-analysis subgrammar.

```
--- ':' { NUM ')' --- ','}+ --- 'and' NUM ')' --- '.'
 #0              IFL1      IFE2              IFE3
```

(a) Pattern to extract fragments from a sentence

```
%( #0 COLON IFL1 IFE2 AND IFE3 PERIOD);
IFL1: LENGTH(1,10,%(NUM1 RIGHT_PAREN1 #1 COMMA));
IFE2: %(NUM2 RIGHT_PAREN2 #2);
IFE3: %(NUM3 RIGHT_PAREN3 #3);
```

(b) GRADE pattern

var	Assigned Word Sequence
#0 :	Four Major Factors affect the cost of ownership:
IFL1 :	1) purchase price, 2) investment tax credits,
IFE2 :	3) cost of maintenance and repairs and
IFE3 :	4) efficiency costs.

(c) Correspondences between the GRADE pattern and fragments of the sample sentence (1)

Figure 2 An example of the pattern to extract adequate fragments from the itemized form

In this way, the grammatical rules for analysing sentence constructions are placed in the pre-analysis subgrammar and are separated from the main-analysis subgrammar where the syntactic and semantic analysis of the input sentences is performed. This separation avoids the degradation of both analysis efficiency and the grammar's maintainability - both serious problems in parsers based on CFG.

4. Ambiguity Problems

The ambiguity of natural language is one of the biggest problems in an MT parser. The MT parser often outputs a wrong analysis result, or exhausts computer time and memory, due to ambiguity. Probably, knowledge-bases combined with context analysis and an inference mechanism might enable disambiguation in the parsing of natural language. Nevertheless, the current knowledge engineering technology has not yet developed large scale and high quality knowledge-bases usable by a practical MT parser. Furthermore, disambiguation based on such knowledge bases is itself still a research problem for which we have not yet had any concrete solutions. Since the MT parser has to analyze fairly long sentences containing various sorts of constructions, such as long conjuncted phrases, appositions, ellipses, etc., it has to use heuristic methods for the disambiguation to determine the priorities of the possible syntactic and semantic interpretations produced for an input sentence.

Since the analysis methods based on CFG usually handle each interpretation independently, the following two methods are typically useful for determining the priorities of such interpretations:

a) First, obtaining all possible interprations from the input sentence and then comparing them. Note that such rules have to compare different interpretations (tree structures) and cannot be CFG rules.
b) Heuristically adjusting the order of application of CFG rewriting rules, for example, to obtain just one interpretation from the input sentence and to ignore others [NAGAO 82].

Unfortunately, the first method lowers parsing efficiency because of the inherent 'combinatorial explosion'. It often exhausts the limited computer time and memory and still gives no interpretaion. The second method has difficulties in maintaining the priority ordering of rewriting rules adequately. This difficulty increases with any increase of rules in the grammar whole.

However, many kinds of ambiguity can be solved by adopting the controlling mechanism provided by GRADE's subgrammar and the 'procedural analysis' method [TSUJII 84]. In this section, we will focus on how to disambiguate the interpretation of functional words like 'after', 'as', and 'for' which can be used both as prepositions and as (sentential) conjunctions. These functional words bring about the two problems as follows:

1) Processing efficiency; there are certain kinds of ambiguities which may be solved automatically, when we use a CFG based parser augmented by simple semantic checking. For example, 'as' and 'after' in the following sentences are used as prepositions and are not used as conjunctions:

> Remarkably, the printed board can be executed as a one-sided or a double-sided unit. ---(2)

> The solderbility of reflowed tin and 40 percent load coated copper has been examined after thermal aging designed to include extensive copper-tin intermetallic compound growth. ---(3)

However, a lot of computer time and memory are necessary, because the number of possible partial interpretations increases combinatorially. For the correct interpretation of (3), we need such a semantic constraint as the agent of 'to design' should be a human. However, such a semantic constraint is more preference than a real constraint.

2) Ambiguous interpretation; for the correct disambiguation, a complete semantic and contextual analysis is necessary. The word 'for' in the following sentence is ambiguous.

> Many opportunities occur for contractors to obtain electrical maintenance work in factories. ---(4)

This sentence has two possible syntactic structures as follows:

> Many opportunities occur $[_S$ for contractors $[_{VP}$ to obtain $[_{NP}$ electrical maintenance work in

factories.]]] ---(4.1)

Many opportunities occur for [$_S$ [$_{NP}$ contractors to obtain electrical maintenance] [$_{VP}$ work in factories.]] ---(4.2)

The dominant reading is (4.1), but we cannot reject (4.2).

Both these problems must be solved by an MT parser. The efficiency problem can be solved by adopting the '3 stage procedural analysis' as follows [YAMAMOTO 86]:

Step 1. Disambiguation by simple but reliable cues.
Rules for disambiguating parts-of-speech are applied. For example, it is tentatively determined that 'as' in (2) is a preposition by simple but rather reliable cues such as:

"If there are no verbs after the ambiguous function word, its part-of-speech is a preposition."

Such a rule is easily expressed by using the flexible pattern matching functions of GRADE.

Step 2. Disambiguation based on intermediate analysis result.
'After' in (3) cannot be disambiguated in step 1. The ambiguity is solved in this step by the following rule:

"If the word sequence after the ambiguous function word is to be analyzed as a sentence, the word is a conjunction. If the phrase is a noun-phrase, then the word is a preposition."

For (3), first, the word sequence 'thermal ... growth' after 'after' is extracted and analysed completely. The grammar for complete analysis, written as a subgrammar network, is called from this rule. For this sentence, the word 'after' is determined as a preposition, because the word sequence following 'after' is analyzed as a noun-phrase (Figure 3).

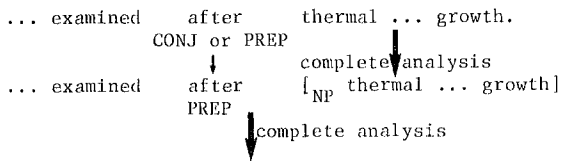

Figure 3. An example of disambiguation by partial analysis

This kind of top-down processing is easily realized by the pattern matching functions and the invocation of subgrammar-networks.

Step 3. Complete Analysis.
After step 2, each word (or phrase) is no longer (syntactically) ambiguous. The complete analysis of the sentence therefore becomes straightforward. However, the determinations of parts-of-speech in step 1 and step 2 are tentative, and it sometimes happens that complete analyses cannot be obtained becuase wrong decisions have been made. In such cases, these tentative decisions may be changed by step 3.

During step 1 and step 2's construction of tentative interpretations, various sorts of heuristic rules are applied; these are ordered according to their relative 'strengths'. The part-of-speech interpretation of 'for' in (4), for example, shows that a heuristic rule based on surface syntactic cues such as

"If there is a word sequence '... for NP to verb ...', the 'for' is a preposition which marks the subject of the infinitive clause."

is useful. This kind of heuristic rules is useful for choosing the most feasible interpretations, even if there are several syntactically and semantically possible interpretations.

The '3 stage procedual analysis', in which bottom-up processes (step 1) and top-down processes (step 2) are carefully combined, can be implemented straightforwardly by utilizing the rich control schemes provided in GRADE.

5. Conclusion

Methods for solving the efficiency and maintainability problems of the MT parser have been discussed. The MT parser of the Mu-project is designed to solve the problems by utilizing the several features of GRADE. Those methods are based on the following ideas:

1) The separation of the sentence construction analysis from the structural analysis.
2) Independent processing by dividing the sentence into several fragments by means of heuristic rules.
3) The carful combination of bottom-up and top-down processing.

There are other problems in the MT parsing, for example, problems of contextual analyses, which remain to be attacked. The work of improving the grammars and the software system must be continued in the future.

Acknowledgement

This study is a part of "Research on the machine translation system (Japanese-English and English-Japanese) of scientific and technological documents" and is being performed through Special Coordination Funds for Promoting Science & Technology of the Science and Technology Agency of the Japanese Government.

Finaly, we would like to thank the members of Mu-project for their useful comments.

REFERENCES

[NAGAO 82] Nagao, M., Nakamura, J., A Parser Which Learns the Application Order of Rewriting Rules, Proc. of COLING82.
[NAGAO 84] Nagao, M., Nishida, T., Tsujii, J., Dealing with Incompleteness of Linguistic Knowledege on Language Translation, Proc. of COLING84 (1984).
[NAGAO 85] Nagao, M., Tsujii, J., Nakamura, J., The Japanese Government Project for Machine Translation, Computational Linguistics, Vol. 11, No. 2-3 (1985).
[NAGAO 86] Nagao, M., Tsujii, J., The Transfer Phase of the Mu Machine Translation System, Proc. of COLING86 (1986).
[NAKAMURA 84] Nakamura, J., Tsujii, J., Nagao, M., Grammar Writing System (GRADE) of Mu-Machine Translation Project and its Characteristics, Proc. of COLING84 (1984).
[TSUJII 84] Tsujii, J., Nakamura, J., Nagao, M., Analysis Grammar of Japanese in Mu-project, Proc. of COLING84 (1984).
[TSUJII 85] Tsujii, J., et al, Configuration of English-to-Japanese Translation System in Mu-project, WGNL50-3, IPSJ (1985) (in Japanese).
[YAMAMOTO 86] Yamamoto, T., Tsujii, J., Nagao, M., Disambiguation Method for Part-of-Speech of English-to-Japanese Translation System in Mu-Project, WGNL53-7, IPSJ (1986) (in Japanese).