1998 International Computer Symposium
Workshop on Artificial Intelligence
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

Guarded Parsing of Chinese Sentence without Word Segmentation

Chi-Hong Leung and Kwok-Ping Chan and Zhi-Hong Zuo and Ge-Yang Liu
Department of Computer Science & Information Systems
The University of Hong Kong
Email: {chleung,kpchan,zhzuo,gyliu}@csis.hku.hk

Abstract

One problem in Chinese language processing is the lack of grammatical regularity in the language. This leads to very complex Chinese grammar in order to obtain satisfactory results, which in term increases the complexity in the parsing process. In order to simplify this process, it is desired that simple grammar is to be used. However, this will cause ambiguities in the parse result.

Word segmentation is an tradition preprocessor to a Chinese parser. Unfortunately, word segmentation is a process which is error-prone, and once an error is made, the parsing cannot be recovered. In this paper, a preprocessor called *lexical scanner* is used to replace the word segmentation process. The lexical scanner is a integrated part of the parser, and is a kind of *segment on the fly* approach. However, this has to be combined with guarded parsing in order to achieve good results.

In this paper, we introduced a technique called *Guarded Parsing* which was first introduced by us for syntactic pattern recognition [1]. In this approach, simple grammar is used so that the effort in building and maintaining the grammar is much reduced. The parsing process will also be simplified. Ambiguity is resolved by the use of guards which not only checks the syntactic compatibility, but also take semantic information into consideration.

This work is a part of the Hong Kong University Machine Translation (HKUMT) project. The project is supported by the Industry Department, Hong Kong Government.

I. Introduction

Although it is often said that Chinese language does not have a well structured grammar, parsing technique is still an indispensable part in natural language processing and machine translation. Also, the resulting parse tree gives invaluable information on each word of the sentence, such as the parts of speech, the thematic roles etc. This information is important for us to understand the sentence and eventually perform the translation. This is especially important, because if high quality translation is to be achieved, language understanding is a necessity.

Also, even when a sentence is grammatically correct, it still may not be a valid sentence. This will cause further problem because in Chinese language, a lot of words can perform more than one role, e.g. both noun and verb. Also, some elements in a sentence can be implicit. For example, the subject or object of the sentence can be skipped, and still be a valid sentence. Furthermore, the subject/object can be another sentence, which does not contain subject or object [2]. This will cause a lot of problem during parsing, and a lot of possible parse will be resulted, if only simple grammar is used [3].

The following two sentences are examples where sentences are used as subjects or objects of the sentence.

我看見小明在吃飯
勤力讀書是考試成功的必要條件

In order to reduce the number of possible parses, both syntactic and semantic information have to be used to prune away the invalid sentences. For example, only sentences/clause that represent events can be used as subjects/objects. This simple rules immediately reduces a lot of possible parse trees.

Natural language is not context free, although mostly is. However, efficient parser can only be build for context free, or context free based grammar. The advantage of guarded parsing is that the parser is basically a context free parser. Only simple modification is required.

This work is a part of the Hong Kong University Machine Translation (HKUMT) project, which is supported by the Industry Department, Hong Kong Government. More detail about the project will be described in the next section.

II. The HKUMT Project

The HKUMT project is our first attempt to Machine Translation between Chinese and English. The source language is Chinese while the target language is English. The project was started in mid 1996, and is

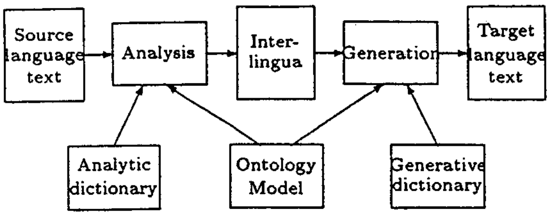

Figure 1: Block diagram of Interlingua approach in HKUMT

expected to finish by mid 1999. We plan to handle the finance domain first.

There are basically three strategies for Machine Translation — The *Direct*, *Transfer* and *Interlingua* approach [4,5] In order to achieve high quality translation, the interlingua approach is selected. This requires understanding the meaning of the source language and represent it in an interlingua text format, which contains only the meaning of the sentence and is supposed to be language independent. Figure 1 shows the block diagram of our approach.

The guarded parsing process is an integral part of the analysis stage. The interlingua text is generated from an internal frame-based structure produced using a rule-based system from the resulting parse tree.

The analytic dictionary include information such as the word sense of the words, the attributes associated with each word sense, such as whether the object is edible, etc. The ontology model include information such as what kind of attributes that an entity can possess, the associated slots in the frame representation, and what information can be filled in the slot. The ontology model also contain the inheritance information between different object classes. The analytic dictionary and the ontology model contain information that is required for the guard checking in the parsing process. More detail of the parsing process will be described in the following section.

If more than one possible parse tree is produced, the parse tree with the highest score is used. The scoring system will not be discussed here.

An frame based structure containing the information about the sentences will be generated from the resulting parse tree using mapping rules for the concepts included in the sentence. The interlingua text representation is then generated from this frame structure. An explicit text representation is required for later extension, when we want to perform translation for more than one language.

The target language generation is relatively simple when compared with the analysis part, because we do not need to handle peculiar sentence structures. Apart from literal beauty consideration, we can always use a particular form for each sentence structure and obtain correct result. Of course we have also to consider rhetoric in order to achieve better translation.

III. Guarded Parser

Guarded Attributed Context Free Grammar (*GACFG*) was first introduced in [1] for syntactic pattern recognition. It has been shown that *GACFG* has the same expressive power as regular tree grammar, web grammar and plex grammar. Since context free grammar is well understood can its parser can be constructed efficiently, it is a natural choice to be the basis.

It is observed that natural language, although not context free, but is mostly context free. Attributed grammar was first used by Knuth [6] to describe the semantic of context free grammar. In order to improve the expressiveness of the language, attributes are used to represent the complicated properties and relationships between the lexical entities.

Definition 1 *A Guarded Attributed Context Free Grammar (GACFG) is an ordered quadruple* (V_T, V_N, S, P) *where,*

V_T	*is a finite set of terminals*
V_N	*is a set of nonterminals*
$S \in V_N$	*is a start symbol*
P	*is a set of production*

Each instance of $x \in V_N \cup V_T$ *is associated with a set of attributes* $\pi(x) \in \Pi$, *(the set of attributes-value pairs), and each production* $p \in P$ *is of the form*

$$p : G | X_0 \rightarrow X_1 X_2 ... X_{np}, E$$

where $np > 0$, $X_0 \in V_N$ *and* $X_j \in V_N \cup V_T$ *for* $1 \leq j \leq np$.

$$G : \Pi^{np} \rightarrow \{0, 1\}$$

is a predicate determined from the attributes, and

$$E : \Pi^{np} \rightarrow \Pi$$

is a semantic rule that determines the attributes of resultant non-terminals.

Each production rule in the context free grammar is consisted of 3 parts — production of ordinary context free grammar, a guard G which determine whether the production can be applied, and a semantic rule E which determines the resultant attributes of the nonterminal thus produced.

The following example is for illustrative purpose, and is much simplified.

Example Consider the very simple grammar rule:

$$VP \rightarrow V\ NP$$

The *GACFG* version of this rule will be:

$$G \mid V \rightarrow V\ NP,\ E$$

where

$$G = compatible(V, NP)$$

$$E: \begin{array}{l} VP(action) = V \\ VP(theme) = NP \end{array}$$

The compatible function depends on the verb, and can be stored in the analysis dictionary entry of the verb. For example, consider the sentence 吃水果 where the compatible entry of the verb "eat" in the dictionary is such that the object should be edible. The dictionary entry of the word "fruit" contains the attribute "edible". The resulting VP will then contain the attributes (action, eat) and (theme, fruit) in the attribute frame.

The parser for *GACFG* can be constructed using a Modified Earley's Parsing Algorithm which will be discussed in Section V..

IV. Sentence Analysis without word segmentation

Word segmentation is traditionally an integral part in Chinese language processing. This is a process to divide a Chinese sentence, which consists of individual characters, into a sequence of words. Conventional methods include maximum matching or statistical approach, which can achieve quite satisfactory result. However, once incorrect segmentation is generated, correct parsing of the sentence cannot be done [7,8].

In our approach, instead of doing word segmentation, we propose to segment the sentence on the fly. A preprocessor called *Lexical Scanner* is inserted between the source language and the parser. It will return a word of a particular word sense starting at the current location, whenever the parser request one. Different possible segmentation is actually generated and being parsed.

Strictly speaking, segmentation is still performed. It also looks like that all possible segmentations are generated. However, this is not the case. Since the lexical scanner only returns a word starting at the current location that matches the word sense that is requested by the parser, not all possibility are actually tested. Furthermore, with the help of guarded parsing, the paths containing invalid segmentation will be pruned away very soon. Figure 2 shows the block diagram of the lexical scanner. We will discuss how this can be incorporated into the Modified Earley parser in Section V..

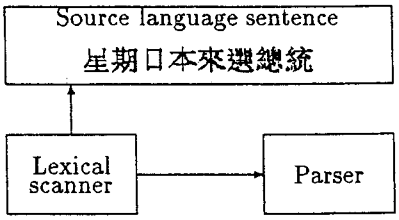

Figure 2: Integrate word segmentation into the parser

V. Modified Earley's Parser

The Earley Parsing Algorithm [9] is a parser for context free grammar. It is consisted of 3 processes — scanner, predictor and completer. Scanner is applied when an input terminal is consumed by the parser. Predictor is to generate new states when a non-terminal is encountered, while completer is used when a grammar rule is to be applied. Please refers to [9] for details of the algorithm. You can also use other parsers [10].

The simplest way is to introduce the guard checking only at the completer stage, when the grammar rule is to be applied. To further improve efficiency, we can divide the guard checking into different stage, so that whenever a lexical unit is encountered, either by a scanner process, when a terminal is encountered or the completer process of another rule, where a nonterminal is scanned over, we can applied partial checking. This have the advantage of earlier pruning of unsuccessful parses so that the number of states of the parse will be much decreased. However, it require more complicated programming and design of the guards.

To incorporate the concept of Lexical Scanner discussed in section IV,. we can modify the scanner process in the Earley parser (hence the use of the name lexical scanner). In the original algorithm, the scanner process read in a terminal. In our revised algorithm, the terminal of the grammar is, instead of words, the word senses such as N, V, ADJ, ADV, PP, etc. During the scanner process, whenever, say, a N is to be scanned, the request is passed to the lexical scanner, which will search from the current position for a Noun. There may be more than one possibilities, because you may get a Noun for one character, two characters, ... Also, the pointers and the states thus generated during the parser process have to be carefully adjusted.

The necessary modification is presented as follows, using the same terminology as in the original paper of Earley's parser.

Completer: If S is final and $\alpha = x$, for some $x \in \{x_1, ...x_n\} - X$, and $G_p(Y) = 1$, then

$\pi(S_{m+1}) \leftarrow E(Y)$ and $m \leftarrow m+1$.

For each $< q, l, g, \beta, X', Y' > \in S_f$ (after all states have been added to S_f), such that $X' = X_f$ where X_f is the first f symbols of X and $C_{q(l+1)} = D_p$, add $< q, l+1, g, \alpha, X, Y' + S_f >$ to S_j.

Scanner: If S is nonfinal and $C_{p(j+1)}$ is a terminal, then if r = lexical-scan($C_{p(j+1)}$), add $< p, j+1, f, \alpha, X + C_{p(j+1)}, Y + C_{p(j+1)} >$ to S_{i+r}, where the function lexical-scan returned a word starting at position i, matching the word sense $C_{p(j+1)}$ and with length r.

VI. Resolve Ambiguity

There exist various types of ambiguities which needed to be resolved during the parsing process. In this section, we will discuss how these ambiguities can be removed or alleviated by using guard checking.

Ambiguity introduced in Segmentation During parsing, whenever the parser request a terminal, the type of terminal is sent to the lexical scanner. If more than one such word exists, both parse will continue, until being pruned because of incompatible guards. For example, consider the Chinese sentence:

星期口本來選總統

For the grammar rule

$$\begin{aligned} S &\rightarrow NP\, VP \\ NP &\rightarrow N \end{aligned}$$

We will find the terminal Noun, for which the lexical-scanner will return both “star”, “week” and “Sunday”. Only one of them can proceed.

Sentence with same lexical units There are sentences which have the same lexical units but of different parse trees. These are typical cases of ambiguities which cannot be resolved by pure syntactic means. For example,

人類登陸月球成為事實
老師希望學生做功課

They both have the same structure

```
Noun Verb Noun Verb Noun
```

but have to be handled by guard checking. For the first sentence, we have the case where Noun Verb Noun forms a sentence, which in term acts as the subject of the sentence. This structure is not possible in the second sentence because 老師希望學生 is not a valid sentence by guard checking. The same applies for the second sentence where the last Noun Verb Noun forms a valid sentence, but not for the first example.

There are cases which cannot be handled even by guard checking, because both cases are acceptable. Then we need to choose one of the interpretation by using scores. For example,

我要努力讀書
我要梳頭化裝

We can associate each binding of words to terminals, and the application of production rules with scores, which reflect the possibility of the association. The parse with higher combined score will be selected.

VII. Conclusion

The technique discussed was used as the front end of the HKUMT project to parse Chinese sentences in the finance domain. Simple grammars is used and ambiguities are resolved by guard checking. Complicated sentences have been tried and successfully parsed. For example, the following sentence will have hundreds of parse trees before using guard checking, which is reduced to only 4. We expect to further reduce the number of parse trees after an improved set of guards and ontology model is completed.

Sentence =

市場憧憬分拆的大昌行,
去年明顯受到金融風暴的波及

The four parses are:

```
1>CLAUSE( NP( CLAUSE( NP( N() ) VP V()
     VP( V() ) ) ) DE() NP( PROPN() ) )
     VP( ADVP( ADV() ) VP( ADVP( ADV() )
     VP( V() NP( NP( NP( N() ) NP( N() ) )
     DE() NP( N() ) ) ) ) ) )

2>CLAUSE( NP( CLAUSE( NP( N() ) VP( VP( V() )
     VP( V() ) ) ) DE() NP( PROPN() ) )
     VP( ADVP( ADV() ) VP( ADVP( ADV() )
     VP( V() NP( NP( NP( N() ) NP( N() ) )
     DE() NP( N() ) ) ) ) ) )

3>CLAUSE( NP( NP( N() ) NP( VP( V() VP(
     V() ) ) DE() NP( PROPN() ) ) )
     VP( ADVP( ADV() ) VP( ADVP( ADV() )
     VP( V() NP( NP( NP( N() ) NP( N() ) )
     DE() NP( N() ) ) ) ) ) )

4>CLAUSE( NP( NP( N() ) NP( VP( VP( V() )
     VP( V() ) ) DE() NP( PROPN() ) ) )
     VP( ADVP( ADV() ) VP( ADVP( ADV() )
     VP( V() NP( NP( NP( N() ) NP( N() ) )
     DE() NP( N() ) ) ) ) ) )

Total of parse trees = 4
```

Furthermore, when attribute building process (by using the semantic rules of the guarded grammar) will produce the internal structure of the interlingua text at the end of the parsing process. The text version can then be generated.

As a conclusion, the guarded parser for Chinese sentence is a promising approach to efficient Chinese sentence processing using simple grammar rules and guard checking.

Reference

[1] K. P. Chan, "Guarded Fuzzy-Attribute Context Free Grammar and its Application on Structural Pattern Recognition," *Int. J. Pattern Recognition and Artificial Intelligence*, Vol. 6, No. 5, 1992.

[2] H. Sun, "A Survey of Verbs used As Nouns in Written Chinese," *Proc. Int. Conf. On Chinese Computing*, Singapore, pp. , 1996.

[3] S. Lu, *Problems of the Grammatical Analysis of the Chinese Language.* Commercial Press, 1979.

[4] S. Nirenburg, *Machine Translation: a knowledge-based approach.* Kaufmann, 1992.

[5] ———, *Machine Translation: theoretical and methodological issues.* Cambridge Univ. Press, 1987.

[6] D. E. Knuth, "Semantics of context free language," *Mathematical Systems Theory*, Vol. 2, pp. , 1968.

[7] S. He, "A Study of Problematic Structures for Chinese Word Segmentation," *Proc. Int. Conf. On Chinese Computing*, Singapore, pp. , 1996.

[8] Y. Wang, "On Chinese Word Division by Computer," *Journal of Applied Sciences*, Vol. 3, pp. , 1985.

[9] J. Earley, "An efficient context-free parsing algorithm," *Commun. ACM*, Vol. 13, No. 2, pp. , 1970.

[10] M. Tomita, *Generalized LR Parsing.* Kluwer, 1991.