Chinese Syntactic Parsing Based on Extended GLR Parsing Algorithm with PCFG*

Yan Zhang, Bo Xu and Chengqing Zong
National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of sciences, Beijing 100080, P. R. China
E-mail: {yzhang, xubo, cqzong}@nlpr.ia.ac.cn

Abstract

This paper presents an extended GLR parsing algorithm with grammar PCFG* that is based on Tomita's GLR parsing algorithm and extends it further. We also define a new grammar—PCFG* that is based on PCFG and assigns not only probability but also frequency associated with each rule. So our syntactic parsing system is implemented based on rule-based approach and statistics approach. Furthermore our experiments are executed in two fields: Chinese base noun phrase identification and full syntactic parsing. And the results of these two fields are compared from three ways. The experiments prove that the extended GLR parsing algorithm with PCFG* is an efficient parsing method and a straightforward way to combine statistical property with rules. The experiment results of these two fields are presented in this paper.

1. Introduction

Recently the syntactic parsing system is one of significant components in natural language processing. Many parsing methods have been developed as the development of corpus linguistics and applications of linguistics. Tomita' GLR parsing (Tomita M., 1986, 1987) is the most general shift-reduce method of bottom-up parsing and widely used in syntactic parsing. Several methods are based on it. Lavie (Lavie A., 1996) used the GLR* parsing algorithm for spoken language system. It uses a finite-state probabilistic model to compute the action probabilities. Inui (Inui K. et al., 1997, 1998) presented a formalization of probabilistic GLR (PGLR) parsing model which assigns a probability to each LR parsing action. To shallow parsing, many researchers have made experiments with identification of noun phrases. Abney (Abney S., 1991) used two level grammar rules to implement the noun phrase parsing through pure LR parsing algorithm.

Some new methods based on GLR algorithm aim to capture action probabilities by statistics distribution and context relations. This paper combines rule approach and statistics approach simultaneously. Furthermore, based on GLR and PCFG, we present an extended GLR parsing and a new grammar PCFG* that provides the action probabilities to prune the meaningless branches in the parsing table. Our experiments are also made in two parts: Chinese base noun phrase parsing and Chinese full parsing. The former is a simplified formalization of full parsing and is relatively simpler than the latter.

This paper includes four sections. Section 2 presents a brief description of rule structure system-PCFG*. Section 3 gives our extended GLR parsing algorithm and the parsing processing. Section 4 shows the experiment results of our parser including Chinese base noun phrases (baseNP) identification and Chinese full syntactic parser. The conclusions are drawn in section 5.

2. A New Grammar (PCFG*) and the Rule Structure

Grammar system is one of the important pars of a parsing system. We explain it in detail in the following section.

2.1 Structure of Rules

The definition of symbols in our system inherits the classifications of Penn Chinese tree-bank (Xia F., 2000). There are totally 33

part-of-speech tags, 23 syntactic tags and 26 functional tags in the Chinese tree-bank tag set. The POS tags belong to terminal symbols, while others belong to non-terminal symbols.

In the final rule base there are about 2000 rules and 400 rules learned from corpus for full parsing and base noun phrases identification respectively. The rules have the following format showed in table 1.

num	rule	probability	frequency
1	VCD→VV +VV	0.754491	126
2	VCP→ VV+VC	0.545455	6
3	VCP→ VV+VV	0.454545	5

Table 1: the format of grammar rules

In order to denote each rule explicitly, the mark '+' is used as the junction mark. In above examples, symbols VP, VCD and VCP are verb phrase and verb compounds. Symbols VV and VC stand for common verbs and copula "是" respectively.

2.2 A New Grammar (PCFG*)

Context-free grammars (CFGs) are widely used to describe the grammar structures in natural language processing. And probabilistic context-free grammars (PCFGs) directly add the probabilities to the rules. But it is sometimes not sufficient to only associate probability with each rule. So we define a new grammar system-PCFG*: each rule is assigned probability distribution and frequency distribution simultaneously. The probability number is the relative value since it is the percentage value in the rule group that have the same left sides. While the frequency number is the absolute value because it is the total numbers occurred in whole corpus. The probability property is the key value to full parsing. The probability attribute is superior to frequency attribute.

A sample is presented to show how to use probability and frequency of a rule.

Suppose there are three rules showed in table 2 and the relations is displayed in figure 1.

Rule	F(r)	P(r)
X→A+C	f1	p1=f1/(f1+f2)
X→A+B+C	f2 >f1	p2=f2/(f1+f2)
Y→A+C	f3 <f1	p3 =1>p1

Table 2: the examples of rule

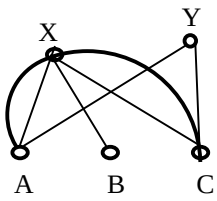

Figure 1: structure of rules

Suppose the input symbols contain A, B and C. When rule 1 and rule 3 simultaneously satisfy the reduce condition, rule 3 is executed and the left side item 'Y' is pushed to the stack because p3 is bigger than p1. To complete parsing, probability always has the priority to frequency. But to baseNP parsing, frequency is superior to probability attribution. Since f1>f3, rule1 is executed first. If f1 is equal to f3, then go on to compare probability.

3. Parsing Algorithm

The parsing algorithm is very significant as well as the grammar rules to the parsing system. We produce an extended GLR parsing algorithm based on the Tomita's GLR parsing algorithm in our system.

3.1 the Extended GLR Parsing Algorithm

The GLR method augments the LR parser and overcomes the drawback of the LR parser. In fact, from the point of parsing algorithm, there are no clear differences between LR and GLR algorithm. In parsing processing, there are also four actions in GLR algorithm that are similar to the LR parsing. But GLR parsing algorithm admits multiple entries in the parsing table. Our extended GLR algorithm also permits that several shift and reduce actions exist in one branch in the parsing table simultaneously. So there are mainly two types of conflicts: shift-reduce conflict and reduce-reduce conflict. These conflicts are the most difficult problems of GLR algorithm. In the parsing process, when the conflicts between shift and reduce occur, the principle of our parsing method is that the reduce action is superior to the shift action.

If only grammar rules are used to describe the context relations, they may produce many conflicts when several rules satisfy the conditions. So we use the grammar system--PCFG* to add statistical information. The probabilities distributions are associated

with the rules to each parsing action and decide which step to continue.

Therefore the extended GLR algorithm handles the conflicts with two steps: (1). The reduce action is always executed first, then the shift action. (2). When more than one reduce actions satisfy the conditions, probability and frequency decide the order of these reduce actions.

3.2 Parsing Actions and Parsing Process

3.2.1 Parsing Table and Actions

The parsing table consists of two sub-tables: ACTION table and GOTO table that are constructed by the grammar rules. The GOTO table is not different from GLR table. Just ACTION table is modified a little. Figure 2 shows the structure of the parsing table.

State	ACTION				GOTO
	X1, X2, ⋯, Xi , #				Y1, ⋯, Yj
S0	Sh1				
S1		Re1			
⋯			Re-Sh		
Sn				Accept	

Figure 2: the parsing table

The ACTION table contains four action sub-tables: Sh1, Re1, Re-Sh and Accept. They stand for shift part, reduce part, reduce-shift part and accept part respectively. Because the error action is similar to accept action, it is not explained here. The Re-Sh part is the key part in the table. It contains multiple entries while the others have no conflicts. In the Re-Sh part, the rules are firstly arranged according to the probabilities and then compared based on the frequencies. The maximum probability is put on the top. This sequence continues until the last rule with minimum probability. According to the order of Re-Sh sub-table, the parsing program is transformed to the corresponding state of the stack. This order suits for the full parsing. But to the base noun phrases identification, frequency is firstly compared.

Since the ambiguities and conflicts existed in the Re-Sh sub-table, we give a limit that no more than 20 entries in the Re-Sh part. From the experiment results, it is better to select 20 rules as the branch limit in the parsing process because it not only permits the multiple entries, but also fits for the performance efficiency of our program.

Since the parser uses PCFG*, it has strong control to handle action conflicts and rule ambiguities. The parsing process need to prune the meaningless parsing branches. Excessive pruning may cause the loss of some grammar rules and add the error opportunities. Reasonable pruning can improve efficiency.

3.2.2 the Parsing Process

We give the following the symbols definition and interpretation to explain the parsing process. Let '#' denotes the start and the end of the input Chinese sentence. The system contains a list of stacks simultaneously. The parsing table contains two elements: *state nodes* and *symbol nodes*. The parsing stack includes state stack (*StateStack*, name in the program), symbol state (SymbolStack) and input stack (*InputStack*) whose pointers are *ps*, *pb* and *pi* respectively.

Following algorithm is established for the shift-reduce parsing process.

Input:

An input Chinese words sequence W in which each word has its part-of-speech and a parsing table produced by grammar rules;

Output:

If the input word sequence W satisfies the grammar rules and is accepted according to the parsing table, then output the parsing result of W, otherwise give error result;

Main Loop:

It mainly consists of four parts: shift, reduce, accept and error in the parsing process.

Repeat
Begin
s := **ps*++; //s is current state
b := **pb*++; //to the next symbol
c := **pi*++; //to the next input word
if *Action[*reduce rule $A \to \partial, A \in Vn, \partial \in Vn \cup Vt$ *]* = reduce(),
then begin

1) Pop $|\beta|$ symbols from top of the symbol stack, and push the left side symbol A to the symbol state;
2) Pop $|\beta|$ symbols from top of the state stack, and push s*

3) *ps* -= | β |; **ps* := s*;
end reduce(); //reduce part

else if *Action[]* = shift(input s*),
then begin
pi++; **pi* := s*; *pb*++; **pb* := s*;
end shift(); //shift part

else if Action[] = accept()
then Success and Output; //the parsing succeeds
else
error(); // parsing is error here
End

Until: The input symbol is the end of the sentence. Or accept function occurs or error function occurs.

(1) Reduce Action

When the reduce action is performed, the rule candidates are selected in the list from the first rule to the last one that are arranged according to the probabilities and frequencies. If one of these rules satisfies the condition, then the flag of this rule is changed from FALSE to TRUE and stop here, and continue to read input word. Otherwise trace back.

(2) Shift Action

Shift action is executed under two conditions. One is based on the action table. The other is that when error action occurs, the base noun phrase identification continues to perform shift action while the full parsing enters trace part.

(3) Error Action

When error action occurs, trace back to the previous branch and perform another rule candidate listed in the entry. If there is no path can be searched in the current branch point or all routes are not passed through, the parsing fails and output the final error symbol. This situation is only used to the full parsing.

3.2.3 the Comparison with GLR

In order to explain explicitly our extended GLR parsing algorithm, we compare it with GLR algorithm. Table 3 gives the comparison results.

methods / aspects	GLR algorithm	Our algorithm
Grammar System	CFG	PCFG*
Statistical Information	no	Probability, Frequency
Data Structure	Graph-Structured Stack	Stack List
Parsing Process	Not simplified	Pruning
Other Attributes	Augmentation to each rule	no

Table 3: Comparison with GLR

4. Experiment and Results

Our experiments include two parts: Chinese base noun phrase parsing and Chinese full syntactic parsing.

The obvious difference of Chinese baseNP parsing and full parsing is that the former must give the parsing results while the latter sometimes need to trace back and output the error symbols. Because baseNP identification belongs to the shallow parsing, it only need to gives the recognized noun phrase structures. If there are no phrases found, then output the original sentence. Obviously Chinese baseNP parsing is much simpler and more efficient than the full parsing from the point of the method and the runtime.

Our experiments are performed based on Chinese tree-bank corpus. There are totally 10,000 Chinese sentences whose grammar structures are described by brackets. Table 4 shows the characteristic of the corpus in the parsing process.

Corpus / Style Of Parsing	Number of the Sentences.	Average length of each sentence
BaseNP Identification	Training: 97%	22 words
	Testing: 3%	15 words
Full Parsing	Training: 98%	22 words
	Test: 2%	15 words

Table 4: characteristic of corpus

To two styles of parsing presented above, we give two types of results respectively.

(1). Chinese BaseNP identification

In our system, base noun phrases are defined to include not only pure noun phrase (NP) but also quantifier phrase (QP), such as QP (一亿多/CD 元/M).

To each Chinese sentence, baseNP identification always gives the final parsing results in which

the base noun phrases are distinguished by brackets. Some samples are listed.

1. 确保/VV 了/AS NP (浦东/NR 开发/NN) 的/DEG NP(有序/JJ 进行/NN)
2. (这/DT 种/M 做法/NN) 受到/VV 了/AS (国内外/NN 投资者/NN) 的/DEG (好/JJ 评/NN)

There are two and three base noun phrases in sentence 1 and sentence 2 respectively.

(2). Chinese full parsing

Following sentences are the results of Chinese full parsing.

1. VP (VP (确保/VV 了/AS) NP (NP (浦东/NR 开发/NN) 的/DEG NP (有序/JJ 进行/NN)))

2.IP (NP (这/DT 种/M 做法/NN) VP(受到/VV 了/AS) NP (NP (国内外/NN 投资者/NN) 的/DEG NP (好/JJ 评/NN))))

In order to display the parsing result clearly, sentence 2 is showed in the tree bank format.

IP (NP (DT 这
M 种
NN 做法)
VP (VV 受到
AS 了）
NP (NP (NN 国内外
NN 投资者)
DEG 的
NP (JJ 好
NN 评)))

Type	Precision (%)	Recall (%)	Num of Rules
BaseNP	87.42	81.4	400
Full parsing	70.56	67.77	2000

Table 5 is the results of these types of parsing.

The experimental results show that our parsing algorithm, extended GLR parsing algorithm, is efficient to both Chinese baseNP parsing and full parsing.

5. Conclusions

In our system, we present the extended GLR parsing algorithm that is based on the Tomita's GLR algorithm. A new grammar system PCFG* based on PCFG is proposed to describe the grammatical rules that are added probability and frequency attributes. So our parsing system combines Chinese grammar phenomena with statistics distribution. This is feasible and efficient to implement Chinese shallow parsing and full parsing. In the future task, we further improve the efficiency and robust of our parsing algorithm and expand Chinese grammatical rules with both statistical attributions and language information. It is important to utilize the results of base noun phrases identification and to improve the precision of Chinese full parsing.

Acknowledgements

The research work described in this paper is supported by the National Nature Science Foundation of China under grant number 9835003 and the National Science Foundation of China under grand number 60175012 and the National Key Basic Research Program of China under grand number G1998030504.

References

Masaru Tomita, Efficient Parsing for Natural Language – A Fast Algorithm for Practical Systems, Kluwer Academic Publishers, 1986

Tomita M., an Efficient Augmented-Context-Free Algorithm, Computational Linguistics, Volume 13, Numbers 1-2, 1987

Inui K., Sornlertlamvanich V., Tanaka H. and Tokunaga T., Probabilistic GLR Parsing: a New Formalization and Its Impact on parsing Performance, Journal of Natural Language Processing, Vol.5, No.3, pp.33-52, 1998

Sornlertlamvanich V., Inui K., Tanaka H. and Tokunaga, T., A New Probabilistic LR Parsing, Proceedings of Annual Meeting of the Japan Association for Natural Language Processing, 1997

Lavie A., GLR*: A Robust Grammar-Focus Parser for Spontaneously Spoken Language, Ph.D. thesis, Carnegie Mellon University, USA, 1996

Abney S., Parsing by Chunks, Kluwer Academic Publishers, 1991

Xia F., the Segmentation Guidelines for the Penn Chinese Treebank (3.0), 2000