

## AN UNGREEDY CHINESE DETERMINISTIC DEPENDENCY PARSER CONSIDERING LONG-DISTANCE DEPENDENCY

Lei Wang

South China Sea Information Center of State Oceanic Administration, Guangzhou, 510300, China.

Email: wangl@scsb.gov.cn

### ABSTRACT

This paper presents a two-step dependency parser to parse Chinese deterministically. By dividing a sentence into two parts and parsing them separately, the error accumulation can be avoided effectively. Previous works on shift-reduce dependency parser may guarantee the greedy characteristic of deterministic parsing less. This paper improves on a kind of deterministic dependency parsing method to weaken the greedy characteristic of it. During parsing, both forward and backward parsing directions are chosen to decrease the unparsed rate. Support Vector Machines are utilized to determine the word dependency relations and in order to solve the problem of long distance dependency, a group of combined global features are presented in this paper. The proposed parser achieved significant improvement on dependency accuracy and root accuracy.

**Keywords:** Ungreedy, Chinese dependency parser, Deterministic, Combined global features.

### 1. INTRODUCTION

Syntactic is an important basement of Natural Language Processing (NLP). Many syntactic analyzers for English have demonstrated good performance [1, 3]. But Chinese syntactic parsers do not perform very well. That is because Chinese language, which has more complexity syntax structures, is different from other languages and we couldn't choose the experience in processing western languages directly.

About the previous work of Chinese dependency parser, Zhou [6] proposed a method of exacting phrase structures in the sentences and parsing them separately. In Cheng [9], a NP-Chunk is introduced and headword is parsed in the next step. The essential idea of deterministic parsing is estimating a globally optimal solution in a sequence of locally optimal choices and it has the characteristic of greedy [10]. This greedy algorithm is simplicity and efficiency. But sometimes, suffers from the problem of error accumulation. Jin [2] used two-phase parsing for Chinese. This method not only decreases the early reduce caused by verbs but also weakens the greediness of deterministic parsing. But it couldn't avoid the error caused by preposition. In order to solve the problem of error accumulation, Cheng [4] divided one sentence into two parts by the root word and parsed them separately. But the unparsed rate isn't very satisfactory. In order to decrease the unparsed rate, we propose a method of using both forward and backward parsing direction.

In Nivre's algorithm [4], the operation reduce has a precondition that the node  $N$  should have no other child anymore. It is difficult to check this condition especially in long distance dependency. Jin [2] avoids this problem successfully by using the two-phase parsing method. Cheng [4] presents a method of using global features to solve long

distance dependency relation problem. In order to improve the parsing accuracy, we have made many experiments to explore the effective global features and proposed a group of combined global features.

This paper is organized as follows. Sentence segmentation is introduced in section 2. Section 3 describes the parsing method including the parsing algorithm and parsing direction. Section 4 presents the experimental results. Discussion is described in section 5 and future work is presented in section 6.

### 2. SENTENCE SEGMENTATION

Root plays a very important role in a sentence. Some previous works have showed that the information of root is a beneficial feature for dependency parsing. They considered the information of root word as a new feature of machine learning. In a sentence, there is one and only one root word in a dependency structure. After analyzing the dependency structure of sentences in Penn Chinese Treebank 5.0, a phenomenon can be found easily, that is root is a key node of one sentence and it can link sub-sentences into a whole one. In Isozaki's work [5], their experiments showed that information of root can be used as a feature of machine learning. On the other hand, it can also be used to divide the sentence into two sub-sentences. Then we can analyze the sub-sentences respectively. By this way, the error-propagation can be prevented because the sub-sentences are self-governed. The error which happens in one sub-sentence does not affect the other one.

Cheng [4] proposed a method of dividing sentence by root. As in Figure.1, The root divides the sentence into two parts. The root word of the original input sentence is “或者 (or)”.

So we can divide the sentence into two sub-sentences: “开车 (drive) / 去 (go) / 郊游 (outing) / 或者 (or)” and “或者 (or) / 到 (go) / 海边 (seaside) / 去 (do) / 游泳 (swim)”. Both the sub-sentences have the root word: “或者 (or)”. First, analyze the dependency structure of sub-sentences separately. Then, by combining the two dependency structure we can get the full structure of original sentence.

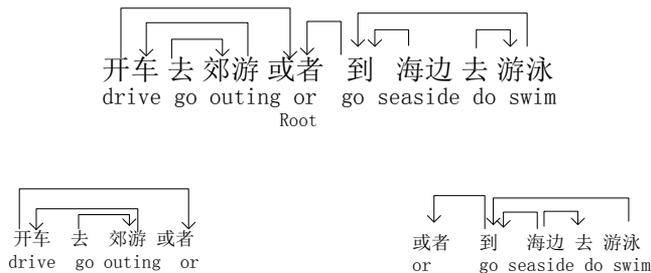


Figure 1. Dividing sentence

## 2.1 Searching root

Isozaki [5] constructed a root finder in their system to find the root word of the input sentence. Similarly to Isozaki’s work [5], we consider the root searching problem as a classification. A set of features is used to evaluate each word in a tagged sentence. We select SVM to construct the root searcher. The classifier is trained to classify each word to be root or not. Then the word with the highest classification score is chosen as root. In addition to the features used in Isozaki’s work [5], we introduce other effective features including the word relationship, the number of tokens, the tokens between the child and the parent and the distance between child and parent.

## 3. PARSING METHOD

This section describes the second part of our two-step dependency parser. In step two, we parse the sub-sentences separately and combine the sub-tree into a whole dependency tree by making the root as the head of the tree. We choose the parsing algorithm proposed by Nivre [1] as the basic algorithm

Previous works have showed that global features can improve the accuracy of classification effectively [8]. Therefore, searching effective global features is an important and challenging work. In this section, we present a group of effective combined global features to deal with long dependency parsing problem.

Parsing direction is also a key factor influencing the parsing accuracy. After dividing the sentence into two sub-sentences, it is clear that the position of the root is either the front of the sub-sentence or the end of the other one. Our strategy is to parse the root word first when parsing the sub-sentences. The experiments indicate the number of unparsed words will reduce markedly if we parse the root word first.

### 3.1 Parsing algorithm

Nivre [1] proposed a shift-reduce dependency parsing algorithm which can parse sentences in linear time. In the

algorithm, two stacks  $S$  and  $I$  are defined. All the words which are being in consideration are kept in the stack  $S$  and stack  $I$  remains all the unparsed words. Queue  $A$  is defined to store all the dependency relations created by the parser. At the beginning, stack  $I$  contains all the words in the given sentence. Stack  $S$  and queue  $A$  are empty. Then, the word on the top of stack  $I$  is pushed into stack  $S$ . The focused words are the top elements of stack  $S$  and stack  $I$ . Suppose the focused word pair is  $\langle m, n \rangle$ . The four transitions are defined for word  $\langle m, n \rangle$  as follows:

*Left*: if word  $m$  depends on word  $n$ , push the dependency relation  $(m \rightarrow n)$  into  $A$  and push  $n$  into  $S$ .

*Right*: if word  $n$  depends on word  $m$ , push the dependency relation  $(n \rightarrow m)$  into  $A$  and pop  $m$ .

*Reduce*: If word  $m$  has its head in stack  $S$  and there is no more word  $m'$  in stack  $I$  which depends on  $m$ , then pop  $m$ .

*Shift*: If there is no dependency relation between  $m$  and  $n$  and word next to  $m$  in stack  $S$  is not  $m$ ’s head, then push  $n$  into stack  $S$ .

Many papers have showed that Nivre’s algorithm has some mistakes in determining the right-side dependents because of the wrong decision of reduce transition. Due to this, some potential head words will lose the chance to be head of the following words. Jin [2] pointed out that in Chinese, the problem occurs mainly in sentences with verbs or prepositions whose children are on the right side. Such situations are categorized as V-V and V-P, where, V-V means the top of the stack and the next input word are verbs. While, V-P means the top of the stack is a verb and the next input word is a preposition [11].

Jin [2] proposed a two-phase method to solve the problem of early reduce caused by V-V. In the paper, the parsing procedure is divided into two phases. In phase 1, three actions including Shift, Left-Arc and Right-Arc are defined. Right-Arc is, differently from that in [1], does not push the word to the stack. The action of V-V is always Shift at phase 1. The parser finishes phase 1 with a stack containing and with all the verbs of original sentence and with all dependency relations between verbs not being constructed. Right-side verbal dependents are determined in phase 2 according to the contents of the stack generated in phase 1. Then, the entire dependency structure is built.

The two-phase method has two advantages. Firstly, It avoids the probability of errors caused by reduce. Secondly, after phase 1, more contexts are available for phase 2 to determine the dependency relation between verbs [11]. So that according to these information, the parser can choose the most appropriate parsing action much more easily at every step to reduce the greediness of deterministic algorithm effectually.

The case of V-P can be dealt with in a similar way. Experiments of the two-phase method about V-P and V-V are both carried out. During parsing, the actions for V-P and V-V are all Shift in phase 1. After phase 1, a stack containing verbs and prepositions in reverse order of the original sentence is generated. The dependency relations among these words will be constructed in phase 2.

### 3.2 Parsing direction

As to the parsing direction, few works have been done on this subject. In our experiments, the results show that it is also an important factor which affects the parsing accuracy in a certain extent. During the process of parsing, we use both

forward and backward parsing direction according to the character of Chinese. As shown in Figure 2, after the sentence segmenting in step 1, the original sentence has been divided into two sub-sentences by the root. The root word has reversed position in the sub-sentences. If the root of the sub-sentence is in the end, we choose the forward parsing direction, otherwise the parsing direction we choose is backward. That is to say, every time, root is the first word to be parsed. The experiments showed that the unparsed rate decreases to 2% if we use both the forward and backward direction.

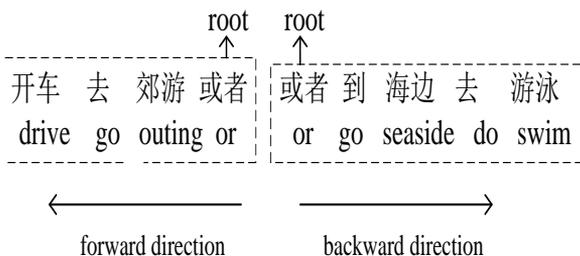


Figure 2. Parsing direction of sub-sentences

Finally, we combine the dependency structure of sub-sentence by making the root word as the head of the tree.

### 3.3 Dealing with long distance dependency

During parsing, the parser extracts the effective features to decide the optimum action of every step. In Chinese sentence, some words depend on their neighbor nodes but some may have a child in the remaining token sequence. If the distance between the two words is far, it is difficult to check the dependency between them.

Cheng [4] pointed out that lacking long distance features may cause decision-making error easily. Take a sentence for example. In Figure.3, assume the parser is analyzing the relation between focused words “转告 (tell)” and “他(him)”, then features considered in this original analysis are the information of words “请(please)”, “转告(tell)”, “他(him)”, “(我 me)”, “三点(three o'clock)” and “准时(on time)”. These features used are local feathers, including lexical forms, POS tags of parent tokens, child tokens, top word of the stack, input word and next word of the input.

It is clear that the word “转告 (tell)” has a child “到达 (arrive)” which is not yet analyzed, so the correct action of this situation is “Shift”. But actually, the parser may possibly estimate the action as “Reduce”. That is because the information of “到达 ( arrive )” is not included in the local features. Such an error is caused mainly by lacking long distance information. In order to solve this problem, Cheng [4] proposed a method of using global features. The information of the words which remain in stack I but haven’t been considered in local features is defined as global features. In Figure.3, dotted line identifies local features and solid line signs the global features.

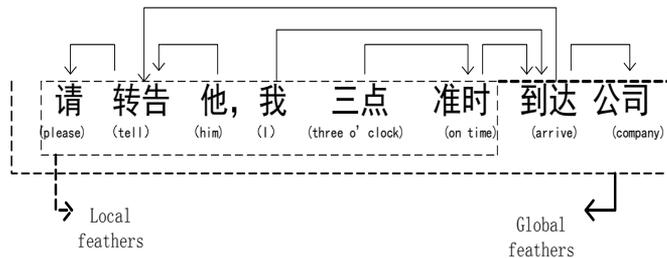


Figure 3. Description of local and global features

Previous works have showed that using global features which capture different information in dependency tree can improve the parsing accuracy, especially that of the long dependency parsing. Therefore, searching for effective global features is a significant work. We try to explore the combined global features in order to obtain a higher parsing accuracy. After many experiments, the combined global features we use are as follows:

- (1) the child node;
- (2) the parent node;
- (3) the grandparent node;
- (4) the nearest outer sibling node;
- (5) the direction from the parent node to child node;
- (6) the distance from parent node to child node.

## 4. EXPERIMENTAL RESULTS

We use Penn Chinese Treebank 5.0 as data set which contains 500k words mostly from different resources of XinHua newswire and Hong Kong news. To transfer the phrase structure into dependency structure, head rules are defined based on Xia’s head percolation table [7]. The same data set is also used to train the root finder. For evaluation matrix, we use dependency accuracy and root accuracy defined by Yamada [3]. We also use none head rate defined by Jin [2].

We make an experimental comparison with Maltparser 0.4, an implementation of Nivre’s algorithm. Maltparser chooses SVM and memory-based learner as its machine learning methods. In our parser, we use the software packages Lib-SVM for machine learning. The Training set used in the comparison is the Penn Chinese Treebank 5.0. Table 1 shows that 1.51% of the word cannot find their head after parsing if using Nivre’s algorithm.

Table 1. Comparison of dependency accuracy

Parsing Strategy	Dependency accuracy	Root accuracy	None head
Nivre’s algorithm	83.34%	69.38%	1.51%
Our parser	85.65%	72.35%	----

In this paper, we propose a two-step parsing method only for Chinese. We compared this method to the one-step parsing. The same combined global features are used in both of them. Results (Table 2) show that because of the shortening of sentence length and the prevention of error accumulation parsing, we got 1.53% increase on dependency

accuracy. In addition, the proposed parser got 1.37% increase on root accuracy.

**Table 2.** Comparison of one-step and two-step parsing

Parsing Strategy	Dependency accuracy	Root accuracy
One-step parsing	83.12%	70.98%
Two-step parsing	84.65%	72.35%

As to parsing direction, we use both the forward and backward direction, because the first word we parse is always the root word. Thus we can decrease the unparsed rate in a way. We also estimate average time the parser spends when parsing one sentence. Average sentence length is 16.32 words. The experiment result about unparsed rate as follows (Table 3). It is clear that the unparsed rate reduce to 2% if we use both forward and backward parsing directions.

**Table 3.** Comparison of different parsing directions

Parsing direction	Average time(sec)	Unparsed rate
Forward direction	19.42	5%
Backward direction	20.13	4%
Forward & Backward direction	22.57	2%

Although, comparing with the forward direction, the strategy of using two parsing directions cost 3.15 seconds more, it is more significant in improving the unparsed rate.

## 5. DISCUSSIONS

In this paper, we propose a two-step dependency parser to parse Chinese deterministically. Step1: dividing the sentence into sub-sentence and Step 2: parsing the sub-sentence separately and linking the sub-tree into a whole one. In step 2, we improve the method proposed by Jin [2] to avoid the problem of early reduce caused by V-P. Thus, it can weaken the greediness of deterministic algorithm. In order to deal with the problem of parsing long distance dependency, we explore the effective combined global features to improve the parsing accuracy. Results show that comparing with the deterministic parser which parsed a sentence in a sequence, the proposed parser achieved extremely significant increase on dependency accuracy and root accuracy. We should pay attention to the problem of average parsing time of each sentence. When parsing one sentence, our experiments show that compared with one-step parsing, the proposed parser use 3.18 more seconds averagely. In addition, we use different parsing directions according to the root position in a sentence. This method can improve the unparsed rate effectively.

## ACKNOWLEDGMENT

We would like to thank every member of the research group for their helpful discussions.

## REFERENCES

1. Joakim Nivre, Mario Scholz, Deterministic Dependency Parsing of English Text, *Proceeding of the 20th International Conference on Computational Linguistics COLING*, Geneva Switzerland, pp. 64-70, 2004.
2. Meixun Jin, Mi-Young Kim and Jong-Hyeok Lee, Two-Phase Shift-Reduce Deterministic Dependency Parser of Chinese, *Proceeding of the Second International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 256-261, 2005.
3. H., Yamada and Y., Matsumoto, Statistical Dependency Analysis with Support Vector Machines, *Proceeding of IWPT*, pp.195-206, 2003.
4. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto, Chinese Deterministic Dependency Analyzer: Examining Effects of Global Features and Root Node Finder, *Proceeding of the Fourth SIGHAN workshop on Chinese Language Processing*, pp.17-24, 2005.
5. Hideki Isozaki, Hideto Kazawa and Tsutomu Hirao, A Deterministic Word Dependency Analyzer Enhanced With Preference Learning, *Proceeding of The 20th International Conference on Computational Linguistics COLING*, Geneva, Switzerland, pp.275-281, 2004.
6. Ming Zhou, A Block-Based Robust Dependency Parser for Unrestricted Chinese Text, *Proceeding of the Second Workshop on Chinese Language Processing*, pp.78-84, 2000.
7. Fei Xia, Martha Palmer, Converting Dependency Structures to Phrase Structures, *Proceeding of the First International Conference on Human Language Technology Research*, pp.1-5, 2001.
8. Tetsuji Nakagawa, Multilingual Dependency Parsing using Global Features, *Proceeding of the CONLL Shared Task Session of EMNLP-CONLL*, pp.1175-1181, 2007.
9. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto, Machine Learning-Based Dependency Analyzer for Chinese, *Proceeding of the International Conference on Chinese Computing (ICCC)*, vol.15, No.1, pp.13-24, 2005.
10. Johan Hall, Joakim Nivre and Jens Nilsson, Discriminative Classifiers for Deterministic Dependency Parsing, *Proceeding of COLING/ACL*, Sydney, pp.316-323, 2006.
11. Xiangyu Duan, Jun Zhao and Bo Xu, Ungreedy Methods for Chinese Deterministic Dependency Parsing, *Proceedings of Twenty-second Conference of Association for Artificial Intelligence Student Session (AAAI)*, pp.22-26, 2007.