Two-pass Deterministic Dependency Parsing for Long Chinese Sentences

Ping Jian, Chengqing Zong
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
Beijing, China
E-mail: {pjian, cqzong@nlpr.ia.ac.cn}

***Abstract*—This paper proposes a two-pass parsing approach to improve the performance of deterministic dependency parser for long Chinese sentences. In the first pass, the sentence is divided by every comma, semicolon and colon to be parsed separately, and in the second pass, the built sub-trees are integrated to do a reparsing. The error propagation of the deterministic parser is effectively reduced through the two-pass parsing. The experimental results show that although neither punctuation classifier nor root finder is adopted, the proposed parser still achieves a desirable performance, especially the root accuracy.**

***Keywords*—dependency parsing; long Chinese sentences; punctuation-based segmentation; skeleton; reparsing**

I. INTRODUCTION

Recently, deterministic model has become one of the most popular data-driven dependency parsing approaches for its comparable parsing accuracy to state-of-the-art parsers but linear parsing time [1, 2]. However, when sentences become longer, the parsing performance deteriorates seriously because of the error propagation coming from the greedy strategy adopted by the models. It gets more severe when parsing Chinese sentences. Because Chinese is a kind of ideographic language that simple sentences sharing one identical complete meaning can be connected only by punctuations without any obvious conjunctions.

Sentence segmentation is one of the effective avenues to handle the problems in long sentence parsing. Cheng et al. [3] constructed a root finder to identify the root node and parsed the sentence segmented by the root node separately. Besides the root node, punctuations that commonly appear in long sentences are also good separators for sentence segmentation. Shiuan and Ann [4] segmented English sentences at prosodic and clausal conjunctive commas, clausal conjunctions and subordinating prepositions. The separate parse trees of the segmented sub-sentences were then synthesized to form the final parse. A neural network was adopted to disambiguate the roles of the commas, conjunctions and prepositions in the sentences. Considering that [4] sieved out only two roles of commas, prosodic and conjunctive, Jin et al. [5] analyzed the complete usages of the comma in Chinese, where the punctuations are more important cues for sentence decomposition. But the "phrase" and "clause", two indicators they employed to identify the punctuation roles, are not quite distinctively used in practical. Different from the work mentioned above, Mao et al. [6] categorized the chunks divided by the punctuations instead of the punctuations themselves. Chunks which could be identified as the "separate parsing phrase" were parsed individually and the parsed structures were incorporated into that of the other chunks. For dependency parsing, Yu et al. [7] argued that punctuations whose head is the root of the sentence can divide the sentence into basically independent ones. In their work, the sub-sentences divided by the punctuations were parsed separately using the deterministic parser and adjoined to the final tree through combining their roots. This strategy decreases the average dependency length effectively but it is designed to only cut the highest branches of the dependency tree. The embedded but complex elements such as subordinate clauses can not benefit.

Actually, let's not discuss whether the separating potential of punctuations is well exploited or not, an inherent problem this kind of punctuation-based segmentation approaches facing is how to identify the roles of the punctuations (or the chunks delimited by the punctuations) that have been defined in the sentences. The experimental results in previous works imply that it is not an easy task especially for Chinese [4-7]. One of the reasons is that the discriminating ability of the features extracted from sides of the punctuations becomes weaker when the number and the length of the segments increase. Similarly, models reckoning on root finder also suffer from the low precision of the root identification. It thus shackles the improvement of the final parsing performance they deserved.

Li et al. [8] presented a hierarchical approach for constituency analysis which needn't to classify the punctuations automatically. Every comma, semicolon and colon was sorted as the "divide" punctuation to segment the sentence, and other punctuations were induced to expand the PCFG grammars. However, the "improper division detector" they applied could only find a part of wrongly divided structures.

In this paper, we propose a novel two-pass approach for long Chinese sentence deterministic dependency parsing. Similar to [8], the sentences are divided by every comma, semicolon and colon to be parsed separately[1] in the first pass. But in the second pass, parts of the sub-trees generated in the first pass are selected to do a reparsing to complete the final tree. Compared with the existing approaches in literature, our parser has the following characteristics: (1) neither a root finder nor an automatic punctuation classifier is necessary; (2) a novel reparsing is employed instead of just adjoining the segment parsing

[1] We simply call the fraction split by these three types of punctuations as *segments* in this paper.

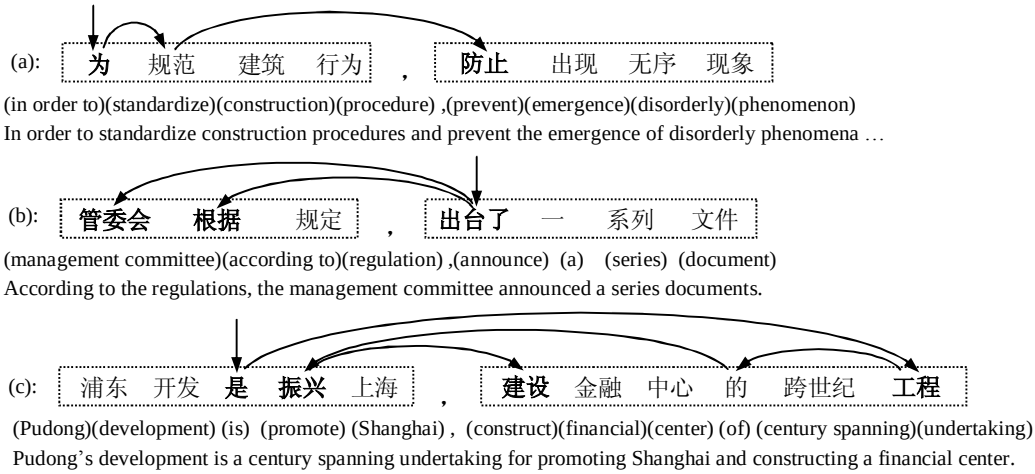

Figure 1. Examples of the segment depending structures. Words which are not linked by arcs own their head inside the segment. The roots of the segments are in bold.

outputs; (3) the input of the higher level parsing still keeps a relatively complete syntactic and semantic format.

Remainder of the paper is arranged as follows: Section II surveys the depending structures of the segments separated by commas, semicolons and colons. Section III describes the proposed two-pass approach and the experimental analysis is presented in Section IV. Conclusions and future work are drawn in Section V.

II. SEGMENT STRUCTURE ANALYSIS

Before giving a detail description of the proposed approach, we define some glossaries to describe the dependency structures of the segments:

(1) *unified:* single rooted;
(2) *centralized:* only the roots have relations with other segments;
(3) *self-governed:* unified and centralized.

In Fig. 1, the segments on the right side of (a) and (b) are self-governed. The segment on the left side in (a) is unified but not centralized, while the ones on the left side in (b) and (c) are centralized but not unified. The one on the right side in (c) belongs to none of the three types. All of these complicate the analysis of the punctuations and the segments they delimit. In case (a), although all the words in the left segment are governed by the root "为 (in order to)", the relationship of the two segments exists between the words "防止 (prevent)" and "规范 (standardize)", not the word "为". In this situation, parsing the segments solely and choosing the roots of them to carry out the remainder analysis may render a great amount of attachment errors.

III. TWO-PASS DEPENDENCY PARSING

A. The First Pass

In the first parsing pass of our parser, all the commas, semicolons and colons in the sentence are regarded as the separating punctuations and the separated segments are analyzed deterministically and independently. To cater for the disunified cases among the segments, we try to acquire appropriate number of unattached tokens, i.e., the roots of the segments, which we call *sub-roots*. When the *arc-standard* deterministic parsing algorithm [9] is adopted, the preliminary experiments reveal that attaching root nodes with a default root label after the parsing process will reserve more unattached tokens than attaching them during parsing.

B. The Second Pass

Most of the existing approaches regard the self-governed segment as the smallest unit to be parsed since it expresses the minimal integrated syntactic structure. In our approach, these integralities are destroyed partially because of the loose segmentation strategy. To solve this problem, we design a reparsing on the pre-parsed sub-trees.

We found that the words depending directly on the roots of the sentence segments are more prone to have relations with words in other segments besides the roots themselves. As far as Penn Chinese Treebank V5.0 [10] is concerned, the first layer children of the segment root own 5 times more exterior links than other child words do. Furthermore, most of these words are the key elements to constitute the main structure of the initial sentence, such as the subject and object of the predicate verb. Therefore, we select the top layer children of the sub-roots of the separate parse trees to do a reparsing together with the sub-roots themselves and separating punctuations. Under this framework, some cue words which are important for clausal level parsing will also survive. For instance, the adverbial predicate modifiers, the clausal conjunctions and the subordinating conjunctions. All of these make the second parsing pass as a reasonable *skeleton-based* parsing.

In this way, the breakages of the structure will be partially corrected by the reparsing. Take the uncentralized case in Fig. 1 for example. Word "规范" is a direct child of sub-root "为" and its recall favors word "防止" to be attached correctly. Compared with the detection and combination strategy in [8], this kind of reparsing possesses more universality for structure retrieval. Fig. 2 is an instance of the skeleton extraction and reparsing procedures. It can be seen that the skeleton still has a reasonable syntactic and semantic form as a normal sentence. It is helpful for the higher level structure analysis.

The tail of the parsing procedure is to combine the outputs of the reparsing pass and the sub-structures built in the first pass to form the whole dependency tree of the

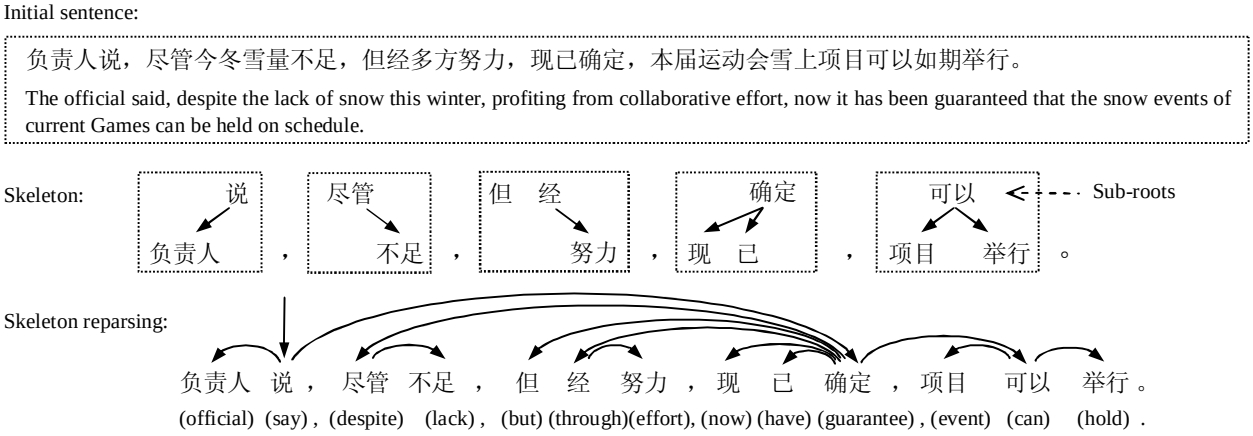

Figure 2. An initial sentence and its skeleton-based reparsing. The arcs of the punctuations are not drawn.

initial sentence. For the words whose head is assigned twice in the two passes, the result of the second pass is completely used.

IV. EXPERIMENTS

We presented the experimental evaluation on data in Penn Chinese Treebank V5.0. The corpus was split into training, development and testing set as [11] did to balance the different resources. 16,079 sentences were for training, 803 for development and 1,905 for testing. All the sentences with more than 27 words in the testing set were extracted as the "long sentences" and 790 ones were obtained in an average length of 43.7. The threshold 27 was chosen according to the average sentence length of the corpus, 27.1. It is 3.44 separating punctuations (commas, semicolons or colons) in average per extracted testing sentence.

The original constituent structures of the sentences were converted to dependency ones using head rules as what were used in [12]. The conversion was accomplished by the software Penn2Malt[2]. Gold standard POS tags from the corpus were used in all of the experiments.

We chose MaltParser [13] V1.1 as the deterministic parser for our two-pass parsing. The *arc-standard* algorithm [9] was used for both of the two passes and the *Classifiers Splitting* strategy was involved to save the training and testing time. In addition, the *strict* root handling mode which does not attach root nodes during parsing was employed for the segment analysis.

Four parsers are compared in the experiment:

(1) **Baseline:** Parse in sequence using MaltParser;

(2) **Sub-root only:** Only the roots of the segments are extracted to be parsed in the second pass of the two-pass parser and the final tree is composed by the outputs of the two passes;

(3) **Sub-root limited:** Use a smaller amount of sub-roots to perform the two-pass parser. It is achieved by replacing the *strict* root handling mode with the *normal* one when parsing the segments. It is another strategy that the MaltParser handles the root attachment, where roots nodes are attached during parsing. The sub-root limited system is designed to examine the effect of the sub-root amount to the parsing accuracy;

TABLE I

FEATURE SET FOR THE DETERMINISTIC PARSING. T AND N DENOTE THE TOKENS IN THE STACK AND THE INPUT SEQUENCE RESPECTIVELY. lc AND rc REPRESENT THE LEFTMOST AND RIGHTMOST CHILDREN OF THE SPECIFIED TOKEN.

Feature type	T2	T1	T0	N0	N1	N2	N3
pos	+	+	+	+	+	+	+
word		+	+	+	+	+	
pos_lc			+				
pos_rc		+					
word_rc			+				
relation_lc			+	+			
relation_rc			+	+			

(4) **Two-pass:** the exact proposed scheme.

All the parsers used the same feature model which is compiled in Table I. This set is similar with the one used by MaltParser V0.4 for Chinese parsing in the shared task of CoNLL2007 [3]. The follow metrics are used for evaluation:

Dependency accuracy (DA): The proportion of non-root words that are assigned the correct head;

Root accuracy (RA): The proportion of sentences in which the root word is correctly identified;

Parsing Time: The average parsing time per sentence under the same system condition;

Amount of the sub-roots (Root): The number of the sub-roots built in the segment parsing pass. For the one-pass system (the baseline), it is obtained by dividing the outputs of the parser at comma, semicolons and colons and counting the words whose head is outside the segment.

A. Results

Table II reports a quantitative evaluation of the parsing results. According to the performance on the development set, we did not extract the "AS" tagged words and the direct children of the nominal sub-roots when analyzing the testing set in the two-pass approach.

The proposed two-pass parser almost achieved the best performance among the competitors and outperformed the baseline parser 1.75% on dependency accuracy and 6.45%

[2] http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html

[3] http://w3.msi.vxu.se/users/jha/conll07/

TABLE II

MAIN RESULTS OF THE COMPARED PARSERS

Parsing strategy	DA (%)	RA (%)	Parsing time (sec)	Root
Baseline	80.81	60.89	1.96	4006
Sub-root only	81.21	55.82	1.17	3901
Sub-root limited	82.12	67.72	1.74	3579
Two-pass	82.56 (+1.75)	67.34 (+6.45)	1.48 (-0.48)	3901

on root accuracy. There is a little deterioration on dependency accuracy when a small sub-root set (3579) is utilized. Actually, this number is 3915 in the gold standard text. It is the exact two-pass parser whose sub-root set size is the closest one to this number. The bigger amount of the unattached words in the segments of the one-pass parser results implies that parsing sentence in sequence blurs the independency between the segments to some extent. The paradigm that just reparsed the sub-roots got a lower accuracy, which demonstrates that the loose segmentation strategy indeed brings many integrality destructions. In addition, the context information is scarce for the parser to analysis the higher level structure if only the roots of the segments are taken as the input. The sentence root identification precision is inevitablely poor.

Compared with the existing work [7] which also used Nivre's parser [2, 9] for Chinese dependency parsing, the improvement of the two-pass parser is satisfying. In [7], additional root finders and baseNP chunker were adopted to enhance the parser and the final parser got an increase of 2.38% on dependency accuracy against the baseline for complex sentences and 0.3% on root accuracy. Remarkably, the increase of the root accuracy of our parser is pronounced. In addition, since the sequences to be parsed are simpler and no accessional classifier is needed, our parser runs faster than the baseline.

Moreover, the improvement of the two-pass parser climbs as the sentence length increases. A gap of 2.09% on dependency accuracy was accessed when we tested on the sentences longer than 40 words. About 4.5 separating punctuations are located in each of these sentences.

B. Analysis

Like the existing sentence segmentation approaches, the two-pass parser aims to reduce the error propagation of the deterministic parsing models by shortening the dependency length of the input. In our experiments, the average length of the word sequences to be parsed is reduced from 43.7 to 10.7 and the average dependency length is decreased to 2.94 from 4.71. These large curtailments enable our approach to be powerful.

It is interesting to count the amount of the clause level cue words which the skeleton can reserve in the two-pass parsing. Table III gives a summary where **Clause AD** represents the adverbs that modify the verbal root of clauses, **Clause CC** represents the conjunctions connecting clauses and **CS** is the subordinating conjunctions. Since the two-pass parser takes a segment as the processing unit, the AD/CCs here also include the ones modifying the verbal roots of the segments. It is clear that most of these words are survived and employed for analyzing the higher level structure of the sentence. We also examined the

TABLE III

RESERVATION OF THE CUE WORDS

	Clausal AD	Clausal CC	CS	Miss head
Initial	1680	84	59	-
Skeleton	1539	76	58	51

TABLE IV

PARSING RESULTS OF THE NORMAL LENGTH SENTENCE

Parsing strategy	DA (%)	RA (%)
Baseline	82.43	72.44
Two-pass	83.87 (+1.44)	75.91 (+3.47)

number of words that still can not find the head in the skeleton even the first layer children of the sub-roots are recalled. Only 51 words in the 790 testing sentences missed their head which reveals that the skeleton extracting strategy is sound to deal with the segment level dependency structure.

Table IV displays the accuracy of the parsers performed on all the 1,905 testing sentences of which the average length is 26.4. Each sentence has 1.89 separating punctuations in average. The two-pass parser still gains the better performance than the baseline. It proves that our approach is also effective for normal sentences.

V. CONCLUSION

In this paper we present a novel two-pass deterministic parser, in which the sentence is divided by every comma, semicolon and colon to be parsed separately and a skeleton-based reparsing is introduced to mend the parsing errors brought by the loose segmentation strategy. It bypasses the intractability of constructing a punctuation classifier. On the other hand, the separating ability of the punctuations is extremely exploited under this strategy. The reparsing of the direct dependents of the segment roots makes the parser easy to grasp the higher level structure of the input sentence, especially for the root identification. Experimental results and comparisons with other approaches confirm the effectiveness of the proposed one.

Future work may lie in specifying the finer skeleton extracting strategies and developing the more effective features. Different feature sets may be used to the two parsing passes of the parser to further improve the parsing accuracy.

ACKNOWLEDGMENT

The research work described in this paper has been supported by the Natural Science Foundation of China under grants 60723005, 60736014 and the Hi-Tech Research and Development Program (863) of China under grant 2006AA010108-4, and also supported by the National Key Technology R&D Program of China under grant 2006BAH03B02.

REFERENCES

[1] H. Yamada and Y. Matsumoto, "Statistical dependency analysis with support vector machines," Proc. IWPT, Nancy, France, Apr. 23-25, 2003, pp. 195-206.

[2] J. Nivre, "An efficient algorithm for projective dependency parsing," Proc. IWPT Nancy, France, Apr. 23-25, 2003, pp. 149-160.

[3] Y. Cheng, M. Asahara, and Y. Matsumoto, "Chinese deterministic dependency analyzer: examining effects of global features and root node finder," Proc. SIGHAN, Korea, Oct. 11-13, 2005, pp. 17-24.

[4] P. L. Shiuan and C. T. H. Ann, "A divide-and-conquer strategy for parsing," Proc. ACL/SIGPARSE, Santa Cruz, USA, Jun. 24-27, 1996, pp. 57-66.

[5] M. Jin, M-Y. Kim, D. Kim, and J-H. Lee, "Segmentation of Chinese long sentences using commas," Proc. SIGHAN, Barcelona, Spain, Jul. 25-26, 2004, pp. 1-8.

[6] Q. Mao, L. Lian, C. Zhou, and C. Yuan, "Chinese syntactic parsing algorithm based on segmentation of punctuation," Journal of Chinese Information Processing, Commercial Press, vol. 21, issue 2, Mar. 2007, pp. 29-34. (in Chinese)

[7] K. Yu, S. Kurohashi, and H. Liu, "A three-step deterministic parser for Chinese dependency parsing," Proc. NAACL, Rochester USA, Apr. 22-27, 2007, pp. 201-204.

[8] X. Li, C. Zong, and R. Hu, "A hierarchical parsing approach with punctuation processing for long Chinese sentences," Proc. IJCNLP, Korea, Oct. 11-13, 2005, pp. 7-12.

[9] J. Nivre, "Incrementality in deterministic dependency parsing," Proc. ACL, Barcelona, Spain, Jul. 21-26, 2004, pp. 50-57.

[10] N. Xue, F. Xia, F.-D. Chiou, and M. Palmer, "The Penn Chinese Treebank: phrase structure annotation of a large corpus," Natural Language Engineering, Cambridge University Press, vol. 11, issue 2, Jun. 2005, pp. 207-238.

[11] X. Duan, J. Zhao, and B. Xu, "Probabilistic models for action-based Chinese dependency parsing," Proc. ECML/ECPPKDD, Warsaw, Poland, Sep. 17-22, 2007, pp. 559-566.

[12] J. Hall, J. Nivre, and J. Nilsson, "Discriminative classifiers for deterministic dependency parsing," Proc. COLING/ACL, Sydney, Australia, Jul. 17-21, 2006, pp. 316-323.

[13] J. Nivre, J. Hall, and J. Nilsson, "MaltParser: a data-driven parser-generator for dependency parsing," Proc. LREC, Genoa, Italy, May 22-28, 2006, pp. 2216-2219.