

# PCFG parsing with CRF tagging for head recognition

Yan Song and Chunyu Kit

Department of Chinese, Translation and Linguistics

City University of Hong Kong

83 Tat Chee Avenue, Kowloon, Hong Kong

{yansong, ctckit}@cityu.edu.hk

## Abstract

This paper presents our work for participation in the 2009 CIPS-Parseval shared task on Chinese syntactic tree parsing, for which we adopt a general PCFG parsing procedure with a conditional random fields (CRF) tagger for head constituent recognition. Our experiments show that an acceptable tagging result is obtained on the basis of a standard PCFG parsing output and a further evaluation on other parsing result also illustrates its effectiveness.

## 1 Introduction

Syntactic tree parsing, also known as full parsing, is a challenge but useful task aiming at analyzing the phrase structure of a sentence and therefore supports many complex applications for natural language processing. A great deal of researches have been conducted on this topic with promising progress (Magerman, 1995);(Collins, 1999);(Charniak, 2000);(Charniak and Johnson, 2005);(Sagae and Lavie, 2006); (Petrov and Klein, 2007);(Finkel et al., 2008);(Huang, 2008). The most widely used framework is the probabilistic context-free grammar (PCFG). Regarding whether the lexical information is attached to syntactic structure, the PCFG parsing may take a lexicalized or unlexicalized approach. With the help of machine learning techniques, both approaches especially the unlexicalized one enjoyed remarkable improvement in recent researches.

It is worth noting that, since parsing is mostly a data driven process, its performance is determined by the amount of data in a treebank on which a parser is trained. Much more data for English than for any other languages have been available so far. Thus most researches on parsing are concentrated on English. It is unrealistic to directly apply any existing parser trained on an English treebank for Chinese sentences. But the methodology is, without doubt, highly applicable. Even for those corpus with special format and information integrated, some modification and enhancement on a well-performed parser to fit the special structure for the data could help to obtain a good performance. In this work, we exploit an exist powerful parser, which has showed its effectiveness on English, with necessary modifications for parsing Chinese for the shared task.

A part of Tsinghua Chinese Treebank (TCT) (Zhou, 2004);(Zhou, 2007);(Chen et al., 2008) is used as our training and testing data for the CIPS-Parseval parsing shared task. The treebank uses a special annotation that not only marks a syntactic constituent with its border, but also labels the head constituents for its child nodes in each syntactic production. Specifically, there can be always multiple heads in a production. Thus our task is to complete both of the two parts for a given segmented and POS-tagged sentence. In order to fit to the special annotation of TCT, we divide the parsing into two major cascade stages, namely PCFG parsing and head constituent recognition, which are connected as a pipeline of processing. For the former, we use a standard PCFG parsing; and for the latter, we apply a sequence tagging method to label head constituents. Head recognition is an important part in this parsing process, since head constituents are the key components to provide complete syntactic and semantic information.

In the next section, we will present the details of our approach. The data and experimental results are presented in Section 3. The last section is the conclusion and future work.

## 2 The approach

As mentioned above, a two-stage cascade routine for parsing is used in this work for the shared task. The two stages, are conducted one by one independently. The PCFG parsing is performed by Berkeley parser (Petrov et al., 2006);(Petrov and Klein, 2007). For the head constituent recognition, according to (Zhou, 2007), the outside and inside syntactic constituents for a single syntactic production can provide enough information to locate the head constituent(s). Thus, to design a proper framework to incorporate the syntactic features could help to label the head constituents correctly. We propose a sequence labeling approach as CRF learning and tagging. The following subsections present the details of our approach.

### 2.1 PCFG parsing

The Berkeley parser is used as our syntactic constituent parsing tool for the first stage. It is a PCFG parser, written in Java, that can be trained on standardized collection of hand parsed sentences. It uses EM training to estimate the probabilities involved in the context-free grammars in use, usually beginning with the barest possible initial structure and then refining the grammars via a hierarchical coarse-to-fine scheme until the predicted syntactic structures fit the training data well enough to a certain degree. Specifically, the parser provides horizontal and vertical markovization<sup>1</sup> modes to support integrating more syntactic information. The parser is originally designed for parsing Penn Treebank (PTB) sentence. Although it provides Chinese grammar package, it is still not capable of dealing with TCT sentences. Thus we need to have a few modifications on it for our task, e.g., add multiple root support based on PTB parsing configuration for TCT sentences since the TCT roots can be *-dj*, *-vp* or *-np* and so on.

### 2.2 Head recognition

To recognize the head constituents(s), an extra step is needed since ordinary parsing could not provide a straight forward way for this. Consider that head constituents are always determined by their syntactic symbol and their neighbors, whose order and relations strongly affects the head labeling. Like chunking (Sha and Pereira, 2003);(Tsuruoka et al., 2009), it is natural to apply a sequence labeling strategy to tackle this problem. We adopt the linear-chain CRF (Lafferty et al., 2001), the most successful sequence labeling framework so far, for the head recognition in this stage. Although in real applications the CRF tagger considers only a fixed context window, while the heads are to be located within a whole production span, it is questionable whether CRF tagging could do the job. We find that most multi-head cases are shown in juxtaposition syntactic productions, which contain repeat constituents, a local context of the neighbors could be enough for the CRF tagger to correctly label the heads just as to incorporate global information. We assume that the sentences in TCT data are always short, thus finding an appropriate window size for learning and tagging could help improve the performance on this task.

We use a binary tag set to determine whether a constituent is a head, e.g. *H* for a head, *O* for a non-head. And we use constituent symbols and the type of the symbols (e.g. whether the symbol is a terminal) in our feature templates, as shown in Table 1, where *c* and *t* represent a constituent symbol and its type respectively, with the subscript for their relative position.

To test the effectiveness of our CRF tagger, we remove all head information from the development set (see Section 3.1), and use the CRF tagger to retrieve the head. The outcome strongly proves its power by showing an accuracy rate of 98.58%.

## 3 Experiments

### 3.1 Data

In TCT, sentences are divided into event descriptive clauses (EDC)<sup>2</sup>, with syntax structure built on them. Our task is to perform full parsing on them. A part of the TCT data is used for CIPS-ParsEval-2009 evaluation. There are 67174 lines in official released training data, with 32771 lines containing syntactic information, which compose our training data for the PCFG parser and the CRF head constituent tagger.

<sup>1</sup>Our experiments show that vertical markovization does not help to improve the parsing performance but causes a huge computation burden.

<sup>2</sup>Actually, an EDC can be considered as a small complete sentence, but with strong connection to context EDCs.

Table 1: Feature templates used in head constituent recognition

Description	Template
Constituent Unigrams	$c_{-2}, c_{-1}, c_0, c_{+1}, c_{+2}$
Constituent Bigrams	$c_{-2}c_{-1}, c_{-1}c_0, c_0c_{+1}, c_{+1}c_{+2}$
Constituent Trigrams	$c_{-3}c_{-2}c_{-1}, c_{-2}c_{-1}c_0, c_{-1}c_0c_{+1}, c_0c_{+1}c_{+2}, c_{+1}c_{+2}c_{+3}$
Type Unigrams	$t_{-1}, t_0, t_{+1}$
Type Bigrams	$t_{-1}t_0, t_0t_{+1}$
Type Trigram	$t_{-1}t_0t_{+1}$
Combined Bigram	$c_0t_0$

Table 2: Official scores for CIPS-Parseval parsing task

	Score
Without-head match F1	0.8364
Partial-head match F1	0.7914
Complete-head match F1	0.7600*

\* Our calculation using the official score program gives 0.7723 for complete-head match F1, with 0.7737 for precision and 0.7708 for recall rate.

We split one fourth of the data from the part with syntactic information as our development set. On the other hand, there are 16210 clauses in released testing data. We only focus on those lines of two or more words. 7939 clauses are then extracted as the final parsing set. Because of the integrated annotation, we need to remove all of the head information from the training sentences in order to properly train the Berkeley parser. Once a test sentence is parsed, each production on the whole syntax tree is analyzed and then the tagged head position is attached to the parent constituent symbol.

### 3.2 Results

There is only one submission for our parsing task, whose official scores are shown in Table 2. The F1 score is combined via geometry average of precision and recall rate of accurate syntactic constituents. It shows the effectiveness of our head constituent recognition.

In further experiments, we improve the output of the PCFG parser by adjusting some parameters for the training process in order to avoid overfitting, which was found after our submission, and adding the 1st order horizontal markovization to incorporate more information. Consequently, the head recognition get a promising improvement, as shown in Table 3.

In addition, since the head constituent recognition works well as long as there is a parsed syntax tree for a given sentence, we could compare our tagging performance objectively against the parsing outputs by other methods. For an official submitted result, generated by a shift-reduce approach, provided by another CIPS-Parseval participant, we work on it just as we did on our development data, by removing

Table 3: Improved performance on CIPS-Parseval parsing task

	Score	Increment
Without-head match F1	0.8582	+2.60%
Complete-head match F1	0.7937	+2.77%*

\* It is the increment from our recalculated complete-head match F1 score, not the officially released one.

Table 4: Complete-head match results on different parsing output.

	Precision	Recall	F1 score
Original score	0.8111	0.8120	0.8115
Retagged score	0.8190	0.8198	0.8194
Improvement	+0.97%	+0.96%	+0.97%

the original head information and retagging by our CRF tagger. The result with increment are shown in Table 4. Nearly one percent of the improvement could be achieved by our CRF tagger.

## 4 Conclusion and future work

In this paper, we present our approach for the parsing subtask in CIPS-Parseval 2009 shared task. We use a two-stage pipeline to tackle the parsing and head constituent recognition. Berkeley parser is used to generate a syntax tree for a given segmented and POS-tagged sentence and a CRF tagger is used to recognize head constituents within each syntactic production in a syntactic tree. The output of the second stage is combined with that from the first stage and then submitted as our final results. The official evaluation scores indicate that our head constituent recognition performs well on different parsing results. It is also shown in our experiments that the head recognition achieves improvement than the parsing.

For future research, the key problem is that the head constituent recognition should not be just a cascade stage of the whole parsing process. Otherwise, its performance would be passively determined by the output of the previous stage, for it is unable to fix any wrong parsing result in its input. To perform a joint decoding rather than a pipeline may help improve the performance.

## Acknowledgments

We thank Mr. Wenbin Jiang for providing his official parsing results for our head constituent tagging experiments, and Dr. Hai Zhao for helpful discussion. The research described in this paper was supported by City University of Hong Kong through the Strategic Research Grants (SRG) 7002267 and 7002388.

## References

- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML 2001*, pages 282-289, Williams College, Williamstown, MA, USA.
- David M. Magerman. 1995. Statistical decision-tree models for parsing. In *Proceedings of ACL 1995*, pages 276-283, MIT, Cambridge, Massachusetts, USA.
- Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis. University of Pennsylvania.
- Eugene Charniak. 2000. A maximum-entropyinspired parser. In *Proceedings of NAACL 2000*, pages 132-139, Seattle, Washington, USA.
- Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of ACL 2005*, pages 173-180, University of Michigan, Michigan, USA.
- Kenji Sagae and Alon Lavie. 2006. A best-first probabilistic shift-reduce parser. In *Proceedings of COLING-ACL 2006*, pages 691-698, Sydney, Australia.
- Jenny Rose Finkel, Alex Kleeman, and Christopher D. Manning. 2008. Efficient, feature-based, conditional random field parsing. In *Proceedings of ACL-HLT 2008*, pages 959-967, The Ohio State University, Columbus, Ohio, USA.
- Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL-HLT 2008*, pages 586-594, The Ohio State University, Columbus, Ohio, USA.

- Slav Petrov, Leon Barrett, Romain Thibaux and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of COLING-ACL 2006*, pages 433-440, Sydney, Australia.
- Slav Petrov and Dan Klein. 2007. Improved Inference for Unlexicalized Parsing. In *Proceedings of HLT-NAACL 2007*, pages 404-411, Rochester, New York, USA.
- Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL 2003*, pages 213-220, Edmonton, Canada.
- Yoshimasa Tsuruoka, Junichi Tsujii and Sophia Ananiadou. 2009. Fast Full Parsing by Linear-Chain Conditional Random Fields. In *Proceedings of EACL 2009*, pages 790-798, Athens, Greece.
- Qiang Zhou. 2007. Base Chunk Scheme for the Chinese Language. *Journal of Chinese Information Processing*, vol 21(3), pages 21-27.
- Qiang Zhou. 2004. Annotation Scheme for Chinese Treebank. *Journal of Chinese Information Processing*, vol 18(4), pages 1-8.
- Yi Chen, Qiang Zhou and Hang Yu. 2008. Analysis of the Hierarchical Chinese Functional Chunk Bank. *Journal of Chinese Information Processing*, vol 22(3), pages 24-31.