Getting the Deep Parse of Chinese

Kun Yu[1]　Yusuke Miyao[2]　Takuya Matsuzaki[1]　Xiangli Wang[1]　Junichi Tsujii[1]

1. The University of Tokyo　2. National Institute of Informatics

{kunyu, xiangli, matuzaki, tsujii}@is.s.u-tokyo.ac.jp　yusuke@nii.ac.jp

Abstract

This paper introduces our recent work on Chinese deep parsing based on the Head-driven Phrase Structure Grammar (HPSG). We first built a Chinese HPSG Treebank semi-automatically from the Penn Chinese Treebank. Then a Chinese HPSG parser was trained on this treebank. This parser can get both the semantic analysis and syntactic analysis simultaneously. Experimental results showed that the proposed parser achieved comparable performance on both semantic parsing and syntactic parsing with previous works.

1 Introduction

Since deep parsing offers rich information, for example semantic roles and long-distance dependency, it has become more and more important in lots of natural language processing applications, such as statistic machine translation, information extraction, and question answering.

In order to fulfill deep parsing, some researchers paid attention to perform semantic role labeling (Marquez et al., 2009) after syntactic parsing. Another alternative is doing deep parsing based on lexicalized grammar theories, such as Head-driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994), Lexical Functional Grammar (LFG) (Dalrymple et al., 1995), Combinatory Categorial Grammar (CCG) (Steedman, 2000), and Lexicalized Tree Adjoining Grammar (LTAG) (Donovan et al.,). Many researches have been done successfully in this way, such as the case in parsing English with HPSG (Miyao and Tsujii, 2008; Matsuzaki et al., 2007), CCG (Clark and Curran, 2004), and LFG (Kaplan et al., 2004).

To our current knowledge, the only previous work about Chinese deep parsing based on the lexicalized grammar theories was done by Guo et al. (2007). In this work, they converted the Penn Chinese Treebank (Xue et al., 2005) into LFG approximations through annotation rules. However, they did not train a deep parser based on the obtained LFG resources, but relied on an external PCFG parser to create c-structure trees, and then mapped the c-structure trees into f-structures using their annotation rules (Guo et al., 2007).

In this paper, we introduce our recent work on developing a Chinese deep parser based on lexicalized grammars. Since HPSG is a lexicalized grammar that integrates both syntactic phrase structure and semantic structures, we choose HPSG as our basic grammar theory. The proposed Chinese HPSG parser was trained and evaluated on a Chinese HPSG Treebank, which was built from the Penn Chinese Treebank 6.0. Experimental results showed that our proposed parser achieved comparable performance on both semantic parsing and syntactic parsing with previous works.

2 Developing a Chinese HPSG Parser

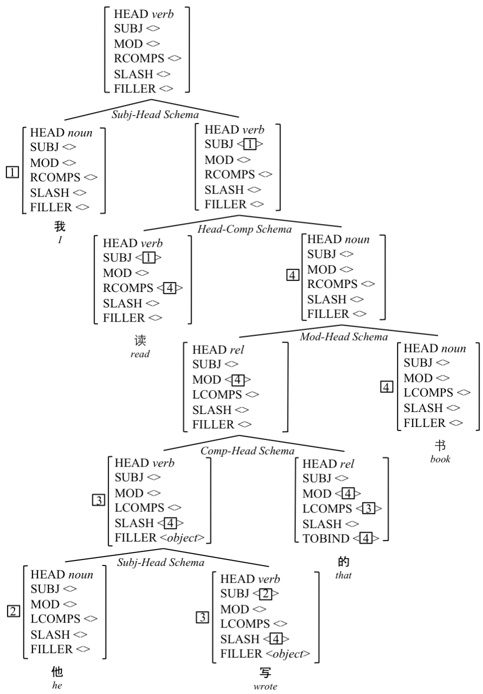
To develop a Chinese HPSG parser, two resources are necessary: a Chinese HPSG Treebank, and a parsing disambiguation model.

2.1 Building a Chinese HPSG Treebank

Yu et al. (2010) proposed creating a Chinese HPSG Treebank with 3 steps: *(1) define a skeleton of the grammar*, *(2) convert an existing treebank (e.g. the Penn Chinese Treebank) into an HPSG-style treebank*, *(3) automatically extract a large-scale lexicon from the obtained HPSG treebank*. Following this approach, we built a Chinese HPSG Treebank from the Penn Chinese Treebank semi-automatically.

This treebank is based on the Chinese HPSG grammar designed in (Yu et al., 2010). From the syntactic point-of-view, besides of keeping the phrase structure of the Penn Chinese Treebank, this

HPSG treebank records the syntactic dependency relations, which are transformed with head rules that are similar as the head rules provided by Yuan Ding[1]. As for semantics, this treebbank uses predicate-argument dependency for semantic representation. 51 types of predicate-argument relations are defined to represent the semantic structures of 13 classes of words. For example, '*verb_arg12*' is a semantic relation defined for transitive verbs, which means a verb takes two argument '*ARG1*' and '*ARG2*'. Figure 1 shows a reduced tree in this HPSG treebank. An HPSG lexicon extracted from this treebank is shown in Figure 2. For the details about how to construct this treebank, please refer to (Yu et al., 2010).



(I read the book that he wrote.)

Figure 1: An HPSG tree in the Chinese HPSG Treebank

2.2 Training a Chinese HPSG Parser on the Treebank

After building a Chinese HPSG Treebank, we used this treebank to train a Chinese HPSG parser.

We used the feature forest model proposed by Miyao and Junichi (2008) as our parsing disambiguation model. This model is a maximum entropy model defined over feature forests. It provides a solution to the problem of probabilistic modeling of complex data structures. This model has been successfully applied to an English HPSG parser – Enju, and achieved good performance (Miyao and Junichi, 2008).

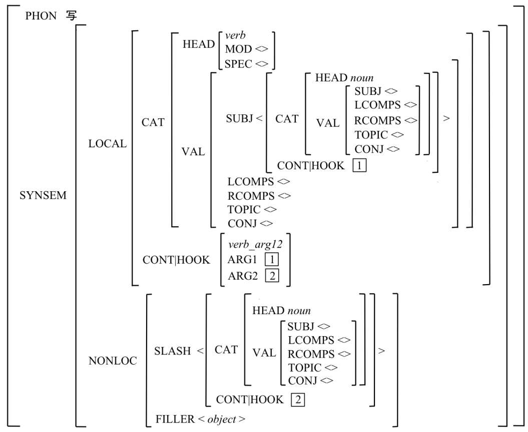


Figure 2: An HPSG lexicon extracted from Figure 1

However, one of the difficulties of doing parsing based on lexicalized grammars is the inefficiency of parsing, due to the complicated data structures used in the lexicalized grammars. Therefore, we applied a supertagging model proposed by Matsuzaki et al. (2007) to reduce the search space explored by the parser and furthermore increase the parsing efficiency.

In short, our HPSG parser works like follows: (1) the supertagging model offers the best maybe-parsable supertag (i.e. lexical template) sequence to the parser; (2) the feature forest model uses this supertag sequence to get an HPSG parser tree; (3) if a well-formed HPSG parse tree can be obtained, the parsing procedure stops; if not, continues (1) and (2), until a well-formed HPSG parse tree can be created.

3 Evaluation Results

3.1 Experimental Setting

We used the Chinese HPSG Treebank converted from the Penn Chinese Treebank 6.0 to evaluate our proposed HPSG parser. We split the corpus into development, testing, and training data sets, following the recommendation from the corpus author. The gold-standard word boundaries and POS tags are applied in all the experiments.

In order to evaluate the performance on semantic parsing, we evaluated the accuracy of the predicate-argument dependencies created by the parser.

[1] http://w3.msi.vxu.se/~nivre/research/chn_headrules.txt

A predicate-argument dependency is defined as <w_p, w_a, r, l>. Here, w_p is the head word of the predicate and w_a is the head word of the argument. r is the type of predicate-argument relation between w_p and w_a. l is the argment label. For example, the sentence '他/he 写/writes 书/books (He writes books)' has two predicate-argument dependencies:

<'写(writes)', '他(he)', *verb_arg12*, *ARG1*>
<'写(writes)', '书(books)', *verb_arg12*, *ARG2*>

We chose the 6 evaluation metrics used by Miyao and Tsujii (2008) for this evaluation. *LP* and *LR* mean the labeled precision and recall of the predicate-argument dependencies. *UP* and *UR* mean the unlabeled precision and recall, regardeless of r and l. *Sem.F1* is the semantic F1-score calculated based on *LP* and *LR*. *Sentence acc.* is the accuracy of the sentences with completely correct predicate-argument dependencies.

Moreover, we evaluated the performance of the proposed parser on syntactic dependency parsing. The evaluation metrics we used are the common metrics used in CoNLL-2007 shared task (Nivre et al., 2007 (b)), which are labeled attachment score (*LAS*) and unlabeled attachment score (*UAS*). In addition, we evaluate the complete sentence accuracy (*COMP*) with labeled dependencies.

3.2 Results on Semantic Parsing

The experimental results on both development and testing data are listed in Table 1. These results indicated that our parser achieved good performance on semantic parsing, which were 77.91% *Sem.F-score* on development data and 77.28% *Sem.F-score* on testing data.

Data	*LP*	*LR*	*UP*	*UR*
Dev	78.19%	77.64%	82.68%	82.10%
Test	76.96%	77.61%	81.27%	81.96%
Data	*Sem.F1*		*Sentence acc.*	
Dev	77.91%		23.93%	
Test	77.28%		22.80%	

Table 1: Accuracy of predicate-argument dependencies

Since there was no benchmark data for Chinese deep parsing, it is difficult to compare the proposed parser with the previous works. In the CoNLL-2009 Shared Task, they did a similar work as our parser, which was joint syntactic dependency parsing and semantic role labeling (Hajic et al., 2009). They merged the Penn Chinese Treebank and the Chinese Proposition Bank (Xue and Palmer, 2009) as training and testing data, and applied a semantic labeled F1-score *(Sem.F1)* to evaluate the performance of semantic role labeling (Hajic et al., 2009). However, different from using the gold-standard POS tags in our parser, this shared task used automatically assigned POS tags.

Table 2 shows the results of the best 3 systems in the closed challenge of Chinese joint task (Hajic et al., 2009). Not like the joint syntactic and semantic parsing done by our parser, all of the 3 systems applied semantic role labeling on the results of state-of-the-art dependency parsers.

System	*Sem.F1*
Nugues	78.60%
Meza-Ruiz	77.73%
Zhao	77.72%

Table 2: Results of the top 3 systems in CoNLL-2009 Shared Task on Chinese

Although we cannot compare our result with the results listed in Table 2 directly because of the different data sets, we can still say that our proposed parser achieved comparable performance to the previous works on semantic parsing.

3.3 Results on Syntactic Parsing

As mentioned in Section 1, our parser is a joint model that can do both syntactic and semantic parsing. Therefore, in order to verify the performance of our parser on syntactic parsing, we compared the labeled and unlabeled dependency relations created by our parser to that from two representative dependency parsers: MaltParser (Nivre et al., 2007 (a)) and MstParser (McDonald et al., 2006).

System	*LAS*	*UAS*	*COMP*
Malt	85.22%	85.62%	34.51%
MST (1^{st} order)	86.68%	87.04%	29.54%
MST (2^{nd} order)	88.26%	88.68%	35.70%
Proposed	**86.70%**	**88.28%**	**34.67%**

Table 3: Accuracy of dependency parsing (dev. data)

System	*LAS*	*UAS*	*COMP*
Malt	85.75%	86.13%	34.42%
MST (1^{st} order)	86.90%	87.37%	31.61%
MST (2^{nd} order)	88.56%	89.03%	37.28%
Proposed	**84.79%**	**86.31%**	**34.53%**

Table 4: Accuracy of dependency parsing (testing data)

We convert the phrase structure of the Chinese HPSG Treebank into dependency structure using the head rules used in Section 2.1. Then, we split the data with the same partition as mentioned in Section 3.1. Table 3 and Table 4 show the comparison results on both development data and testing data. These results indicated that, our parser achieved similar accuracy compared with the MaltParser and the MstParser with 1st order model, but got worse performance than the MstParser with 2nd order model. A possible reason is the short of the second order features in our parsing model. We will consider about enriching these types of features in the future work.

4 Conclusion and Future Work

In this paper, we introduced our work on Chinese deep parsing. For the first time, we proposed a Chinese deep parser based on a lexicalized grammar theory – HPSG. The experimental results on a Chinese HPSG Treebank converted from the Penn Chinese Treebank 6.0 showed that our proposed parser achieved comparable results to previous works on both semantic parsing and syntactic parsing.

There are several future works under consideration, including further improving the design of Chinese HPSG grammar for noun definition and relative clause analysis to relieve the disambiguation burden of parsing, improving the supertagging model, and employing the second-order features in the parsing disambiguation model.

References

Stephen Clark and James R. Curran. 2004. Parsing the WSJ Using CCG and Log-linear Models. Proceedings of ACL 2004.

Mary Dalrymple et al. 1995. Formal Issues in Lexical-Functional Grammar. Cambridge University Press, Stanford, CA.

Ruth O'Donovan et al. 2005. Large-Scale Induction and Evaluation of Lexical Resources from the Penn-II and Penn-III Treebanks. Computational Linguistics: 329 – 366.

Yuqing Guo, Josef van Genabith and Haifeng Wang. 2007. Acquisition of Wide-Coverage, Robust, Probabilistic Lexical-Functional Grammar Resources for Chinese. Proceedings of LFG 2007. 214-232.

Jan Hajic et al. 2009. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. Proceedings of the 13th Conference on Computational Natural language Learning (CoNLL): Shared task. 1-18.

Ronald M. Kaplan et al. 2004. Speed and Accuracy in Shallow and Deep Stochastic Parsing. Proceedings of HLT/NAACL 2004.

Takuya Matsuzaki, Yusuke Miyao, and Junichi Tsujii. 2007. Efficient HPSG Parsing with Supertagging and CFG-filtering. Proceedings of IJCAI 2007.

Lluis Marquez et al. 2009. Semantic Role Labeling: An Introduction to the Special Issue. Computational Linguistics. 34(2): 145-159.

Ryan McDonald, Kevin Lerman and Fernando Pereira. 2006. Multilingual Dependency Analysis with a Two-stage Discriminative Parser. Proceedings of CoNLL-X.

Yusuke Miyao and Junichi Tsujii. 2008. Feature Forest Models for Probabilistic HPSG Parsing. Computational Linguistics. 34(1): 35-80.

Joakim Nivre et al. 2007 (a). MaltParser: A Language-independent System for Data-driven Dependency Parsig. Natural Language Engineering. 13(2): 95-135.

Joakim Nivre et al. 2007 (b). The CoNLL 2007 Shared Task on Dependency Parsing. Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007. 915-932.

Carl Pollard and Ivan A. Sag. 1994. Head-Driven Phrase Structure Grammar. The University of Chicago Press and CSLI Publications, Chicago, IL and Stanford, CA.

Mark Steedman. 2000. The Syntactic Process. The MIT Press.

Nianwen Xue and Martha Palmer. 2009. Adding Semantic Rules to the Chinese Treebank. Natural Language Engineering. 15(1): 143-172.

Nianwen Xue, Fei Xia, Fudong Chiou, and Martha Palmer. 2005. The Penn Chinese Treebank: Phrase Structure Annotation of a Large Corpus. Natural Language Engineering. 11(2): 207-238.

Kun Yu, Yusuke Miyao, Xiangli Wang, Takuya Matsuzaki, Junichi Tsujii. 2010. Semi-automatically Developing Chinese HPSG Grammar from the Penn Chinese Treebank for Deep Parsing. In Proceedings of COLING 2010.