A Description of The CTB-to-Dependency Convertor

Liangchen Luo
Peking University, Beijing
luolc@pku.edu.cn

Abstract

As part of the assignment 2 of course EMNLP, this document contains an overview of the algorithm to convert the Chinese Treebank to a dependency structure in the CoNLL format. The key point of the algorithm is to detect the head word of each phrase. This is done by several hand-crafted rules, called head rules. We describe our head rules in detail and explain how to handle the exceptions to the rules.

1 Algorithm Overview

In the bracketed CTB, each sentence is represented as a phrase tree in string format, which can be easily read and parsed by external toolkits such as NLTK[1]. We can use a pre-order traversal to get the index of each token in the tree, while also clean the constituent tags to ease finding heads.

Then, we find the head word of each phrase and mark them through a depth first traversal based on several hand-crafted rules, which will be described in the next section. Finally we can get the dependency relations and convert them to the CoNLL format.

2 Head Rules

In this section, we describe the rules used to find heads of constituents in the Chinese Treebank[2]; i.e., for a context free grammar $\langle X \rightarrow Y_1, \cdots, Y_n \rangle$ these rules indicate which of $\langle Y_1, \cdots, Y_n \rangle$ is the head of the phrase.

2.1 Hand-crafted Rules

Table 1 shows the head-finding rules for most constituents in the Chinese Treebank. The rules mainly refer to those proposed by Zhang and Clark (2008) and Sun and Jurafsky (2004), with some slight updates. We use the same rules format as described by Zhang and Clark (2008) in Table 1. There are a couple of exceptions to this table, such as INTJ and SKIP, which will be described later.

As an example of how the rules are used, for rules $\langle X \rightarrow Y_1, \cdots, Y_n \rangle$ where X is a VP, the algorithm will first search from the left of the sequence $\langle Y_1, \cdots, Y_n \rangle$ for the first Y_i of type VE; if no VEs are found it will then search for the first Y_i of type VC; if no VCs are found it will search for a VV; and so on. If none of the items on the rule list are found, the left-most child of the rule (Y_1 in this case) will be chosen.

2.2 Handling the exceptions

As we mentioned earlier, there are a couple of exceptions to this table. Although the amount of the exceptions is relatively small, we still need a proper way to handle them.

We use a simple right-most strategy: if a constituent tag is not covered by the head rules, we just select the right-most child of a phrase as the head. According to our observation, this method stands a strong baseline for Chinese language.

References

Honglin Sun and Daniel Jurafsky. 2004. Shallow semantc parsing of chinese. In *HLT-NAACL 2004: Main Proceedings*, pages 249–256, Boston, Massachusetts, USA. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 562–571, Honolulu, Hawaii. Association for Computational Linguistics.

[1] https://www.nltk.org/

[2] http://www.cs.brandeis.edu/~clp/ctb/

Constituent	**Rules**
ADJP	r ADJP JJ AD; r
ADVP	r ADVP AD CS JJ NP PP P VA VV; r
CLP	r CLP M NN NP; r
CP	r CP IP VP; r
DNP	r DEG DNP DEC QP; r
DP	r M; l DP DT OD; l
DVP	r DEV AD VP; r
FRAG	r VV NR NN NT; r
IP	r VP IP NP; r
LCP	r LCP LC; r
LST	r CD NP QP; r
NP	r NP NN IP NR NT; r
NN	r NP NN IP NR NT; r
PP	l P PP; l
PRN	l PU; l
QP	r QP CLP CD; r
UCP	l IP NP VP; l
VCD	l VV VA VE; l
VCP	r VV VA VE; r
VNV	r VV VA VE; r
VP	l VE VC VV VNV VPT VRD VSB VCD VP; l
VPT	l VA VV; l
VRD	l VV VA; l
VSB	r VV VE; r

Table 1: The head-finding rules to extract dependency relations from the Chinese Treebank.