Resolving Coordinate Structures for Chinese Constituent Parsing

Yichu Zhou, Shujian Huang(✉), Xinyu Dai, and Jiajun Chen

State Key Laboratory for Novel Software Technology, Nanjing University,
Nanjing, China
zhouyc@nlp.nju.edu.cn, {huangsj,daixinyu,chenjj}@nju.edu.cn

Abstract. Coordinate structures are linguistic structures consisting of two or more conjuncts, which usually compose into larger constituent as a whole unit. However, the boundary of each conjunct is difficult to identify, which makes it difficult to parse the whole coordinate and larger structures. In labeled data, such as the Penn Chinese Tree Bank (CTB), coordinate structures are not labeled explicitly, which makes solving the problem more complicated. In this paper, we treat resolving coordinate structures as an independent sub-problem of parsing. We first define coordinate structures explicitly and design rules to extract the coordinate structures from labeled CTB data. Then a specifically designed grammar is proposed for automatic parsing of coordinate structures. We propose two groups of new features to better model coordinate structures in a shift-reduce parsing framework. Our approach can achieve a 15% improvement in F-1 score on resolving coordinate structures.

Keywords: Coordinate structure · Grammar · Shift-reduce · Phrase similarity

1 Introduction

Over the past decades, the Chinese constituent parsing task has been rapidly improved. However, there are still several structures that can not be parsed correctly. One of the most difficult structures is the coordinate structure. Kummerfeld et al.[4] showed that the coordinate structures cause 10% of the total errors in Chinese parsing. So, resolving the coordinate structures is critical for improving the performance of Chinese parsing.

In linguistics, a coordinate structure is a complex, frequently occurring type of syntactic structure which links together two or more elements, known as conjuncts or conjoins. Identifying these conjuncts may need high order information from other conjuncts, which may highly increase the complexity of a parsing system.

In this paper, we resolve coordinate structures by separating this task from the overall parsing process to make it an independent sub-task. To investigate the problem independently, there are some problems to be solved. First of all, the commonly used syntactic human labeled data (Penn Chinese Tree Bank[1] (CTB))

[1] http://www.cis.upenn.edu/~chinese/

J. Li et al. (Eds.): NLPCC 2015, LNAI 9362, pp. 353–361, 2015.
DOI: 10.1007/978-3-319-25207-0_30

does not contain explicit label information for coordinate structures. So there is no explicitly labeled data for training. Secondly, coordinate structures have various numbers of conjuncts and may be nested which makes searching and modeling not trivial.

We analyze possible cases of coordinate structures from both the tree bank data and the CTB label guidance and derive three extraction rules to manually convert a CTB style constituent tree to another tree style which can describe coordinate structures (Section 3). We propose to adapt a context free grammar from Hara [3] to describe multi-conjuncts or nested coordinate structures. Then, we propose two groups of features to model the validity of a single conjunct and similarity between conjuncts, respectively. The proposed grammar and features could be easily integrated into a standard shift-reduce parser to perform efficient search (Section 4).

Experiments are conducted on CTB to verify our proposed solutions (Section 5). The results show that our proposed grammar and features could improve the F1 score of coordinate structures by 15%.

2 Related Work

There have been several researches about coordinate structures in English and other languages. Popel et al. [9] discussed the different representations of coordinate structures in different formats and different languages (not including Chinese).

Hara et al. [3] used a grammar to construct a coordination tree and used edit graphs to evaluate the similarity among the possible conjuncts. They did not evaluate the validity of each single conjunct. They used a simple chart parsing algorithm to generate the possible coordination trees, which searches a significant large number of wrong coordinations.

Ogren [8] introduced language model as the main feature into the detection task. Although language model probabilities could give a implicit and rough indication of structural similarity among conjuncts, it is not sufficient to determine a valid coordinate structure.

Maier and Kübler [5] focused on classifying the punctuations as whether it is the separator of a coordinate structure or not. But no experiments are presented about the identification of coordinate structures in the paper.

3 Coordinate Structure in Chinese

CTB does not label coordinate structures in an explicit way, which gives no explicit target for the identification task. Our first step is to extract correct coordinate structures from labeled trees.

According to the definition of coordinate structures from the bracketing guidlines of CTB [10], the coordinate structures are divided into 3 different levels: **Word Level**, **Phrase Level** and **Clause Level**. In these three levels of coordinate structure, the **Clause Level** is much more complicated than the other

Table 1. Notions of symbols

Symbols	Explanation
conjunct	The conjunct of a coordinate structure.
CC	All possible conjunction words, as listed in Xue et al. [10]
ETC	The Chinese word for ETC (” 等”).
PU	Punctuations act as the separator of conjuncts, e.g. ”， ”

two levels and requires sentence level information to resolve. So at present, we do not consider the clause level in this paper.

For the other two levels, we design 3 rules to extract different coordinate structures from the CTB trees. For simplicity, we first define some notions, which are used in the this section (Table 1). The extracting rules are listed as follows:

- **Rule 1**: Extract subtree structures of the following form:
  conjunct {PU conjunct} PU conjunct[ETC]
- **Rule 2**: Extract subtree structures of the following form:
  conjunct {PU conjunct} [PU] CC conjunct [ETC]
- **Rule 3**: Extract subtree structures whose children are leaf nodes with the same POS labels, or node with the POS labels CC or PU.

In these rules, content in [•] can only appear zero or once; content in {•} can appear zero or more times. Rule 2 handles a special case in Chinese which is showed in Figure 1. Examples of other rules are also showed in Figure 1.

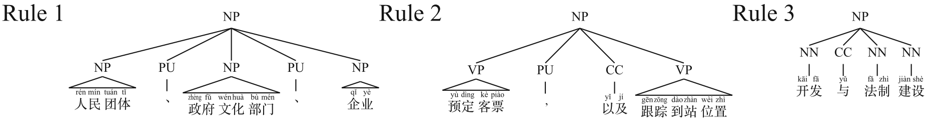

Fig. 1. Examples of 3 different rules

4 Learning to Resolve Coordinate Structures

In this section, we present our methods to resolve coordinate structures. We separate the identification of coordinate structures from the parsing process to be an independent sub-problem. So complex and higher order features could be used in the process.

The general framework is a shift-reduce parsing framework with a perceptron learner [1]. We propose a grammar specifically designed for Chinese coordinate structures (Section 4.1). We propose features to evaluate the validity of a conjunct and the similarity between conjuncts (Section 4.2).

4.1 Grammar of Chinese Coordinate Structures

Hara [3] proposed a grammar for English coordinate structures. We modify it to adjust the specialty of Chinese. We call a parse tree of this grammar a *coordination tree.*

Our grammar, which can cover both nested and flat cases of coordinate structures, is composed of non-terminals (Table 2(a)) and productions (Table 2(b)).

Table 2. Grammar tables

(a) Non-terminals

S	Start symbol
COORD	Complete coordination
COORDX	Partially-built coordination
N	Non-coordination
CJT	Conjunct
CC	conjunction words like ” 和”
W	Any word
ETC	only for ” 等”
SEP	Connector of conjuncts other than CC

(b) Productions

S	⟶	COORD
S	⟶	N
N	⟶	W
N	⟶	COORD N
N	⟶	COORD
N	⟶	W N
CJT	⟶	N
COORD	⟶	COORDX
COORD	⟶	COORDX ETC
CC	⟶	SEP CC
COORDX	⟶	CJT CC/SEP CJT
COORDX	⟶	CJT CC/SEP COORDX

Non-terminals. In Table 2(a). *CC* represents coordination words. *COORD* represents a complete coordinate structure and *COORDX* represents a partially coordinate structure need to be completed. *N* represents all inner nodes except for *COORDX*, *COORD*, *CC* and *CJT*.

Production Rules. In these production rules, the two productions of *COORDX* are the core productions in this grammar. These two productions are used to describe both nested and flat cases of coordinate structures. An example of a *coordination tree* is illustrated in the Figure 2.

4.2 New Features

In this section, we focus on the new features that can model coordinate structures. We split these new features into two different groups: **structural/semantic similarity** and **conjunct validity** . According to our experiment results, only similarity related features between two spans is not strong enough to decide if these two spans should be conjuncted. An important source of errors is the wrong identification of a single conjunct. So, we use the **conjunct validity** to evaluate if the given span is a valid span. A valid span means this span can constitute a syntax node in the CTB tree.

There are two different information sources for the new features. First, inspired by the recent success of the distributed word representation in many NLP tasks, we use word embeddings to describe the semantic similarity and

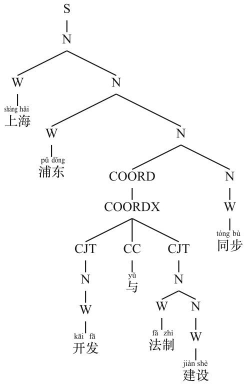

Fig. 2. Example of coordination tree

structural similarity. Another information source is statistical results from CTB data, we use this information to evaluate the validity of conjunct.

We define some notions as follows (w_i represent the i-th word):

- S is a sentence in the form: $w_1 w_2 \cdots w_i \cdots w_j\ CC\ w_k \cdots w_t \cdots w_n$
- $w_i \cdots w_j$ is the first conjunct of S
- $w_k \cdots w_t$ is the second conjunct of S

Similarity. As usual word embeddings, we represent each word as d-dimensional vector $e_i \in R^d$ and use the cosine value of two words to describe the similarity between the two words. For two given spans, we calculate the **semantic similarities** in two perceptive: (i) the average word similarity of the two spans based on different alignments which can be left-most and right-most; (ii) word similarity between the span and context of the two spans. Context of a span is the adjacent words of the span. As an example, the left-most alignment feature is calculated as follows:

$$sim\_align(e_{i,j}, e_{t,k}) = \operatorname*{average}_{\substack{i \leq q \leq j \\ k+q \leq t}}(\cos(e_q, e_{k+q})) \tag{1}$$

where cos is the cosine function to calculate the cosine value of two vectors; *average* is the average function to calculate mean value.

For **structural similarity**, the features are almost the same, except that POS tags are used to replace the words.

Conjunct Validity. Another aspect of information that describes coordinate structures is the validity of a single conjunct. We use statistical probability tables calculated from the CTB tree bank to evaluate this **conjunct validity**. The base idea is calculating the conditional probability from the spans extracted from CTB trees. We use different probability tables which condition on words from the previous, current or next span, respectively. For example, $P_{left\_coherence}(w_{i+1}|w_i)$ represents the conditional probability of w_{i+1} conditioned on w_i which is in the same span of w_{i+1}, while $P_{left\_split}(w_i|w_{i-1})$ represents the conditional probability of w_i conditioned on w_{i-1} which is in the previous spans of w_i.

Complete list of **semantic similarity** and **conjunct validity** features are listed in Table 3. For simplicity, we do not list the **structural similarity** features which use POS tag embeddings instead of word embeddings for calculation.

Table 3. Feature templates

	semantic similarity	conjunct validity

left CJT	sim_align($e_{i,j}$, $e_{k,k+j-i+1}$);
	cos(e_i, e_k);cos(e_i, e_{i-1});
	cos(e_{i-1}, e_i) − cos(e_{i-1}, e_k);
		$P_{left\_coherence}(w_{i+1} \mid w_i)$;
		$P_{left\_split}(w_i \mid w_{i-1})$;
		$P_{left\_boundary}(w_i)$;

right CJT	sim_align($e_{i,t-k+1}$, $e_{k,t}$);
	sim_align($e_{j-(k-t+1),j}, e_{k,t}$);
	cos(e_j, e_t);cos(e_t, e_{t+1});
	cos(e_j, e_{t+1}) − cos(e_t, e_{t+1});
		$P_{right\_coherence}(w_k \mid w_{k-1})$;
		$P_{right\_split}(w_{k+1} \mid w_k)$;
		$P_{right\_boundary}(w_k)$;

COORD	cos($averge(e_{i,j}), averge(e_{t,k})$); $bool(w_{i,j} == w_{t,k})$;	$P_{cond}(w_{i,j} \mid w_{i-1}, w_{t+1})$; $P_{cond}(w_{k,t} \mid w_{i-1}, w_{t+1})$;

5 Experiments

We conduct our experiments on Penn Chinese Tree Bank (CTB 5.1) ^{2} data sets and adapt the same training-test split as described in Zhang and Clark[11]. The embeddings of words and POS tags are trained on data set composed of Chinese gigaword[2] and CTB data using word2vec tools [6]. An in-house implemented Shift-Reduce parser is used as the baseline parser in out experiments.

5.1 Extraction of Coordinate Structures

We apply the three rules (Section 3) to all CTB trees to get well defined coordinate structures, which serve as the learning targets. Among all 18,776 sentences, there are 5,830 sentences which has at least one coordinate structure. The total number of coordinate structures is 8,255, suggesting that a large portion of sentences have multiple coordinate structures. These multiple structures in one sentence may increase the difficulty of resolving coordinate structures. The numbers of coordinate structure with different syntactic tags are showed in the Table 4. As we can see, most of the coordinate structures are under the tag *NP*, *VP* and *QP*. Considering the difficulty of resolving tag *NP* and *VP*[4], resolving the coordinate structures under *NP* and *VP* is much more difficult.

Table 4. Syntax tags distribution of coordinate structures

VCD	ADJP	UCP	ADVP	VP	CLP	PP	DNP	QP	LCP	NP	IP	CP
2	50	5	21	1054	2	13	2	216	10	6880	0	0

Table 5. Experiments results

(a) The different results of sentences with and without coordinate structure

	all test	with coordinate structure	without coordinate structure
Recall	0.7880	0.7849	0.7919
Precision	0.8437	0.8321	0.8585
F-measure	0.8149	0.8078	0.8239

(b) Coordination tree result

	Recall	Precision	F-measure
baseline	0.6339	0.6418	0.6378
new features	0.6717	0.6730	0.6723
basic features	0.6683	0.6708	0.6696
basic + new featrues	0.7775	0.7836	0.7805

5.2 Effects on the Parsing Process

There are no previous experimental results that demonstrate the influence of Chinese coordinate structures[3]. After we get the well defined coordinate structures, we conduct two experiments to check out the real influence of Chinese coordinate structures on the parsing process. Firstly, we separated the CTB test data into two parts, each sentence of the first part has at least one coordinate structure while the sentences of the second part have no coordinate structures at all. Then, we parse and score the two parts separately using the same training data set and the results are showed in the Table 5(a). As we can see, the sentences without coordinate structures get 1.6% higher score in F-measure than the sentences with coordinate structures, which proves coordinate structures have great effect on the overall parsing task.

Secondly, we apply the same extracting rules(Section3) to the output result of our baseline parser to check out how many coordinate structures can be parsed correctly by traditional parsers. This parser achieved 63% recall and 64% precision. This result tells us that the state-of-art parser can only parse about 63% coordinate structures correctly. We use this result as our baseline results in the following experiments.

5.3 Resolving Coordinate Structures

Scoring methods of *coordination tree* is similar to the traditional scoring methods in parsing. But in this *coordination tree*, we only score the spans of the node *COORD* and do not count other spans which is not related to the coordinate structures. We trained *coordinate tree* with a perceptron learner on different

[2] https://catalog.ldc.upenn.edu/LDC2005T01

[3] Ng and Curran[7] has showed the influence of coordinate structures on dependency parsing.

features set. *Base features* means the traditional feature template described in Zhang and Clark[12]. *New features* means the features we discussed in this paper. We also conduct an experiment on the combination of these two feature sets.

As showed in the Table 5(b), using *base features* and *new features* separately can only achieve a little improvement. While when we combine these features, we can achieve 15% improvement in F-measure. This indicates that some coordinate structures can simply be remembered (*base features*) by the model while other coordinate structures need more information (*new features*) to resolve.

6 Conclusion and Future Work

In this paper, we discuss the problem of coordinate structures in Chinese constituent parsing. We separate the problem of identifying the coordinate structure from parsing task. We present how to extract coordinate structures from CTB style trees according to their definitions. Then we presented a framework to solve the identification problem, which includes a specifically designed grammar and newly designed features. Our new features focusing on evaluating coordinate structures include two different groups: **similarity between conjuncts** and **conjunct validity** .

Experiment results show these new features have advantages on modeling coordinate structures. With these features and the grammar, we achieved 15% improvements on detecting coordinate structures.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (Grant No. 61300158, 61170181), the Jiangsu Provincial Research Foundation for Basic Research (Grant No. BK20130580).

References

1. Collins, M.: Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In: Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 1–8. Association for Computational Linguistics (2002)
2. Graff, D., Chen, K.: Chinese gigaword. LDC Catalog No.: LDC2003T09, ISBN 1, 58563–58230 (2005)
3. Hara, K., Shimbo, M., Okuma, H., Matsumoto, Y.: Coordinate structure analysis with global structural constraints and alignment-based local features. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, vol. 2, pp. 967–975. Association for Computational Linguistics (2009)
4. Kummerfeld, J.K., Tse, D., Curran, J.R., Klein, D.: An empirical examination of challenges in chinese parsing. In: ACL (2), pp. 98–103 (2013)
5. Maier, W., Kübler, S.: Are all commas equal? detecting coordination in the penn treebank. In: The Twelfth Workshop on Treebanks and Linguistic Theories (TLT 2012), p. 121 (2013)
6. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013). arXiv preprint arXiv:1301.3781

7. Ng, D., Curran, J.R.: Identifying cascading errors using constraints in dependency parsing
8. Ogren, P.V.: Improving syntactic coordination resolution using language modeling. In: Proceedings of the NAACL HLT 2010 Student Research Workshop, pp. 1–6. Association for Computational Linguistics (2010)
9. Popel, M., Marecek, D., Stepánek, J., Zeman, D., Zabokrtskỳ, Z.: Coordination structures in dependency treebanks. In: ACL (1), pp. 517–527 (2013)
10. Xue, N., Xia, F., Huang, S., Kroch, A.: The bracketing guidelines for the penn chinese treebank (3.0) (2000)
11. Zhang, Y., Clark, S.: A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 562–571. Association for Computational Linguistics (2008)
12. Zhang, Y., Clark, S.: Transition-based parsing of the chinese treebank using a global discriminative model. In: Proceedings of the 11th International Conference on Parsing Technologies, pp. 162–171. Association for Computational Linguistics (2009)