An End-to-End Chinese Discourse Parser with Adaptation to Explicit and Non-explicit Relation Recognition

Xiaomian Kang[1,2], **Haoran Li**[1,2], **Long Zhou**[1,2], **Jiajun Zhang**[1,2], **Chengqing Zong**[1,2,3]
[1]National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences
[2]The University of Chinese Academy of Sciences, Beijing, China
[3]CAS Center for Excellence in Brain Science and Intelligence Technology
{xiaomian.kang,haoran.li,jjzhang,cqzong}@nlpr.ia.ac.cn

Abstract

This paper describes our end-to-end discourse parser in the CoNLL-2016 Shared Task on Chinese Shallow Discourse Parsing. To adapt to the characteristics of Chinese, we implement a uniform framework for both explicit and non-explicit relation parsing. In this framework, we are the first to utilize a seed-expansion approach for the argument extraction subtask. In the official evaluation, our system achieves an F1 score of 26.90% in overall performance on the blind test set.

1 Introduction

Discourse parser analyzes the relations underlying text units to uncover abstractive structure information, which has a wide usage in different tasks in natural language processing, such as text summarization, question answering, information extraction and machine translation.

Since the release of the Penn Discourse TreeBank (PDTB) (Prasad et al., 2008), discourse parsing has drawn more and more attention. The PDTB-style parser puts emphasis on shallow discourse parsing, which annotates a piece of text with a set of discourse relations. The relations are divided into two types, explicit or non-explicit, depending on whether connectives exist or not. A complete discourse relation contains two discourse units called Argument1 (Arg1) and Argument2 (Arg2). An end-to-end parser usually consists of some components, such as discourse connective identification, argument extraction, explicit sense classification and implicit sense classification.

Pitler and Nenkova (2009) used syntactic features to disambiguate explicit discourse connectives. For argument extraction, Lin et al. (2014) used a tree subtraction algorithm to extract arguments and Kong et al. (2014) proposed a constituent-based approach to solve it. Recent researches mainly focus on the implicit sense classification. In this subtask, Lin et al. (2009) and Rutherford and Xue (2014) explored rich features such as word-pairs, dependency rules, production rules and Brown cluster pairs. Some studies (Rutherford and Xue, 2015) paid attention to the data expansion. Neural network approaches (Ji and Eisenstein, 2015; Zhang et al., 2015) were also applied to improve the classification performance. Lin et al. (2014) implemented a full end-to-end PDTB parser and Wang and Lan (2015) built a more refined system in the CoNLL-2015 Shared Task.

In contrast to English, there are limited studies on Chinese discourse parsing (Huang and Chen, 2011; Zong, 2013; Tu et al., 2014). One of the main reasons is the shortage of Chinese discourse corpus. Zhou and Xue (2012) annotated a PDTB-style Chinese Discourse TreeBank (CDTB), which is the data for Chinese shallow discourse parsing.

In this paper, we describe our approaches to implement the Chinese shallow discourse parser which is participated in the CoNLL-2016 Shared Task (Xue et al., 2016). In view of some typical characteristics in CDTB (Section 2), we adopt and extend the state-of-the-art English parser in CoNLL-2015 (Wang and Lan, 2015). A unified framework for both explicit and non-explicit parsing is built and a seed-expansion approach is utilized for argument extraction. Some useful features are selected to train classifiers (Section 3). Our system achieves 40.89% and 26.90% in F1-measure on the test and blind data set respectively (Section 4).

We make the following main contributions in this work:

- We implement a complete end-to-end PDTB-style discourse parser for Chinese.
- We design a uniform framework to recog-

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 27–32,
Berlin, Germany, August 7-12, 2016.

nize both explicit and non-explicit relations together.

- We utilize an effective seed-expansion approach to determine the exact span boundaries in the argument extraction subtask.

2 Corpus and Resources

In addition to the PDTB-style annotation, there are many special phenomena in CDTB. We enumerate several characteristics in (Zhou and Xue, 2015) and phenomena from the training data set.

- In contrast to the 54.53% in PDTB, the proportion of non-explicit relations is 78.27% in CDTB training set. PDTB's three-level sense hierarchy structure is replaced by 11 flat semantic types.

- Discourse connectives are flexible and the phenomenon of parallel connectives is obvious in Chinese. In our experiment, we extract 385 connectives from the training set as a connective dictionary.

- The span of an argument ranges from several words to sentences even to paragraphs. But in general, the span is in one sentence and the clauses split by punctuations can be regarded as the minimum constituent units.

- As shown in (Yang and Xue, 2012), punctuation marks play a significant role in Chinese discourse. Fortunately, CDTB has annotated those punctuations that may indicate discourse relations.

Inspired by the above phenomena, we design our system by fully considering these Chinese characteristics.

Besides the training data, we simply use skip-gram neural word embeddings provided by the CoNLL-2016 organizers to replace words in some features.

3 System Architecture

Zhou and Xue (2015) pointed out that discourse connectives and punctuation marks in Chinese can serve as anchors, which are clues of discourse relations. This opinion encourages us to treat explicit and non-explicit relations similarly. Therefore, the explicit and non-explicit parsers share the same framework shown in figure 1. We divide the shallow discourse parsing into four subtasks: anchor identification, argument extraction, sense classification and argument relabeling

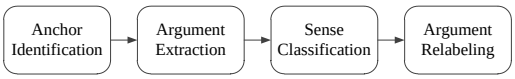

Figure 1: discourse parser framework

Anchor Identification. It is to recognize the anchors from candidates. For explicit parser, the connectives are important relation indicators. And the punctuations play the similar role in non-explicit parser.

Argument Extraction. It is to extract argument pair according to the anchor. We use a seed-expansion approach, transforming this subtask into argument boundary identification.

Sense Classification. It is to predict the type of relation sense between Arg1 and Arg2.

Argument Relabeling. It is to re-label the labels of two arguments. Although Arg1 is in front of Arg2 in most cases, the "Arg1" and "Arg2" labels for the argument pair are defined based on the semantics in CDTB (Zhou and Xue, 2012).

3.1 Anchor Identification

A full text is scanned to pick out the anchor candidate set. Then, a binary classifier is designed to check whether each candidate is anchor or not. The explicit connective candidate set is generated by matching the text with our connective dictionary. The non-explicit punctuation candidate set consists of all punctuations except for quotes, parentheses, and pause marks.

3.1.1 Connective Identification

A classifier is trained to recognize connectives. The features are chosen by referring to the best system in CoNLL-2015 (Wang and Lan, 2015). Zhou and Xue (2012) found that a discourse connective is almost always accompanied by punctuations, which help us to design the features.

The features we used are as follows:

- Lexical features: candidate itself, number of the candidate words, POS of the candidate, POS of the previous word, embeddings of the next three words, the previous word combined with the next word, location of the candidate in the sentence (start, middle, end), the

previous/next punctuation, whether the previous or next character is punctuation.

- Syntactic features: the parent of candidate's node (the lowest node in the syntax tree that completely covers the candidate words), the left and right siblings of candidate's node, the production rules of candidate, the path from the candidate's node to root, whether the left sub-tree or right sub-tree contains VP or IP.

3.1.2 Punctuation Identification

According to their locations in sentences, punctuations are divided into two cases: MOS (middle of sentence) and EOS (end of sentence). In the 56.18% of non-explicit relations in the training data, Arg1 and Arg2 are in the same sentence. The anchor punctuation must be in the middle of the sentence in this case and we extract features from its left and right clauses. In another case that Arg1 and Arg2 is in different sentences, the anchor must be in the end of the sentence and we extract features from its left and right sentences. Since we cannot get the syntactic features from two different syntactic trees, the two classifiers' features are designed respectively.

MOS Punctuation Classification. By referring to (Yang and Xue, 2012; Xu et al., 2012), we extract features from the context clauses:

- Lexical features: embeddings of the first and last word in the context clauses, POS of the first and last word in the context clauses, punctuation itself.

- Syntactic features: the parent, left and right siblings of the punctuation's node, the left and right clause's node, the path from the punctuation's node to the right clause's node, whether the left sub-tree or the right sub-tree contains VP or IP, whether the leftmost sibling of punctuation's parent node is PP, the number of IP in siblings of the punctuation's parent node, whether the right sub-tree contains AD or CS if the leftmost sibling is IP.

EOS Punctuation Classification. We only use lexical features from the context sentences: punctuation itself, embeddings of the first and last three words in the context sentences, POS of the first and last three words in the context sentences.

3.2 Argument Extraction

Our approach is based on the following observations. It should be noted that "Arg1" and "Arg2" are defined by semantics rather than location. But for convenient expression, we temporarily name the front argument as "Arg1" and the following argument as "Arg2" before Argument Relabeling [1].

- ***Observation* 1**: In most cases, Arg1 and Arg2 are in the same sentence or two adjacent sentences respectively.

- ***Observation* 2**: An argument consists of one or several consecutive clauses.

- ***Observation* 3**: Explicit Arg2 is located in the same sentence as its connective anchor.

- ***Observation* 4**: The span of Arg1 and the span of Arg2 are adjacent. There is no clause between them.

The anchor can provide useful location information to determine the span of the argument especially in non-explicit relations. So after considering the special characteristics of argument pairs in CDTB, we utilize a seed-expansion approach to extract Arg1 and Arg2 based on the anchor. According to *Observation* 2, we regard the clauses as the minimum argument units. A seed is a clause which contains or adjoins the anchor. We think the seed must be in the argument and provides a good starting point for argument extraction.

The approach contains three steps: sentence scope determination, seed pair generation and seed expansion. Figure 2 shows the detailed process in explicit argument extraction to vividly explain the approach.

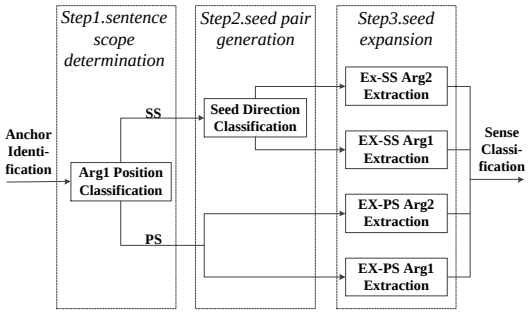

Figure 2: the explicit argument extraction process

[1] In Section 3.3, for convenience, we still temporarily name the "Arg1" and "Arg2" following the sequence order.

First, according to *Observation* 1, we determine the rough sentence-level scope of argument. Then according to *Observation* 4, we obtain a pair of adjacent clauses as the seed pair based on the anchor. Finally, we expand the seed clause-by-clause to obtain the argument pair.

3.2.1 Sentence Scope Determination

We determine the sentence scope of the argument in this step.

Explicit: *Observation* 3 has given the sentence scope of Arg2 in explicit relation. So we discuss the scope of Arg1. We divide the Arg1 into two cases: SS (Arg1 is in the same sentence as its connective anchor) and PS (Arg1 is in the previous sentence of the connective anchor). A classifier is trained to determine which case an Arg1 is.

For the Arg1 Position Classification, the features are about connective: connective itself, POS of the connective, the number of connective words, location of the connective in the sentence, whether the connective is in the first clause of the sentence, previous/next punctuation, the path from connective's parent node to root.

Non-Explicit: In Section 3.1.2, the punctuation anchors have been divided into MOS and EOS, respectively correspond to SS (Arg1 is in the same sentence as Arg2) and PS (Arg1 is in the previous sentence of the Arg2 sentence). So there is no need to take into account the sentence-level scope of non-explicit argument.

3.2.2 Seed Pair Generation

The seed pair is a pair of two adjacent clauses, which must be in Arg1 and Arg2 respectively.

Explicit: For SS case (Arg1 is in the same sentence as the connective), the current clause (clause contains the connective [2]) is a seed. Anther seed is the clause adjacent to the current. But when the current clause is in the middle of sentence, a question comes up: is anther seed the previous or the next clause? The Seed Direction Classification helps us to answer it.

The features are as follows: connective itself, POS of the connective, whether there are co-occurrence of nouns, verbs and quantifiers between current clause and previous/next clause, the parent of previous/next clause's node, the punctuation between previous/next clause and current clause, the relationship of previous and current clause's node (left, right, middle, contain, none).

For PS case (Arg1 is in the previous sentence of the connective), there is no need to judge the seed direction. We directly take the last clause of the previous sentence as the front seed and the clause contains the connective as the following seed.

Non-Explicit: No matter where the location of the punctuation anchor, we treat the nearest left and right clauses of the punctuation as the seed pair.

3.2.3 Seed Expansion

After obtaining the seed pair, we expand the seed to grow into argument. The front seed expands forward and the following seed expands backward. We expand the span of argument clause-by-clause, from the seed clause, toward a fixed direction (forward or backward) in the sentence scope to generate candidate sets. So each candidate contains the seed clause. The current candidate has one clause more than the previous one. The classifiers decide whether the current candidate span is beyond of the argument boundary. We select the longest candidate predicted OK as the argument.

There are four cases totally: explicit SS, explicit PS, non-explicit SS and non-explicit PS. So we train eight classifiers for each case to extract Arg1 and Arg2 respectively. Each classifier uses the same feature template while Arg1 and Arg2 extraction have the opposite expansion direction. The features are as follows, and some are borrowed from (Lin et al., 2014; Wang and Lan, 2015).

- Lexical features are from the previous candidate and the current clause: embeddings of the first/last three words of them, POS of the first/last word of them, punctuations between them, whether there are co-occurrence of nouns/verbs between them, anchor itself.
- Syntactic features: the parent of anchor's node, the current clause's node and its left and right siblings, the current candidate's node and its parent, the path from previous candidate's node/seed clause's node to current clause's node, the relationship of current clause and the seed clause/previous candidate (left, right, middle, contain, none).

[2] If the connective is a parallel connective that spans clause boundaries, we regard these clauses as a whole current clause.

- Others: whether the current clause is the start/end of sentence, the relative length of current clause and seed clause/previous candidate (short, middle, long).

Through the above method, we can get the clause boundary of the argument pair. Finally, the post-processing is done: connectives and punctuations appear at the start or end of span are deleted.

3.3 Sense Classification

The sense of relation is decided after the anchor identification and argument extraction by a multi-class classifier.

Explicit: The huge contribution of discourse connectives to the explicit sense classification makes it possible that a small amount of features about connectives will produce good enough results.

- Lexical features: connective itself, POS of the connective, embedding of the connective, the previous and next punctuation of the connective.
- Syntactic features: the parent, left and right siblings of connective's node, the Arg1's node and Arg2's node, the parent of Arg1's node and Arg2's node, the relationship of Arg1's node and Arg2's node.

Non-Explicit: In this work, we decided to only use the production rules of Arg1, Arg2 and co-occurrence after trying other features in our experiments. We choose from all the production rules whose frequency is over 5 and finally select the 100 ones by calculating the information gain.

3.4 Argument Relabeling

This component is to re-label the argument labels. The features are listed as follows:

- Lexical features: anchor itself, POS of previous and next word of the anchor, location of the anchor in the sentence, whether there are co-occurrence of nouns, verbs and quantifiers between Arg1 and Arg2.
- Syntactic features are the same as the syntactic features in explicit sense classification (Section 3.3).
- Others: the relative length of Arg1 and Arg2, the relation sense.

4 Experiments and Results

Our end-to-end parser consists of 4 subtasks and 17 classifiers, trained on the corpora provided in the CoNLL-2016 Shared Task. All of the models are trained using the maximum entropy algorithm implemented in MALLET toolkit [3]. The system was evaluated on the TIRA evaluation platform (Potthast et al., 2014) on 3 data sets offered by CoNLL-2016: development set, test set and blind test set. Table 1 reported the official results of our parser.

	Task	Dev	Test	Blind
Explicit	*Conn*	0.8356	0.7263	0.5627
	Arg1	0.5479	0.5587	0.3853
	Arg2	0.6849	0.6816	0.4444
	Both	0.4521	0.4916	0.2650
	Sense	0.7534	0.6480	0.4811
	Parser	0.4521	0.4859	0.2446
Non-Explicit	*Conn*	–	–	–
	Arg1	0.6282	0.6266	0.5526
	Arg2	0.6798	0.6762	0.6017
	Both	0.5341	0.5379	0.4457
	Sense	0.5068	0.4987	0.4082
	Parser	0.3982	0.3869	0.2712
All	*Conn*	0.8356	0.7263	0.5627
	Arg1	0.6261	0.6328	0.5439
	Arg2	0.6932	0.6921	0.5843
	Both	0.5317	0.5418	0.4178
	Sense	0.5640	0.5333	0.4326
	Parser	**0.4120**	**0.4089**	**0.2690**

Table 1: The official subtasks and overall F1-measures of the parser on the development, test and blind test sets for explicit, non-explicit and all relations.

We provide some analysis from the results:

- More than 20% sharp decrease of F1 in explicit parser on the blind set is mainly due to the error propagation of connective identification. The error is mainly from two aspects. One is the flexible parallel connectives. Another is the ambiguous definition of connectives, especially in the middle of the sentence.
- The seed-expansion method can get acceptable results for argument extraction. It is hard to determine whether a clause is in the span of argument when it plays a role of supplement or conjunction to the basic semantic.

[3] http://mallet.cs.umass.edu/

This causes the main error. So more features about the span cohesion should be tried in future. The F1 of Arg1 and Arg2 individually is about 10% higher than jointly. Besides, the assumption that the span of argument is in one sentence is too strong.

5 Conclusion

We have built a PDTB-style end-to-end Chinese shallow discourse parser for the CoNLL-2016 Shared Task. Our system is adapted to the Chinese characteristics. A seed-expansion approach is proposed to extract the arguments correctly. On the official blind test set, we achieve the 26.90% in F1-measure.

Acknowledgments

The research work has been partially funded by the Natural Science Foundation of China under Grant No. 61333018 and No. 61303181 and supported by the Strategic Priority Research Program of the CAS (Grant XDB02070007).

References

Hen-Hsen Huang and Hsin-Hsi Chen. 2011. Chinese discourse relation recognition. In *IJCNLP*, pages 1442–1446.

Yangfeng Ji and Jacob Eisenstein. 2015. One vector is not enough: Entity-augmented distributed semantics for discourse relations. *Transactions of the Association for Computational Linguistics*, 3:329–344.

Fang Kong, Hwee Tou Ng, and Guodong Zhou. 2014. A constituent-based approach to argument labeling with joint inference in discourse parsing. In *EMNLP*, pages 68–77.

Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the Penn Discourse Treebank. In *Proceedings of the EMNLP 2009*, pages 343–351.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20(02):151–184.

Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 13–16.

Martin Potthast, Tim Gollub, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. 2014. Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In Evangelos Kanoulas, Mihai Lupu, Paul Clough, Mark Sanderson, Mark Hall, Allan Hanbury, and Elaine Toms, editors, *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14)*, pages 268–299, Berlin Heidelberg New York, September. Springer.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The Penn Discourse TreeBank 2.0. In *LREC*. Citeseer.

Attapol Rutherford and Nianwen Xue. 2014. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *EACL*, volume 645, page 2014.

Attapol Rutherford and Nianwen Xue. 2015. Improving the inference of implicit discourse relations via classifying explicit discourse connectives. In *Proceedings of the NAACL-HLT*.

Mei Tu, Yu Zhou, and Chengqing Zong. 2014. Automatically parsing Chinese discourse based on maximum entropy. *Acta Scientiarum Naturalium Universitatis Pekinensis*, 50(1):125–132.

Jianxiang Wang and Man Lan. 2015. A refined end-to-end discourse parser. *CoNLL 2015*, page 17.

Shengqin Xu, Fang Kong, Peifeng Li, and Qiaoming Zhu. 2012. A Chinese sentence segmentation approach based on comma. In *Chinese Lexical Semantics*, pages 809–817. Springer.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Bonnie Webber, Attapol Rutherford, Chuan Wang, and Hongmin Wang. 2016. The CoNLL-2016 Shared Task on multilingual shallow discourse parsing. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning - Shared Task*, Berlin, Germany, August. Association for Computational Linguistics.

Yaqin Yang and Nianwen Xue. 2012. Chinese comma disambiguation for discourse analysis. In *Proceedings of the ACL 2012*, pages 786–794.

Biao Zhang, Jinsong Su, Deyi Xiong, Yaojie Lu, Hong Duan, and Junfeng Yao. 2015. Shallow convolutional neural network for implicit discourse relation recognition. In *Proceedings of the EMNLP 2015*, pages 2230–2235.

Yuping Zhou and Nianwen Xue. 2012. PDTB-style discourse annotation of Chinese text. In *Proceedings of the ACL 2012: Long Papers-Volume 1*, pages 69–77.

Yuping Zhou and Nianwen Xue. 2015. The Chinese Discourse TreeBank: A Chinese corpus annotated with discourse relations. *Language Resources and Evaluation*, 49(2):397–431.

Chengqing Zong. 2013. *Statistical Natural Language Processing*. Tsinghua University Press.