Shallow Discourse Parsing Using Constituent Parsing Tree*

Changge Chen[1,2], **Peilu Wang**[1,2], **Hai Zhao**[1,2†]
[1]Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
[2]Key Laboratory of Shanghai Education Commission
for Intelligent Interaction and Cognitive Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
{changge.chen.cc,plwang1990}@gmail.com, zhaohai@cs.sjtu.edu.cn

Abstract

This paper describes our system in the closed track of the shared task of CoNLL-2015. We formulize the discourse parsing work into a series of classification subtasks. The official evaluation shows that the proposed framework can give competitive results and we give a few discussions over latent improvement as well.

1 System Overview

We design our shallow discourse parser as a sequential pipeline to mimic the annotation procedure as the Penn Discourse Treebank (we will use PDTB instead in the rest of this paper) annotator (Lin et al., 2014). Figure 1 gives the pipeline of the system. The system can be roughly split into two parts: the explicit and the non-explicit. The first part consists of three steps, which sequentially are Explicit Classifier, Explicit Argument Labeler, and Explicit Sense Classifier. While the non-explicit part consists of Filter, Non-explicit and Non-explicit Sense Classifiers. Non-explicit relations include *'Implicit'*, *'AltLex'*, *'EntRel'*, but not *'NoRel'*.

We adopt an adapted maximum entropy model as the classification algorithm for every steps. Our system only exploits resources provided by the organizer.

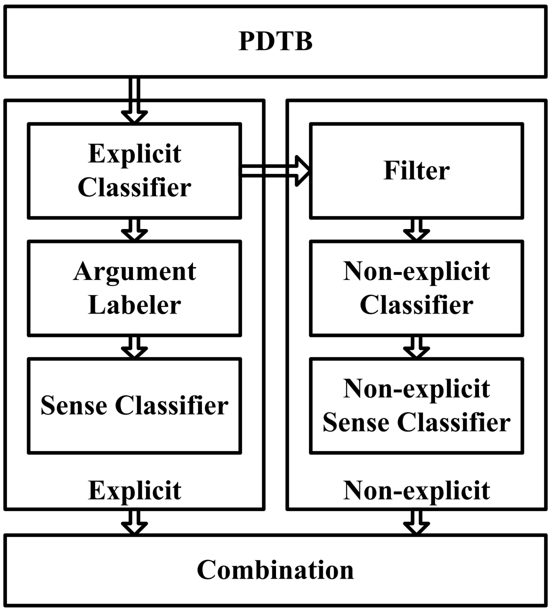

Figure 1: Pipeline of the system

Explicit connectives in train set	14722
Level-order Scan	13911

Table 1: Performance of level-order scan

We first give a brief introduction over each step of the entire system as the following. After the Explicit Classifier detects explicit connectives, the Explicit Argument Labeler then prunes and classifies the *'Arg1'* and *'Arg2'* of the detected connective. Then, Explicit Sense Classifier integrates results of previous two steps when trying to distinguish different senses. The second part of the system starts with filtering out obvious false cases. Then the Non-explicit classifier classifies the non-relations into three classes, i.e., *'Implicit'*, *'AltLex'*, *'EntRel'*. Finally, the Non-explicit Sense classifier determines the sense of the non-explicit relation. In the last two steps, we take the *'EntRel'* as a sense of implicit relation, which we will explain later.

*This work of C. Chen, P. Wang, and H. Zhao was supported in part by the National Natural Science Foundation of China under Grants 60903119, 61170114, and 61272248, the National Basic Research Program of China under Grant 2013CB329401, the Science and Technology Commission of Shanghai Municipality under Grant 13511500200, the European Union Seventh Framework Program under Grant 247619, the Cai Yuanpei Program (CSC fund 201304490199, 201304490171, and the art and science interdiscipline funds of Shanghai Jiao Tong University (a study on mobilization mechanism and alerting threshold setting for online community, and media image and psychology evaluation: a computational intelligence approach under Grant 14X190040031(14JCRZ04).

†Corresponding author

Proceedings of the Nineteenth Conference on Computational Natural Language Learning: Shared Task, pages 37–41,
Beijing, China, July 26-31, 2015. ©2014 Association for Computational Linguistics

Tag	P	R	F
Arg1	0.7748	0.7544	0.7645
Arg2	0.8102	0.9658	0.8812
NULL	0.9922	0.9781	0.9851

Table 2: Performance of Argument Labler

2 System Modules

2.1 Explicit Part

In this part, our parser extracts the explicit relations. An explicit example is given below.

He added that "having just one firm do this isn't going to mean a hill of beans. But **if this prompts others to consider the same thing, then it may become much more important**.

'Arg1' is shown in italic, and *'Arg2'* is shown in bold. The discourse connective is underlined and the sense of this explicit relation is *'Comparison.Concession'*.

2.1.1 Explicit Classifier

There are 100 explicit connectives in PDTB annotation (Prasad et al., 2008). However, some connectives, e.g., *'and'*, do not express a discourse relation. We use a level-order traverse to scan every node in the constituent parse tree to select the connective candidates. This method gives us a high recall in the train set as shown in Table 1.

Seven features are considered (Pitler and Nenkova, 2009):

a) Self Category The highest dominated node which covers the connective.

b) Parent Caterogy The category of the parent of the self category.

c) Left Sibling Category The syntactic category of immediate left sibling of the self-category. It would be *'NONE'* if the connective is the leftmost node.

d) Right Sibling Category The immediate right sibling of the self category. It also would be assigned *'NONE'* if the self-category has been the rightmost node.

e) VP Existence We set a binary feature to indicate whether the right sibling contains a VP.

f) Connective In addition to those features proposed by Pilter and Nenvoda, we introduce connective feature. The potential connective itself would be a strong sign of its function. A few of discourse connectives that are deterministic. For example, *'in addition'* will always be *'Expansion.Conjunction'*.

Maximum Entropy classifier has shown good performance in various previous works (Wang et al., 2014; Jia et al., 2013; Zhao and Kit, 2008). Based on these features, we trained a Maximum Entropy classifier. In order to check the performance of the classifier only, we evaluate the classifier on connective candidates that selected by a level-order traverse. This gives 93.87% accuracy and 90.1% F1 score on dev set.

2.1.2 Explicit Argument Labeler

With all explicit connectives detected, we exploit a constituent-based approach to perform argument labeling (Kong et al., 2014). Along the path from the connective node to the root node in the constituent parse tree, all the siblings of every node on the path are selected as candidates for *'Arg1'* and *'Arg2'*. For these candidates, we compare them with PDTB to label them as *'Arg1'*, *'Arg2'*, or *'NULL'*. However, this argument prune strategy focuses on intra sentence. In addition, Kong et al. unified the intra- and inter-sentence cases by treating the immediate preceding sentence as a special constituent. Based on our empirical results, the inter-sentences only contribute to the augment candidate Arg1. Kong et al. also reported a very high recalls (80-90%) on *'Arg1'* and *'Arg2'* extraction, though our re-implementation only receive recalls 37.5% and 51.3% of the *'Arg1'* and *'Arg2'*, respectively. And about 87.75% of all the pruning out constituents are labeled as *'NULL'*. Similar to treating the immediate preceding sentence as *'Arg1'* candidate, we take the remaining part of the sentence that is adjacent to the connective as *'Arg2'* candidate. This approach gives a boost in *'Arg2'* recall, as high as 93.1%.

We extract features from constituent parser tree (Zhao and Kit, 2008; Zhao et al., 2009). The extracted features can be divided into two parts. The first part captures information about the connective itself:

a) Con-str Case-sensitive string of the given connective.

b) Con-Lstr The lowercase string of the connective.

c) Con-iLSib Number of left sibling of the connective.

d) Con-iRSib Number of right sibling of the connective.

The second part consists of features from the

syntactic constituent:
e) NT-Ctx Context of the constituent. We use POS combination of the constituent, its parent, left sibling and right sibling to represent the context.
f) Con-NT-Path The path from the parent of the connective to the node of the constituent.
g) Con-NT-Position The positive of the constituent relative to the connective: left, right, or previous.

After the parser categories all the candidates constituent into *'Arg1'*, *'Arg2'*, and *'NULL'*, Kong et al. adopted a Linear Integer Programming to impose constraints that the number of *'Arg1'* and **'Arg2'** should no less than one, The extracted arguments should not overlap with the connective. Our experiments also show that some constraints are useless. For example, constraint that the pruned out candidates should not overlap with the connective. The pruning algorithm considers the siblings of the node along the path, there is no chance that the pruned out candidate would overlap with the connective node.

Without considering the error propagated by the pruning process, the argument labeler gives results as Table 2.

2.1.3 Explicit Sense Classifier

In this part we only take a naive approach that take the most frequent sense of the detected explicit connective. A better approach needs to build a sense classifier with syntactic features of the connective such as POS, and position and length of arguments.

2.2 Non-Explicit Part

This part is based on the result of explicit part. We assume that Explicit and Non-explicit relations cannot exist in the same sentence simultaneously. So we take out sentences which have been labeled as Explicit in the first part. Then, we take all the adjacent sentences left in the article as candidate implicit relations. There are 13,155 implicit relations given in the train set.

2.2.1 Filter

Apart form filtering out the explicit connective, we also discard sentences between two paragraphs. After these two filtering steps we get 8,728 non-explicit relations.

2.2.2 Non-explicit Classifier

At first glance, we should build a classifier that can distinguish the relations *'Implicit'*, *'AltLex'*, and *'EntRel'*. We give the distribution of each relations in the train set in Table 3.

	Implicit	AltLex	EntRel
#	13,155	524	4,133
%	73.85	2.94	23.2

Table 3: Distribution of Non-Explicit Relations in train set

Sense	#	%
*Ent Rel	4,133	23.2
*Expn..Conj.	3,321	18.64
*Expn..Rest.	2,543	14.28
*Cont. Cause. Reason	2,135	11.99
*Comp..Cont.	1,646	9.24
*Cont..Cause.Result	1,519	8.53
* Expn..Inst.	1,165	6.54
*Temp..Asyn..Prec.	460	2.58
*Comp..Conc.	197	1.11
*Temp..Sync.	169	0.95
Comparison	145	0.81
*Temp..Asyn..Suc.	143	0.8
*Expn..Alt..Chosen alt.	142	0.8
Expn.	75	0.42
*Expn.Alt.	11	0.06
*Cont.Cond.	4	0.02
*Expn..Exc.	2	0.01
Cont..Cause	1	0.00
Temp.	1	0.00
Cont.	1	0.00

Table 4: Distribution of Non-explicit Senses in train set.*

We can see the *'AltLex'* only covers about 2.94%, which is relatively negligible comparing with *'Implicit'*(73.85%) and *'EntRel'*(23.2%). So we decide to focus only on the latter two relations, and the classifier only works on these two relations. Instead of building a single classifier, we set all the non-explicit relations as *'Implicit'* here, and view *'EntRel'* as a sense of implicit relation.

2.2.3 Non-explicit Sense Classifier

The distribution of all senses in the train set is given in Table 4. The current shared task only asks

*Abbreviations: Expansionz(Expn.), Conjunction(Conj.), Restatement(Rest.), Contingency(Cont.), Instantiation(Inst.), Temporal(Temp.), Asynchronous(Asyn.), Precedence(Prec.), Comparison(Comp.),Concession(Conc.), Synchrony(Sync.), Asynchronous(Asyn.), Succession(Suc.), alternative(alt.), Condition(Cond.), Exception(Exc.)

	Blind	Test	Dev
Explicit Conn. F	0.8168	0.7867	0.8609
Explicit Conn. P	0.8117	0.7784	0.8528
Explicit Conn. R	0.8219	0.7952	0.8691
Arg1 Arg2 Ext. F	0.047	0.0457	0.0681
Arg1 Arg2 Ext. P	0.0453	0.0424	0.0643
Arg1 Arg2 Ext. R	0.0488	0.0495	0.0724
Arg1 Ext. F	0.0629	0.0657	0.0845
Arg1 Ext. P	0.0607	0.061	0.0798
Arg1 Ext. R	0.0653	0.0712	0.0898
Arg2 Ext. F	0.2182	0.2166	0.264
Arg2 Ext. P	0.2104	0.2011	0.2492
Arg2 Ext. R	0.2266	0.2347	0.2806
Parser F	0.0358	0.0443	0.0655
Parser P	0.0346	0.0411	0.0618
Parser R	0.0372	0.048	0.0696

Table 5: Detailed Results

us to detect 15 senses, which are marked by star. We can see that senses below the double line account less than 1%. Based on this observation, we decide only consider those significant sense.

What's more, we can see that the most frequent sense is *'EntRel'*. This leads to our another strategy: At first we set all the candidate non-explicit senses as *'Implicit'* and view *'EntRel'* as a sense. Then when the Non-explicit Sense Classifier labels the sense as *'EntRel'*, the Non-explicit Sense Classifier re-labels the type of corresponding relation as *'EntRel'*.

Previous studies attempt to predict the missing connective of implicit relations (Zhou et al., 2010; Pitler et al., 2009) . It has been shown that connective is very predictive for the sense of the relation (Kong et al., 2014). Consequently, we can get the intuition that features for predicting the missing connective are also useful for predicting the implicit sense. Thus we use word-pair features to train our Non-explicit Sense Classifier:
b) Arg1Last The last word of *'Arg1'*.
a) Arg1First The first word of *'Arg1'*.
c) Arg2First The first word of *'Arg2'*.
d) Arg2Last The last word of *'Arg2'*.
e) FirstS Arg1First + Arg2First.
f) LastS Arg1Last + Arg2Last.
g) Arg1First3 The first three words of *'Arg1'*.
h) Arg1Last3 The last three words of *'Arg2'*.
i) Arg2First3 The first three words of *'Arg2'*.

3 Evaluation

A comprehensive evaluation towards our parser has been given in Table 5. We can see that the first step of our parser, i.e., Explicit Classifier, does a moderate job. However, our work to extract the *'Arg1'* and *'Arg2'* cannot be regarded as success. Since our parser is in a sequential mode, all steps after that receive negative impacts.

4 Conclusion and Future Work

In this paper, a sequential system is proposed to do shallow discourse parsing. We demonstrate that the whole task can be worked out by a pipeline consists of several subtasks.

In future, we will tune our Argument Labeler in order to gain a better result in the explicit part .

References

Zhongye Jia, Peilu Wang, and Hai Zhao. 2013. Grammatical error correction as multiclass classification with single model. pages 74–81, Sofia, Bulgaria, August.

Fang Kong, Hwee Tou Ng, and Guodong Zhou. 2014. A constituent-based approach to argument labeling with joint inference in discourse parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 68–77, Doha, Qatar, October.

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

Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. 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*, pages 13–16, Suntec, Singapore, August.

Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. 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*, pages 683–691, Suntec, Singapore, August.

Rashmi Prasad, Dinesh Nikhil, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The Penn discourse treebank 2.0. In *The International Conference on Language Resources and Evaluation*, Marrakech, Morocco, May.

Peilu Wang, Zhongye Jia, and Hai Zhao. 2014. Grammatical error detection and correction using a single maximum entropy model. pages 74–82, Baltimore, Maryland, USA, July.

Hai Zhao and Chunyu Kit. 2008. Parsing syntactic and semantic dependencies with two single-stage maximum entropy models. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 203–207, Manchester, August.

Hai Zhao, Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009. Multilingual dependency learning: Exploiting rich features for tagging syntactic and semantic dependencies. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 61–66, Boulder, Colorado, June.

Zhi-Min Zhou, Yu Xu, Zheng-Yu Niu, Man Lan, Jian Su, and Chew Lim Tan. 2010. Predicting discourse connectives for implicit discourse relation recognition. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1507–1514, Beijing, China, August.