

Hybrid Approach to PDTB-styled Discourse Parsing for CoNLL-2015

Yasuhisa Yoshida, Katsuhiko Hayashi, Tsutomu Hirao and Masaaki Nagata

NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan

{yoshida.y, hayashi.katsuhiko,
hirao.tsutomu, nagata.masaaki}@lab.ntt.co.jp

Abstract

This paper describes our end-to-end PDTB-styled discourse parser for the CoNLL-2015 shared task. We employed a machine learning-based approach to identify discourse relation between text spans for both explicit and implicit relations and employed a rule-based approach to extract arguments of the discourse relations. In particular, we focus on improving the implicit discourse relation identification. First, we extract adjacent pairs of sentences that have some discourse relationships by exploiting a two-class classifier from an entire document. Second, we assign sense labels for them by utilizing a multiple-class classifier. Our system achieved a 0.316 overall F-score for the development set, 0.249 for the testset and 0.157 for the blind testset.

1 Introduction

In this paper, we describe our end-to-end PDTB-styled discourse parsing system for CoNLL-2015. Our system is an extension of Ziheng et al.’s discourse parser (Ziheng et al., 2014). Our explicit connective-argument structure parser consists of three modules: (1) a connective classifier that classifies connective candidates into discourse connective or not, (2) an argument position classifier that classifies whether *Arg1* and the discourse connective co-occur in the same sentence or not. (3) a rule-based argument extraction that extracts both *Arg1* and *Arg2* using rules derived from a syntactic tree. The implicit parser consists of two modules: (1) argument pair identification that finds the pair of adjacent sentences that have some discourse relation, (2) sense labeler assigning the role of the discourse relation between the sentences.

In addition, we introduce a new evaluation measure for argument extraction. Since exact matching between arguments used in “scorer.py” provided by the organizers of CoNLL-2015 is too strict, we introduce relaxed matching for the task. The evaluation metric measures how close arguments provided by the system are to the gold arguments.

The evaluation results provided by the CoNLL-2015 official scorer show that our system achieved 5th rank in the *Arg1* extractor, 6th rank in the *Arg2* extractor, 4th rank in the *Arg1&Arg2* extractor, and 8th rank in overall performance.

2 Explicit Connective-Argument Identification

The explicit connective-argument parser consists of three steps. First, we identify discourse connectives for an entire document. Second, we determine whether *Arg1* is contained in the same sentence that includes the discourse connective. Third, we assign a sense label for each discourse connective.

2.1 Connective Classification

The connective classifier classifies ambiguous connective candidates such as “and” into discourse connective or not. We exploit lexical features and features obtained from parse trees by extending (Ziheng et al., 2014). Note that connective candidates were extracted from the PYTHON script “conn_head_mapper.py” provided by the organizers of CoNLL-2015. Features that we utilized are shown in Table 1.

We trained the classifier by using SVM with second-order polynomial kernel.

2.2 Argument Position Classification

By following (Ziheng et al., 2014), we implemented an argument position classifier that classifies the location of the arguments of arbitrary

Type	Features
Context	$C_s, \text{POS}_s, \{\text{Word}_u\}, \{\text{POS}_u\}$
Parse Tree	$\text{Path}(C_s, \text{root}), \text{Parent}(C_s), \text{depth}(C_s), \text{RightSib}(C_s), \text{LeftSib}(C_s)$
	$u = s - 5, \dots, s - 1, s + 1, \dots, s + 5$

Table 1: Features used in connective classifier

discourse connective into “same sentence” (SS) or “previous sentence” (PS). SS indicates both *Arg1* and *Arg2* are located in the same sentence that contains the discourse connective. PS indicates *Arg1* is located in the sentence previous to that containing both the discourse connective and *Arg2*. We utilized context features in Table 1 and the position of the connective C_s : start, middle, or end.

We also trained the classifier by using SVM with second-order polynomial kernel.

2.3 Sense Classification

We assign majority sense ℓ^* for each discourse connective C_s as follows:

$$\ell^* = \arg \max_{\ell \in L} \text{freq}(C_s, \ell). \quad (1)$$

L is a set of sense labels used in training data and freq returns the frequency of co-occurrences of the discourse connective and sense label.

3 Implicit Connective-Argument Relationship Identification

The implicit parser consists of two steps. First is the argument identification step. In this step, we examine whether an adjacent sentence pair in the same paragraph has a discourse relation or not. Second is the sense classification step. Given a pair of sentences, we classify it into a predefined sense label.

3.1 Argument Position Identification

In the argument identification step, following Ghosh et al. (2011), the identifier examines all adjacent sentence pairs within each paragraph. For each pair of sentences (S_i, S_{i+1}) , we identify the existence of a discourse relation. To identify the existence of the relation (binary classification), we used SVM with the following features.

- First unigram, last unigram, and first trigram of S_i and S_{i+1} .
- S_i (or S_{i+1}) contains modality words or not.

- Word pairs $(w_i, w_{i+1}) \in S_i \times S_{i+1}$
- Brown cluster pairs feature defined in Rutherford and Xue (2014)
- Sentence-to-sentence discourse dependency tree features including existence of dependency edges and rhetorical relation labels. Discourse dependency trees are defined in Li et al. (Li et al., 2014).

If the identifier identifies that a pair of sentences (S_i, S_{i+1}) has the discourse relation, we heuristically regard S_i as *Arg1* and S_{i+1} as *Arg2*.

3.2 Sense Classification

In the sense classification step, we classify the discourse relation between a pair of sentences (S_i, S_{i+1}) into five senses: “Expansion”, “Contingency”, “Temporal”, “Comparison”, and “EntRel”. To classify the sense of a pair of sentences, we used multi-class SVM. We used the same features described in the argument position identification step. To increase the number of training data, we used the (inter-sentential) explicit training data as the additional training data (Rutherford and Xue, 2015). We removed a connective from each instance in the explicit training data and treated them as implicit training data. The accuracy of classification into five senses is still low because the distribution of the senses is imbalanced. Following Rutherford and Xue (2014), we resampled the instances in the training data of sense classification to balance the distribution of the senses.

4 Argument Extractor

We utilized two rule-based argument extractors. One extracts both *Arg1* and *Arg2* from the same sentence (SS). The other extracts *Arg1* and *Arg2* from adjacent sentences respectively (PS).

4.1 SS Cases

4.1.1 Subordinating Conjunctions

We adopted Dinesh et al. (2005)’s tree subtraction method for subordinating conjunctions. This method takes a constituent parse tree as an input and detects argument spans as follows:

- (1) set a node variable x to the last word of the target connective,
- (2) set x to the parent node of x and repeat until x has label SBAR or S and set a node variable *Arg2* to the node of x ,

	Dev			Test			Blind		
	P	R	F	P	R	F	P	R	F
Con.	.924	.857	.889	.918	.866	.891	.925	.353	.510
<i>Arg1</i>	.658	.549	.599	.719	.584	.644	.638	.330	.435
<i>Arg2</i>	.768	.640	.698	.587	.477	.526	.765	.395	.521
<i>Arg1&Arg2</i>	.566	.471	.514	.488	.397	.438	.522	.269	.356
Overall	.348	.290	.316	.279	.226	.249	.230	.119	.157

Table 2: Official evaluation results.

- (3) set x to the parent node of x and repeat again until x has label SBAR or S and set a node variable $Arg1$ to the current node of x ,
- (4) consider $span(Arg2)$ as the span of argument2 and $span(Arg1) \setminus span(Arg2)$ as that of argument1, where $span(\cdot)$ is a function mapping a node \cdot to a set of words dominated by the node.

4.1.2 Coordinating Conjunctions

For coordinating conjunctions, we also define a rule-based method that works on a constituent tree:

- (1) set a node variable x to the last word of the target connective,
- (2) set a node variable y to x and x to the parent node of x , and repeat while the leftmost word in $span(x)$ is equal to that in $span(y)$, and after the process, add y and the more right child nodes of x into a set $Arg2_set$,
- (3-1) if a node labeled with S or SBAR is contained in the set of the more right child nodes of x than y , set a node variable $Arg1$ to the node,
- (3-2) otherwise, set x to the parent node x and repeat until x has label SBAR or S, and set a node variable $Arg1$ to the node of x ,
- (4) consider $union_span(Arg2_set)$ as the span of argument2 and $span(Arg1) \setminus union_span(Arg2_set)$ as that of argument1, where $union_span(\cdot)$ is a function mapping a node set \cdot to the union of each word set $span(Arg2)$ for $Arg2 \in Arg2_set$.

4.1.3 Discourse Adverbials & Implicit Argument Structures

We did not treat the discourse adverbial connective-argument and inter-sentential implicit

argument structures because their frequencies are not high in the training data.

4.2 PS Cases

In the PS cases, our rule-based extraction method is very simple and has only two processes: (1) remove sentence end symbols such as . ! ?. and (2) remove brace expressions enclosed in sentence start and end brackets like “”. This method repeats (1) and (2) until unchanged.

5 Evaluation Results

Table 2 shows the official evaluation results. From the results, explicit connective identification and the *Arg2* extractor performed well, but performance of the *Arg1* extractor and sense classification was not very good. Thus, the overall performance is significantly degraded. Table 3 shows the official evaluation results for explicit relations. Compared with the testset, the accuracies for the blind testset drastically dropped. This is because our programs might failed to identify some connectives. Table 4 shows the official evaluation results for implicit relations. Among the participants, our implicit parser performed well (1st rank in the *Arg1&Arg2* extractor and 2nd rank in the overall performance). Previous study like Ghosh et al. (2011) jointly extracted the argument and classified the sense with a single classifier. Our system performed well since we split our system into the argument extractor and the sense classifier.

“scorer.py” employs exact matching for argument extraction, and when the span of the argument provided by systems exactly matches the span of the human annotated argument, the scorer evaluates the system’s tuples. However, the boundaries of human annotated arguments are blurry. The span of the argument may differ from the span annotated by another human. Thus, we evaluate our argument extractor with relaxed

	Dev			Test			Blind		
	P	R	F	P	R	F	P	R	F
Con.	.924	.857	.889	.918	.866	.891	.924	.353	.510
<i>Arg1</i>	.578	.537	.557	.475	.448	.461	.509	.194	.281
<i>Arg2</i>	.749	.696	.722	.705	.664	.684	.689	.263	.380
<i>Arg1&Arg2</i>	.498	.462	.479	.400	.377	.388	.392	.149	.216
Overall	.447	.415	.430	.355	.335	.345	.307	.117	.169

Table 3: Official evaluation results for explicit relations.

	Dev			Test			Blind		
	P	R	F	P	R	F	P	R	F
<i>Arg1</i>	.729	.546	.625	.708	.491	.579	.692	.438	.537
<i>Arg2</i>	.788	.589	.675	.733	.509	.601	.804	.508	.623
<i>Arg1&Arg2</i>	.641	.480	.549	.596	.413	.488	.588	.372	.456
Overall	.237	.177	.203	.184	.128	.151	.191	.121	.148

Table 4: Official evaluation results for implicit relations.

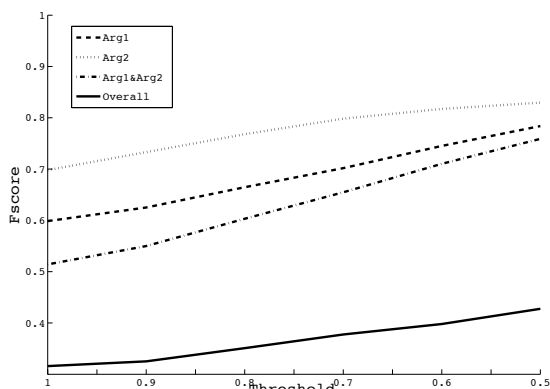


Figure 1: Evaluation results with relaxed matching.

matching. We compute token-based arg-Fscore between the system argument and the gold argument that is defined as follows:

$$\text{Prec.} = \frac{|A_s \cap A_g|}{|A_s|}, \quad (2)$$

$$\text{Rec.} = \frac{|A_s \cap A_g|}{|A_g|}, \quad (3)$$

$$\text{arg-Fscore} = \frac{2 * \text{Prec.} * \text{Rec.}}{\text{Prec.} + \text{Rec.}}. \quad (4)$$

A_s indicates a set of tokenIDs obtained from the system argument. A_g indicates a set of tokenIDs obtained from the gold argument. Then, we regard the system argument that has a certain threshold arg-Fscore as the correct argument.

Figure 1 shows evaluation results with thresholds from 1.0 to 0.5. When we set the threshold to 0.5, *Arg1&Arg2* Fscore achieved 0.7. This implies that our system can detect most of the correct positions of both explicit and implicit connectives but can not extract the correct span of the arguments. Moreover, overall performance is still low because of error caused by the sense classification modules.

6 Conclusion

In this paper, we presented our PDTB-styled full discourse parser for CoNLL-2015. We extended the work by (Ziheng et al., 2014). The experimental results show that our performed well on explicit connective identification and *Arg1* extraction, but not on *Arg2* extraction and sense classification.

References

- Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2005. Attribution and the (non-) alignment of syntactic and discourse arguments of connectives. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 29–36. Association for Computational Linguistics.
- Sucheta Ghosh, Sara Tonelli, Giuseppe Riccardi, and Richard Johansson. 2011. End-to-end discourse parser evaluation. In *Fifth IEEE International Conference on Semantic Computing (ICSC), 2011; September 18-21, 2011; Palo Alto, United States*, pages 169–172.

- Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25–35, Baltimore, Maryland, June. Association for Computational Linguistics.
- Attapol Rutherford and Nianwen Xue. 2014. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 645–654, Gothenburg, Sweden, April. Association for Computational Linguistics.
- Attapol Rutherford and Nianwen Xue. 2015. Improving the inference of implicit discourse relations via classifying explicit discourse connectives. In *Proc. of the 2015 NAACL*. Association for Computational Linguistics.
- Lin Ziheng, Ng Hwee Tou, and Kan Min-Yen. 2014. A PDTB-styled End-to-End Discourse Parser. *Natural Language Engineering*, 20:151–184.