DA-IICT Submission for PDTB-styled Discourse Parser

Devanshu Jain
DA-IICT
Gandhinagar, Gujarat
India
devanshu.jain919@gmail.com

Prasenjit Majumder
DA-IICT
Gandhinagar, Gujarat
India
prasenjit.majumder@gmail.com

Abstract

The CONLL 2016 Shared task focusses on building a Shallow Discourse Parsing system, which is given a piece of newswire text as input and it returns all discourse relations in that text in the form of discourse connectives, its two arguments and the relation sense. We have built a parser for the same. We follow a pipeline architecture to build the system. We employ machine learning methods to train our classifiers for each component in the pipeline. The system achieves an overall F1 score of 0.1065 when tested on blind dataset provided by the task organisers. On the same dataset, for explicit relations, F1 score of 0.2067 is achieved, while for non explicit relations, an F1 score of 0.0112 is achieved.

1 INTRODUCTION

Discourse Parsing is the process of assigning a discourse structure to the input provided in the form of natural language. The term "Shallow" signifies that the annotation of one discourse relation is independent of all other discourse relations, thus leaving room for a high level analysis that may attempt to connect them.

For the purpose of training and testing the system, we used PDTB (Penn Discourse Tree Bank), which is a discourse-level annotation on top of PTB (Penn Tree Bank). The corpus provides annotation for all discourse relations present in the documents. A discourse relation is composed of discourse connectives, its two arguments and the relation sense. PDTB provides a list of 100 discourse connectives, which may indicate the presence of a relation. A discourse connective can fall in any of 3 categories: Coordinating Conjunctions (e.g.: and, but, etc.), Subordinating Conjunctions (e.g.: if, because, etc.) or Discourse Adverbial (e.g.: however, also, etc.).

There are four kinds of relations, namely

1. Explicit
2. Implicit
3. AltLex (Alternative Lexicalisation)
4. EntRel (Entity Based Coherence)

Explicit Relations are marked by the presence of 100 connectives pre-defined by PDTB. Implicit Relations are realised by the reader. There are no words explicitly indicating the relationship. Sometimes, words not pre-defined like connectives by PDTB indicate a relationship. Such relations are called AltLex relations. EntRel relations exist between two sentences in which same entity is being realised. EntRel relations do not have a sense. Some examples are specified in figure 1. Here, the underlined word represents the discourse connective. Italicised text represents argument 1 and bold text represents argument 2. The right indented text following each relation represents the relation sense. The text in the bracket represents the relation type.

There are many challenges associated with this task. Firstly, we need to identify when a word works as a discourse connective and when it does not. In figure 1, consider examples 1 and 3. Both relations contain the word *and* which is present in the list of explicit connectives. But it acts as a discourse connective in example 1 and not in 3. In 3, it just links *political* and *currency* in a noun phrase. Secondly, we need to extract the arguments from sentences. And finally, we need to identify the relation sense.

Study of discourse parsing has a variety of applications in the field of Natural Language Processing. For instance, in summarisation systems,

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 122–128,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

1. *The agency has already spent roughly $19 billion selling 34 insolvent SLs*, and **it is likely to sell or merge 600 by the time the bailout concludes**.

Expansion.Conjunction (Explicit)

2. *But it doesn't take much to get burned.* Implicit = FOR EXAMPLE **Political and currency gyrations can whipsaw the funds.**

Expansion.Restatement.Specification (Implicit)

3. *Political and currency gyrations can whipsaw the funds.* AltLex [Another concern]: **The funds' share prices tend to swing more than the broader declared San Francisco batting coach Dusty Baker after game two market.**

Expansion.Conjunction (AltLex)

4. *Pierre Vinken, 61 years old, will join the board as a non-executive director Nov. 29.* **Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group.**

(EntRel)

Figure 1: Examples of various types of discourse relations

redundancy is an important aspect. We can analyse discourse relations with *Expansion* sense to weed out the redundant material. Also, in Question Answering systems, we can make use of relations with *Cause* senses to answer the *why* questions.

The report is organised as follows. Section 2 gives a brief overview of the system. Section 3 describes each component in detail and features deployed to build our parser. Section 4 reports the evaluation strategy and results achieved by our parser.

2 System Overview

There are five major components involved in the process of discourse parsing as shown in figure 2.

1. Explicit Connective Classifier
2. Explicit Argument Labeller
3. Explicit Sense Classifier
4. Non Explicit Classifier
5. Non Explicit Argument Extractor

Explicit Connective Classifier identifies the cases when explicit connectives are being used as discourse connectives as opposed to when they are not.

Explicit Argument Labeller extracts arguments of the relation. This component itself consists of two sub-components:

- Argument Position Identifier
- Argument Extractor

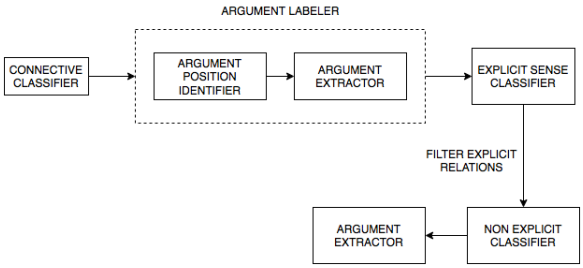

Figure 2: System Pipeline

In PDTB corpus for explicit relations, argument 2 is always syntactically bound to the connective (i.e. it is in the same sentence as connective). As far as argument 1 is concerned, it can either be in one of the previous sentences (PS case), in the same sentence (SS case) or after that sentence (FS case). Since, FS cases' occurance was too low (only 4 instances out of total 32000 relations), therefore, such cases are ignored by our system. Argument Position Identifier tries to identify this relative position of argument 1 with respect to argument 2.

If the PS case appears, then the immediately previous sentence is considered as the sentence containing argument 1. This is true for 92% of the cases in training data. Argument Extractor extracts the argument span from the sentence.

Explicit Sense Classifier identifies the relation sense. It is important to identify this as same connective may convey different meanings in different contexts. For example the word *since* can either be used in different senses as shown in figure 3. In 1, it is used in temporal sense while in 2, it is being used in causal sense.

1. There have been more than 100 mergers and acquisitions within the European paper industry since the most recent wave of friendly takeovers was completed in the U.S. in 1986.

2. It was a far safer deal for lenders since NWA had a healthier cash flow and more collateral on hand.

Figure 3: *Since* being used in different senses

Non Explicit Classifier tries to identify one of the non-explicit relations (Implicit, AltLex, EntRel) and otherwise NoRel (no relation) between adjecent sentences within the same paragraph.

Non Explicit Argument Extractor tries to extract the argument spans for non-explicit relations.

For the purpose of classification, our system uses MaxEnt Classification Algorithm without smoothing.

3 COMPONENTS AND FEATURES

3.1 Connective Classifier

The input to this component is free text from the documents. We sift through all the words in all the documents and identify the occurences of predefined explicit connectives. Then, we identify whether these connectives actually work as discourse connectives or not. For this task, we used Pitler and Nenkova 's (2009) syntactic features. Lin et al. (2014) approached this problem by using POS tags and context based features . They used used features from syntax tree, namely path from connective word to the root and compressed path (i.e. same subsequent nodes in the path are clubbed). We too, have used the similar features, as shown in table 1. Here, C-syn features refer to the combination of Connective string with each of syntactic feature and syn-syn features mean the pairing of a syntactic feature with another different syntactic feature.

3.2 Argument Labeller

Here, we first identify the relative position of argument 1 with respect to argument 2. Given this position, we extract the arguments from sentences.

3.2.1 Argument Position Identifier

To identify the position of argument 1, we extract the features mentioned in table 2:

3.2.2 Argument Extractor

After predicting the position of argument 1, we employed different tactics for different positions:

- If the position is SS (that is, both arguments are in same sentence), then we use constituency based approach by Kong et.al. without Joint Inference to extract arguments. This consists of two steps:
  - Pruning: In the parse tree of sentence, identify the node dominating all the connective words. From that node move towards the root and collect all the siblings. If this node does not exactly contain the connective words, collect all its children too. These nodes are termed as constituents.
  - Classification: For all these constituents, we extract the features mentioned in table 3.
- If the position is PS, then we consider the immediately previous sentence as a candidate for containing argument 1 and the sentence containing connective string as a candidate for containing argument 2. Extracting the arguments from sentence is a two step process:
  - Cause Splitter: We split the sentence into clauses using punctuation symbols. For the resulting clauses, we again separate SBAR (Subordinating clauses) components from them.
  - Now we classify each of these clauses. For immediately previous sentence, a clause can belong to either Arg1 or none and for the sentence containing connetive string, a clause may belong to Arg2 or none. To classify each clause, for both Arg1 and Arg2 , we employ the features mentioned in table 4.

Feature Type	Feature ID	Feature
Lexical	1	Connective String
	2	Lowercased Connective String
	3	POS tag of Connective String
	4	Word previous to first word of connective String
	5	Word previous to first word of connective + Connective String
	6	POS tag of the word previous to first word of Connective String
	7	POS tag of the word previous to first word of Connective String + POS tag of Connective String
	8	Word next to last word of Connective String
	9	Connective String + Word next to last word of Connective String
	10	POS tag of the word next to last word of Connective String
	11	POS tag of Connective String + POS tag of the word next to last word of Connective String
	12	1st Previous Word + Connective String + 1st Next Word
	13	1st Previous Word's POS + Connective POS + 1st Next Word's POS
Syntactic	14	Path of connective to root in syntax tree
	15	Compressed path of connective to root in syntax tree
	16	Self Category : Parent of the connective in syntax tree
	17	Parent Category : Parent of self category in syntax tree
	18	Left Sibling Category : Left sibling of self category in syntax tree
	19	Right Sibling Category : Right sibling of self category in syntax tree
	20	C-syn features
	21	syn-syn features

Table 1: Features for Connective Classifier

Feature ID	Feature
1	Connective String
2	Position of Connective String in sentence
3	POS tag of Connective String
4	1st previous word to Connective String
5	POS tag of 1st previous word to Connective String
6	2nd previous word to Connective String
7	POS tag of 2nd previous word to Connective String
8	1st previous word + Connective String
9	POS of 1st previous word + POS of Connective String
10	2nd previous word + Connective String
11	POS of 2nd previous word + POS of Connective String

Table 2: Features for Argument Position Classifier

Feature ID	Feature
1	Connective String
2	Lowercased Connective String
3	Category of Connective String : Subordinating, Coordinating or Discourse Adverbials
4	Constituent Context: Value of Constituent Node + its parent + its left sibling + its right sibling
5	Path of Connective String to the constituent node in syntax tree
6	Relative Position of constituent node with respect to Connective String
7	Path of Connective String to the constituent node in syntax tree + whether number of left siblings of Connective String ¿ 1

Table 3: Features for Kong's approach in SS case

Feature ID	Feature
1	Production Rules in the clause
2	Lowercased Verbs in the clause
3	Lemmatised Verbs in the clause
4	Connective String
5	Lowercased Connective String
6	Category of Connective String : Subordinating, Coordinating or Discourse Adverbials
7	First word in this clause
8	Last word in this clause
9	Last word in previous clause
10	First word in next clause
11	Last word in previous clause + First word in this clause
12	Last word in this clause + First word in next clause
13	Position of this clause in sentence: start, middle or end

Table 4: Features for Classifying clauses in PS case

Feature Type	Feature ID	Feature
Lexical Features	1	Connective String
	2	Lowercased Connective String
	3	POS tag of Connective String
	4	Previous word to Connective String + Connective String
Syntactic Features	5	Self Category : Parent of the connective in syntax tree
	6	Parent Category : Parent of self category in syntax tree
	7	Left Sibling Category : Left sibling of self category in syntax tree
	8	Right Sibling Category : Right sibling of self category in syntax tree
	9	C-syn features
	10	syn-syn features

Table 5: Features for Explicit Sense Classifier

Feature ID	Feature
1	Production Rules in syntax tree
2	Dependency Rules in dependency tree
3	Word Pair features
4	First 3 terms of argument 2 sentence

Table 6: Features for Non Explicit Classifier

Feature ID	Feature
1	Production Rules in syntax tree
2	Lower cased verbs in this clause
3	Lemmatised Verbs in this clause
4	First Word in this clause
5	Last Word in this clause
6	Last Word in previous clause
7	Fist word in next clause
8	Last Word in previous clause + First word in this clause
9	Last Word in this clause + First Word in next clause
10	Position of this clause in the sentence

Table 7: Features for Non Explicit Argument Extraction

3.3 Explicit Sense Classifier

To determine the relation sense, we use Lin's as well as Pitler's features, as shown in table 5.

3.4 Non Explicit Classifier

Non Explicit Relations occur between adjacent sentences within same paragraph. We consider the first sentence as the one containing argument 1 and second containing argument 2. Then, we extract the features mentioned in table 6.

3.4.1 Argument Extractor

To extract argument spans for Non Explicit and Non EntRel Relations, we first use clause splitter as mentioned before and then extract the features for each clause as mentioned in table 7. For EntRel relations, we simply mention the first sentence as argument 1 and second sentence as argument 2.

4 EXPERIMENTS AND RESULTS

4.1 System Setup

We used the training datasets provided by CONLL 2016 organisers (LDC2016E50). In addition we also used the brown clusters (3200 classes). For Stemming purposes, we used snowball stemmer and for lemmatising, we used stanford core nlp library.

For the purpose of classification, we used Apache OpenNLP implementation of MaxEnt classifier. We used Java programming language to implement the parser.

4.2 Evaluation Strategy

A relation is seen correct iff:

- The discourse connective is correctly detected (for explicit relations)
- Sense of relation is correctly predicted.
- Text spans of two arguments as well as their labels (Arg1 and Arg2) are correctly predicted. Partial matches are not identified as correct.

4.3 Results

Results are mentioned in tables 8. As we can see, explicit connective classifier achieves only a precision score of around 0.77 while the best team previous year (Wang) achieved a precision of 0.93. This is not good enough and perhaps is the major reason for error being propagated towards subsequent components. The results of non explicit relations were also discouraging with an F1 score of only 0.012.

5 Conclusion and Further Work

This paper describes the PDTB-styled discourse parser system we implemented for CONLL '16 shared task. We divided the system into different components and arrange in a pipeline. We apply Maximum Entropy for each of these components.

It is an ongoing work. We plan to incorporate deep learning mehods in each component to try to improve the system. We also plan to do feature selection to optimise the components of our system.

References

DINES, N., LEE, A., MILTSAKAKI, E., PRASAD, R., JOSHI, A., AND WEBBER, B. Attribution and the (non-)alignment of syntactic and discourse arguments of connectives. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie*

	Dev Set			Test Set			Blind Test		
Components	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Connectives	0.6971	0.9405	0.8007	0.6706	0.9436	0.7840	0.7770	0.9371	0.8496
Arg1	0.2820	0.3662	0.3186	0.2517	0.3249	0.2836	0.2349	0.3242	0.2724
Arg2	0.3384	0.4394	0.3824	0.3074	0.3968	0.3464	0.3259	0.4498	0.3779
Arg1 & Arg2	0.1818	0.2360	0.2054	0.1496	0.1931	0.1686	0.1489	0.2055	0.1727
Sense	0.1774	0.1385	0.1556	0.1319	0.1023	0.1153	0.1269	0.0918	0.1065
Overall	0.1774	0.1385	0.1556	0.1319	0.1023	0.1153	0.1269	0.0918	0.1065

Table 8: Overall Results

in the Sky (Stroudsburg, PA, USA, 2005), CorpusAnno '05, Association for Computational Linguistics, pp. 29–36.

KNOTT, A. *A Data-Driven Methodology for Motivating a Set of Coherence Relations*. PhD thesis, Department of Artificial Intelligence, University of Edinburgh, 1996.

KONG, F., NG, H. T., AND ZHOU, G. 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, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL* (2014), pp. 68–77.

LIN, Z., NG, H. T., AND KAN, M. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering 20*, 2 (2014), 151–184.

PITLER, E., AND NENKOVA, A. Using syntax to disambiguate explicit discourse connectives in text. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore, Short Papers* (2009), pp. 13–16.

POTTHAST, M., GOLLUB, T., RANGEL, F., ROSSO, P., STAMATATOS, E., AND STEIN, B. Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14)* (Berlin Heidelberg New York, Sept. 2014), E. Kanoulas, M. Lupu, P. Clough, M. Sanderson, M. Hall, A. Hanbury, and E. Toms, Eds., Springer, pp. 268–299.

RUTHERFORD, A., AND XUE, N. 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, EACL 2014, April 26-30, 2014, Gothenburg, Sweden* (2014), pp. 645–654.

WANG, J., AND LAN, M. A refined end-to-end discourse parser. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task* (Beijing, China, July 2015), Association for Computational Linguistics, pp. 17–24.

XUE, N., NG, H. T., PRADHAN, S., WEBBER, B., RUTHERFORD, A., WANG, C., AND WANG, H. The conll-2016 shared task on shallow discourse parsing. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning - Shared Task* (Berlin, Germany, August 2016), Association for Computational Linguistics.