The Virginia Tech System at CoNLL-2016 Shared Task on Shallow Discourse Parsing

Prashant Chandrasekar[1] **Xuan Zhang**[1] **Saurabh Chakravarty**[1]
Arijit Ray[2] **John Krulick**[1] **Alla Rozovskaya**[1]
[1]Department of Computer Science
[2]Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060
{peecee,xuancs,saurabc,ray93,jtk0,alla}@vt.edu

Abstract

This paper presents the Virginia Tech system that participated in the CoNLL-2016 shared task on shallow discourse parsing. We describe our end-to-end discourse parser that builds on the methods shown to be successful in previous work. The system consists of several components, such that each module performs a specific subtask, and the components are organized in a pipeline fashion. We also present our efforts to improve several components – explicit sense classification and argument boundary identification for explicit and implicit arguments – and present evaluation results. In the closed evaluation, our system obtained an F1 score of 20.27% on the blind test.

1 Introduction

The CoNLL-2016 shared task on shallow discourse parsing is an extension of last year's competition where participants built end-to-end discourse parsers. In this paper, we present the Virginia Tech system that participated in the CoNLL-2016 shared task. Our system is based on the methods and approaches introduced in earlier work that focused on developing individual components of an end-to-end shallow discourse parsing system, as well as the overall architecture ideas that were introduced and proved to be successful in the competition last year.

Our discourse parser consists of multiple components that are organized using a pipeline architecture. We also present novel features – for the explicit sense classifier and argument extractors – that show improvement over the respective components of state-of-the-art systems submitted last year. In the closed evaluation track, our system achieved an F1 score of 20.27% on the official blind test set.

The remainder of the paper is organized as follows. Section 2 describes the shared task. In Section 3, we present our system architecture. In Section 4, each component is described in detail. The official evaluation results are presented in Section 5. Section 6 concludes.

2 Task Description

The CoNLL-2016 shared task (Xue et al., 2016) focuses on shallow discourse parsing and is a second edition of the task. The task is to identify discourse relations that are present in natural language text. A discourse relation can be expressed explicitly or implicitly. Explicit discourse relations are those that contain an overt discourse connective in text, e.g. *because*, *but*, *and*. Implicit discourse relations, in contrast, are not expressed via an overt discourse connective. Each discourse relation is also associated with two arguments – Argument 1 (Arg1) and Argument 2 (Arg2) – that can be realized as clauses, sentences, or phrases; each relation is labeled with a sense. The overall task consists of identifying all components of a discourse relation – explicit connective (for an explicit relation), arguments with exact boundaries, as well as the sense of a relation. In addition to explicit and implicit relations that are related by an overt or a non-overt discourse connective, two other relation types (*AltLex* and *EntRel*) are marked and need to be identified. The arguments of these two relation types always correspond to entire sentences. Examples below illustrate an explicit relation (1), an implicit relation (2); *AltLex* (3) and *EntRel* (4). The connective is underlined; Arg1 is italicized, and Arg2 is in bold in each example. The relation sense is shown in parentheses.

1. *He believes in what he plays*, <u>and</u> **he plays superbly**. (Expansion.Conjunction) [wsj_0207]

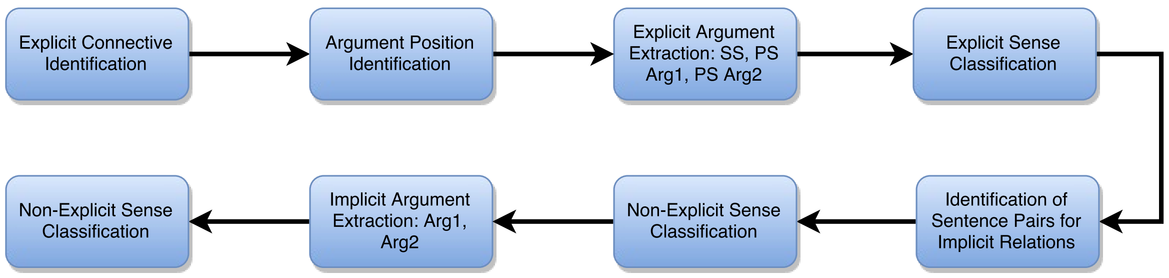
Explicit Connective Identification
Argument Position Identification
Explicit Argument Extraction: SS, PS Arg1, PS Arg2
Explicit Sense Classification
Identification of Sentence Pairs for Implicit Relations
Non-Explicit Sense Classification
Implicit Argument Extraction: Arg1, Arg2
Non-Explicit Sense Classification

Figure 1: Overview of the system architecture.

2. *In China, a great number of workers are engaged in pulling out the male organs of rice plants using tweezers, and one-third of rice produced in that country is grown from hybrid seeds.* Implicit=on the other hand **At Plant Genetic Systems, researchers have isolated a pollen-inhibiting gene that can be inserted in a plant to confer male sterility**. (Comparison.Contrast) [wsj_0209]

3. *On a commercial scale, the sterilization of the pollen-producing male part has only been achieved in corn and sorghum feed grains.* That's because **the male part, the tassel, and the female, the ear, are some distance apart on the corn plant**. (Contingency.Cause.Reason) [wsj_0209]

4. *In a labor-intensive process, the seed companies cut off the tassels of each plant, making it male sterile.They sow a row of male-fertile plants nearby, which then pollinate the male-sterile plants.* EntRel **The first hybrid corn seeds produced using this mechanical approach were introduced in the 1930s and they yielded as much as** 20% **more corn than naturally pollinated plants**. [wsj_0209]

The training and the development data for the shared task was adapted from the Penn Discourse Treebank 2.0 (PDTB-2.0) (Prasad et al., 2008). Our system was trained on the training partition and tuned using the development data. Results in the paper are reported for the development and the test sets from PDTB, as well as for the blind test.

3 System Description

The system consists of multiple modules that are applied in a pipeline fashion. This architecture is a standard approach that was originally proposed in Lin et al. (2014) and was followed with slight variations by systems in the last year competition (Xue et al., 2015). Our design most closely resembles the pipeline proposed by the top system last year (Wang and Lan, 2015), in that argument extraction for explicit relations is performed separately for Arg1 and Arg2, the non-explicit sense classifier is run twice. The overall architecture of the system is shown in Figure 1.

Given the input text, the connective classifier identifies explicit discourse connectives. Next, the position classifier is invoked that determines for each explicit relation whether Arg1 is located in the same sentence as Arg2 (SS) or in a previous sentence (PS). The following three modules – SS Arg1/Arg2 Extractor, PS Arg1 Extractor, and PS Arg2 Extractor – extract text spans of the respective arguments. Finally, the explicit sense classifier is applied.

Next, candidate sentence pairs for non-explicit relations are identified. The non-explicit sense classifier is applied to these sentence pairs. At this stage, it is run with the goal of separating *EntRel* relations from implicit relations, as *EntRel* relations have arguments corresponding to entire sentences, while the latter also require argument boundary identification. Two argument extractors are then used to determine the argument boundaries boundaries of implicit relations. After the argument boundaries of the implicit relations are identified, the non-explicit sense classifier is run again (the assumption is that with better boundary identification sense prediction can be improved).

4 System Components

This section describes each component of the pipeline and introduces novel features.

4.1 Identifying Explicit Connectives

The purpose of the explicit connective classifier is to identify discourse connectives in text. This is a binary classifier that, given a connective word or phrase (e.g. *but* or *if . . . then*) determines whether the connective functions as a discourse connective in the specific context. We use the training data to generate a list of 145 connective words and phrases that may function as discourse connectives. Only consecutive connectives that contain up to three tokens are addressed. The features are based on previous work (Pitler et al., 2009; Lin et al., 2014; Wang and Lan, 2015). Our classifier is a Maximum Entropy classifier implemented with the NLTK toolkit (Bird, 2006).

4.2 Identifying Arg1 Position

For explicit relations, position of Arg2 is fixed to be the sentence where the connective itself occurs. Arg1, on the other hand, can be located in the same sentence as the connective or in a previous sentence. Given a connective and the sentence in which it occurs, the goal of the position classifier is to determine the location of Arg1. This is a binary classifier with two classes: SS and PS.

We employ the features proposed in Lin et al. (2014) and additional features described in last year's top system (Wang and Lan, 2015). The position classifier is trained using the Maximum Entropy algorithm and achieves an F1 score of 99.186% on the development data.

In line with prior work (Wang and Lan, 2015), we consider PS to be the sentence that immediately precedes the connective. About 10% of explicit discourse relations have Arg1 occurring in a sentence that does not immediately precede the connective. These are missed at this point.

4.3 Explicit Relations: Argument Extraction

SS Argument Extractor: SS argument extractor identifies spans of Arg1 and Arg2 of explicit relations where Arg1 occurs in the same sentence, as the connective and Arg2. We follow the constituent-based approach proposed in Kong et al. (2014), without the joint inference and enhance it using features in Wang and Lan (2015). This component is also trained with the Maximum Entropy algorithm.

PS Arg1 Extractor: We implement features described in Wang and Lan (2015) and add novel features. To identify candidate constituents, we follow Kong et al. (2014), where constituents are defined loosely based on punctuation occurring in the sentence and clause boundaries as defined by *SBAR* tags. We used the constituent split implemented in Wang and Lan (2015). Based on earlier work (Wang and Lan, 2015; Lin et al., 2014), we implement the following features: surface form of the verbs in the sentence (three features), last word of the current constituent (*curr*), last word of the previous constituent (*prev*), the first word of *curr*, and the lowercased form of the connective. The novel features that we add are shown in Table 1. These features use POS information of tokens in the constituents, punctuation between the constituents, and feature conjunctions.

PS Arg2 Extractor: Similar to PS Arg1 extractor, for this component we implement features described in Wang and Lan (2015) and add novel features. The novel features are the same as those introduced for PS Arg1 but also include the following additional features:

- *nextFirstW&puncBefore* – the first word token of *next* and the punctuation before *next*.
- *prevLastW&puncAfter* – the last word token of *prev* and the punctuation after *prev*.
- *POS of the connective string.*
- The *distance* between the connective and the position of *curr* in the sentence.

The argument extractors are trained with the Averaged Perceptron algorithm, implemented within Learning Based Java (LBJ) (Rizzolo and Roth, 2010).

4.4 Explicit Sense Classifier

The goal of the explicit sense classifier is to determine what sense (e.g. *Comparison.Contrast*, *Expansion.Conjunction*, etc.) an explicit relation conveys. A 3-level sense hierarchy has been defined in PDTB, which has four top-level senses: *Comparison*, *Contingency*, *Expansion*, and *Temporal*. We use lexical and syntactic features based on previous work and also introduce new features:

- *C* (Connective) string, *C* POS, *prev* + *C*, proposed in Lin et al. (2014).
- C self-category, parent-category of C, left-sibling-category of C, right-sibling-category of C, 4 *C-Syn* interactions, and 6 *Syn-Syn* interactions, introduced in Pitler et al. (2009).
- C parent-category linked context, previous connective and its POS of "as"(the connective and its POS of previous relation, if the connective of current relation is "as"), previous connective and its POS of "when", adopted from Wang and Lan (2015).
- Our new features: first token of *C*, second token of *C* (if exists), next word (*next*), *C* + *next*, *prev* + *next*, *prev* + *C* + *next*.

	Feature name	Description
(1)	isFirst, isLast, sameAs	is the *curr* first (last, same as) one in the sentence
(2)	currFirstWAndCurrSecondW, currLastWAndNextFirstW, prevLastW, nextFirstW	word tokens and conjunctions of *curr*, *prev*, and *next*
(3)	puncBefore, puncAfter	1 if there is a punctuation mark before (after) *curr*
(4)	currFirstPOS, currLastPOS, currFirstPOSAndCurrSecondPOS, prevLastPOS, nextFirstPOS	POS and their conjunctions in *curr*, *prev*, and *next*
(5)	conjunctions	(2)&(3)

Table 1: **Novel features used in the PS Arg1 and PS Arg2 extractors.** *Curr*, *prev*, and *next* refer to the current, previous, and next constituent in the same sentence, respectively. *W* denotes word token, and *POS* denotes the part-of-speech tag of a word. For example, *currFirstWAndCurrSecondW* refers to the first two word tokens in *curr*, while *prevLastPOS* refers to the POS of the last token of *prev*, and *nextFirstPOS* refers to the POS of the first token of *next*.

For this task, we trained two classifiers – using Maximum Entropy and Averaged Perceptron algorithms – and chose Averaged Perceptron, as its performance was found to be superior.

4.5 Identifying Non-Explicit Relations

The first step in identifying non-explicit relations is the generation of sentence pairs that are candidate arguments for a non-explicit relation. Following Wang and Lan (2015), we extract sentence pairs that satisfy the following three criteria:

- Sentences are adjacent
- Sentences occur within the same paragraph
- Neither sentence participates in an explicit relation

For all pairs of sentences that meet those criteria, we take the first sentence to be the location of Arg1, and the second sentence – the location of Arg2. This approach is quite noisy since about 24% of all consecutive sentence pairs in the training data do not participate in a discourse relation. We leave this for future work.

4.6 Non-Explicit Sense Classifier

Following previous work on non-explicit sense classification (Lin et al., 2009; Pitler et al., 2009; Rutherford and Xue, 2014), we define four sets of binary feature groups: Brown clustering pairs, Brown clustering arguments, first-last words, and production rules. Dependency rules and polarity features were also extracted, but did not improve the results and were removed from the final model.

A cutoff of 5 was used to prune all of the features. Additionally, Mutual Information (MI) was used to determine the most important features. The MI calculation took the 50 most important rules in each feature group, for each of the sixteen *level 1* and *level 2* hierarchies and *EntRel*. This provided a total of 4 groups of 800 rules.

Recall that the non-explicit sense classifier has two passes. On the first iteration, its primary goal is to separate *EntRel* from implicit relations. On the second iteration, which is performed after the argument boundaries of implicit relations are identified, the sense classifier is run again on implicit relations with the predicted argument boundaries. Note that the classifier in both cases is trained in the same way, as a multiclass classifier, even though the first time it is run with the purpose of distinguishing between *AltLex* relations and all other (implicit) relations. This component is trained with the Naïve Bayes algorithm.

4.7 Implicit Relations: Argument Extraction

The argument extractors for implicit relations are implemented in a way similar to explicit relation argument extraction. Candidate sentences are split into constituents based on punctuation symbols and clause boundaries using the *SBAR* tag. We use features in Lin et al. (2009) and Wang and Lan (2015) and augment these with novel features.

Implicit Arg1 Extractor: The Implicit Arg1 extractor employs a rich set of features. Most of these are similar to those presented for PS Arg1 and PS Arg2 extractors in that we take into account POS information, punctuation symbols that occur on the boundaries of the constituents, as well as dependency relations in the constituent itself.

One key distinction of how we define the dependency relation features is that, in contrast to prior work that treats each dependency relation as a separate binary feature, we only consider the first two relations (r1 and r2, respectively) in *curr*, *prev*, and *next*, and take their conjunctions. Our intuition is that the relations in the beginning of a constituent are most important, while the other relations are not that relevant. This approach to feature generation also avoids sparseness, which was found to be a problem in earlier work. Overall, we generate seven features that use dependency relations.

Implicit Arg2 Extractor: We use most of the features in Lin et al. (2014) and Wang and Lan (2015) to train the Arg2 extractor (for more details and explanation about the features, we refer the reader to the respective papers):

- Lowercased and lemmatized verbs in *curr*
- The first and last terms of *curr*
- The last term of *prev*
- The first term of *next*
- The last term of *prev* + the first term of *curr*
- The last term of *curr* + the first term of *next*
- The position of *curr* in the sentence: start, middle, end, or whole sentence
- Product of the *curr* and *next* production rules

5 Evaluation and Results

Evaluation in the shared task is conducted using a new web service called TIRA (Potthast et al., 2014). We first evaluate the contribution of new features in individual components in 5.1. In 5.2, we report performance of all components of the final system on the development set using gold. Finally, in 5.3, we show official results on the development, test, and blind test sets. Since the system is implemented as a pipeline, each component contributes errors. We refer to the results as no error propagation (EP) when gold predictions are used, or with EP when automatic predictions generated from previous steps are employed.

The components of our final system are trained as follows: connective, position classifier, SS Arg1/Arg2 extractor and implicit Arg2 extractor (Maximum Entropy); explicit sense, PS Arg1, PS Arg2 extractors, Implicit Arg1 extractor (Averaged Perceptron); non-explicit sense (Naïve Bayes). The choice of the learning algorithms was primarily motivated by prior work. Additional experiments on argument extractors and explicit sense classification indicated that Averaged Perceptron should be preferred for these sub-tasks. Due to time constraints, we did not compare all three algorithms on all sub-tasks.

Features	**P**	**R**	**F1**
Base features	90.96	90.14	90.55
+ new features	91.88	91.05	**91.46**

Table 2: **Explicit sense classifier.** *Base* refers to features described in Wang and Lan (2015). The new set of features is presented in Section 3. Evaluation using gold connectives and argument boundaries (no EP).

Model	**P**	**R**	**F1**
Baseline	64.79	64.79	64.79
Base features	66.67	66.67	66.67
All features	69.48	69.48	**69.48**

Table 3: **PS Arg1 extractor, no EP.** *Baseline* denotes taking the entire sentence as argument span. *Base* features refer to features used in Wang and Lan (2015).

5.1 Improving Individual Components

We first evaluate the components for which we introduce new features. We use gold annotations for evaluating the individual components below.

Explicit Sense Classifier: Table 2 evaluates the explicit sense classifier. We compare our baseline model that implements the features proposed in Wang and Lan (2015) with the model that employs additional features introduced in 4.4. Our baseline model performs slightly better than the one reported in Wang and Lan (2015): we obtain 90.55 vs. 90.14, as reported in Wang and Lan (2015). Adding the new features provides an additional improvement of almost 1 F1 point.

Extraction of Explicit Arguments: We now evaluate explicit argument extractors PS Arg1 and PS Arg2, for which novel features have been intro-

Model	**P**	**R**	**F1**
Baseline	64.32	64.32	64.32
Base features	72.30	72.30	72.30
All features	75.59	75.59	**75.59**

Table 4: **PS Arg2 extractor, no EP.** *Baseline* denotes taking the entire sentence, without the connective words, as argument span. *Base* features refer to features used in Wang and Lan (2015).

Model	P	R	F1
Baseline	58.62	58.62	58.62
All features	70.50	70.50	**70.50**

Table 5: **Implicit Arg1 extractor, no EP.** *Baseline* denotes taking the entire sentence as argument span.

Component	P	R	F1
Explicit connectives	92.80	95.10	93.97
SS Arg1	68.46	71.33	69.86
SS Arg2	83.45	86.95	85.16
SS Arg1/Arg2	63.31	65.97	64.61
PS Arg1	69.48	69.48	69.48
PS Arg2	75.59	75.59	75.59
Explicit sense	91.88	91.05	91.46
Implicit Arg1	70.50	70.50	70.50
Implicit Arg2	70.11	70.11	70.11
Implicit sense	35.25	35.25	35.25

Table 6: **Evaluation of each component on the development set (no EP).**

duced. We implement the features in Wang and Lan (2015) and add our novel features shown in Table 1. Results for PS Arg1 extractor are shown in Table 3. The baseline refers to taking the entire sentence as argument span. Overall, we obtain a 5 point improvement over the baseline method. Similarly, Table 4 shows results for PS Arg2 extractor. For PS Arg2 extractor, the classifiers are able to obtain a larger improvement compared to the baseline method. Adding new features improves the results by three points. We note that in Wang and Lan (2015) the numbers that correspond to the entire sentence baselines are not the same as those that we obtain, so we do not report a direct comparison with their models. However, our base models implement the features they use.

Implicit Arg1 Extractor: In Table 5, we evaluate the Implicit Arg1 extractor. It achieves an improvement of 12 F1 points over the baseline method that considers the entire sentence to be the argument span.

5.2 Results on the Development Set (no EP)

Performance of each component on the development set, as implemented in the submitted system, without EP, is shown in Table 6.

Component	P	R	F1
Explicit connectives	93.09	92.95	93.02
Arg1 extractor	62.60	54.03	58.00
Arg2 extractor	68.38	59.01	63.35
Arg1/Arg2	50.21	43.33	46.52
Overall performance	28.58	33.59	**30.88**

Table 7: **Official results on the development set.**

Component	P	R	F1
Explicit connectives	89.92	91.51	90.71
Arg1 extractor	56.83	45.80	50.72
Arg2 extractor	63.54	51.21	56.71
Arg1/Arg2	42.24	34.04	37.70
Overall performance	20.80	25.84	**23.05**

Table 8: **Official results on the test set.**

5.3 Official Evaluation Results

The overall system results on the three data sets – development, test, and blind test – are shown in Tables 7, 8, and 9, respectively.

Component	P	R	F1
Explicit connectives	88.13	90.41	89.25
Arg1 extractor	50.87	47.02	48.87
Arg2 extractor	65.51	60.55	62.93
Arg1/Arg2	39.45	36.47	37.90
Overall performance	19.51	21.09	**20.27**

Table 9: **Official results on the blind test set.**

6 Conclusion

This paper introduces an end-to-end discourse parser for English developed for the CoNLL-2016 shared task. The entire system includes multiple components, which are organized in a pipeline fashion. We also present novel features and improve performance of several system components by incorporating these new features.

Acknowledgments

The authors thank the anonymous reviewers for the insightful comments on the paper.

References

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics.

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 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 343–351. Association for Computational Linguistics.

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, 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: Volume 2-Volume 2*, pages 683–691. Association for Computational Linguistics.

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)*.

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.

N. Rizzolo and D. Roth. 2010. Learning based java for rapid development of nlp systems. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta, 5.

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

Jianxiang Wang and Man Lan. 2015. A refined end-to-end discourse parser. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 17–24, Beijing, China, July. Association for Computational Linguistics.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Rashmi Prasad, Christopher Bryant, and Attapol T Rutherford. 2015. The CoNLL-2015 shared task on shallow discourse parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 1–16.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Bonnie Webber, Attapol Rutherford, Chuan Wang, and Hongmin Wang. 2016. The CoNLL-2016 shared task on multilungual shallow discourse parsing. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning - Shared Task*.