OPT: Oslo–Potsdam–Teesside Pipelining Rules, Rankers, and Classifier Ensembles for Shallow Discourse Parsing

Stephan Oepen[♣], Jonathon Read[♠], Tatjana Scheffler[♡], Uladzimir Sidarenka[♡◇], Manfred Stede[♡], Erik Velldal[♣]. and Lilja Øvrelid[♣]

[♣] University of Oslo, Department of Informatics
[♠] Teesside University, School of Computing
[♡] University of Potsdam, FSP Cognitive Science
[◇] Retresco GmbH

Abstract

The OPT submission to the Shared Task of the 2016 Conference on Natural Language Learning (CoNLL) implements a 'classic' pipeline architecture, combining binary classification of (candidate) explicit connectives, heuristic rules for non-explicit discourse relations, ranking and 'editing' of syntactic constituents for argument identification, and an ensemble of classifiers to assign discourse senses. With an end-to-end performance of 27.77 F_1 on the English 'blind' test data, our system advances the previous state of the art (Wang & Lan, 2015) by close to four F_1 points, with particularly good results for the argument identification sub-tasks.

1 Introduction

Being able to recognize aspects of discourse structure has recently been shown to be relevant for tasks as diverse as machine translation, question-answering, text summarization, and sentiment analysis. For many of these applications, a 'shallow' approach as embodied in the PDTB can be effective. It is shallow in the sense of making only very few commitments to an overall account of discourse structure and of having annotation decisions concentrate on the individual instances of discourse relations, rather than on their interactions.

Previous work on this task has usually broken it down into a set of sub-problems, which are solved in a pipeline architecture (roughly: identify connectives, then arguments, then discourse senses; Lin et al., 2014). While adopting a similar pipeline approach, the OPT discourse parser also builds on and extends a method that has previously achieved state-of-the-art results for the detection of speculation and negation (Velldal et al., 2012; Read et al., 2012). It is interesting to observe that an abstractly similar pipeline—disambiguating trigger expressions and then resolving their in-text 'scope'—yields strong performance across linguistically diverse tasks. At the same time, the original system has been substantially augmented for discourse parsing as outlined below. There is no closely corresponding sub-problem to assigning discourse senses in the analysis of negation and speculation; thus, our sense classifier described has been developed specifically for OPT.

2 System Architecture

Our system overview is shown in Figure 1. The individual modules interface through JSON files which resemble the desired output files of the Task. Each module adds the information specified for it. We will describe them here in thematic blocks, while the exact order of the modules can be seen in the figure. Relation identification (§3) includes the detection of explicit discourse connectives and the stipulation of non-explicit relations. Our argument identification module (§4) contains separate subclassifiers for a range of argument types and is invoked separately for explicit and non-explicit relations. Likewise, the sense classification module (§5) employs separate ensemble classifiers for explicit and non-explicit relations.

3 Relation Identification

Explicit Connectives Our classifier for detecting explicit discourse connectives extends the work by Velldal et al. (2012) for identifying expressions of speculation and negation. The approach treats the set of connectives observed in the training data as a closed class, and 'only' attempts to disambiguate occurrences of these token sequences in new data. Connectives can be single- or multi-token sequences (e.g. 'as' vs. 'as long as'). In cases

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 20–26,
Berlin, Germany, August 7-12, 2016.

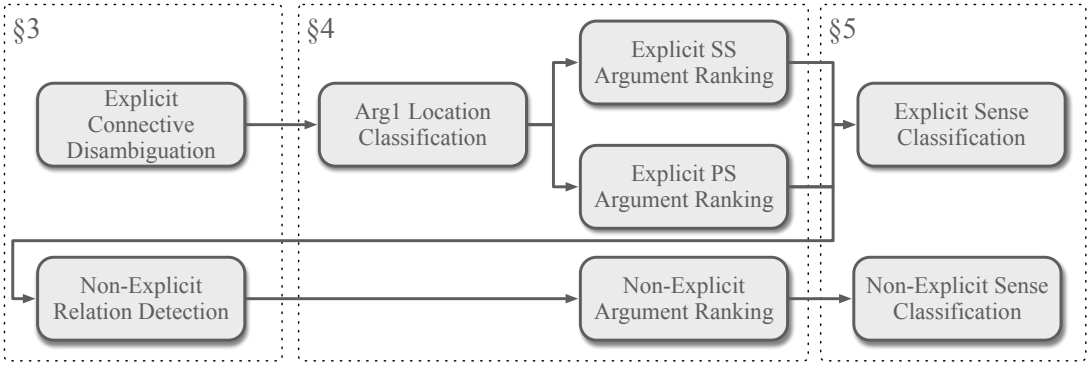

Figure 1: OPT system overview: Dotted boxes indicate sections that describe particular components.

of overlapping connective candidates, OPT deterministically chooses the longest sequence. The Shared Task defines a notion of *heads* in complex connectives, for example just the final token in 'shortly after'. As evaluation is in terms of matching connective heads only, these are the unit of disambiguation in OPT. Disambiguation is performed as point-wise ('per-connective') classification using the support vector machine implementation of the SVM^{light} toolkit (Joachims, 1999). Tuning of feature configurations and the error-to-margin cost parameter (C) was performed by ten-fold cross validation on the Task training set.

The connective classifier builds on two groups of feature templates: (a) the generic, surface-oriented ones defined by Velldal et al. (2012) and (b) the more targeted, discourse-specific features of Pitler & Nenkova (2009), Lin et al. (2014), and Wang & Lan (2015). Of these, group (a) comprises n-grams of downcased surface forms and parts of speech for up to five token positions preceding and following the connective; and group (b) draws heavily on syntactic configurations extracted from the phrase structure parses provided with the Task data. During system development, a few thousand distinct combinations of features were evaluated, including variable levels of feature conjunction (called interaction features by Pitler & Nenkova, 2009) within each group. These experiments suggest that there is substantial overlap between the utility of the various feature templates, and n-gram window size can to a certain degree be traded off with richer syntactic features. Many distinct configurations yield near-identical performance in cross-validation on the training data, and we selected our final model by (a) giving preference to configurations with smaller numbers of features and lower variance across folds and (b) additionally evaluating a dozen candidate configurations against the development data. The model used in the system submission includes n-grams of up to three preceding and following positions, full feature conjunction for the 'self' and 'parent' categories of Pitler & Nenkova (2009), but limited conjunctions involving their 'left' and 'right' sibling categories, and none of the 'connected context' features suggested by Wang & Lan (2015). This model has some 1.2 million feature types.

Non-Explicit Relations According to the PDTB guidelines, non-explicit relations must be stipulated between each pair of sentences iff four conditions hold: two sentences (a) are adjacent; (b) are located in the same paragraph; and (c) are not yet 'connected' by an explicit connective; and (d) a coherence relation can be inferred or an entity-based relation holds between them. We proceed straightforwardly: We traverse the sentence bigrams, following condition (a). Paragraph boundaries are detected based on character offsets in the input text (b). We compute a list of already 'connected' first sentences in sentence bigrams, extracting all the previously detected explicit connectives whose Arg1 is located in the 'previous sentence' (PS; see §4). If the first sentence in a candidate bigram is not yet 'connected' (c), we posit a non-explicit relation for the bigram. Condition (d) is ignored, since NoRel annotations are extremely rare and EntRel vs. Implicit relations are disambiguated in the downstream sense module (§5). We currently do not attempt to recover the AltLex instances, because they are relatively infrequent and there is a high chance for false positives.

4 Argument Identification

Our approach to argument identification is rooted in previous work on resolving the scope of spec-

	Explicit		Non-Explicit	
	Arg1	**Arg2**	**Arg1**	**Arg2**
Ambiguity	3.4	4.6	3.1	3.2
Sentence spanning	.033	.070	.021	.015
Non-SS/PS `Arg1`	.116		.047	
Align. w/o edits	.483	.535	.870	.900
Align. with edits	.813	.840	.882	.900
Upper-bound	.692	.781	.822	.887

Table 1: Observations of arguments in the training data. Alignment rates are with respect to all arguments that do not span sentence boundaries and are located in SS or PS, while the upper-bound is with respect to all arguments.

	Exp. PS		Exp. SS		Non-Exp.	
	Arg1	**Arg2**	**Arg1**	**Arg2**	**Arg1**	**Arg2**
Connective Form			•			
Connective Category		•				
Connective Precedes				•		
Following Token				•		
Initial Token					•	
Path to Root		•	•		•	•
Path to Connective		•	•	•		
Path to Initial Token					•	•
Preceding Token		•	•		•	•
Production Rules	•	•			•	•
Size						•

Table 2: Feature types used to describe candidate constituents for argument ranking.

ulation and negation, in particular work by Read et al. (2012): We generate candidate arguments by selecting constituents from a sentence parse tree, apply an automatically-learned ranking function to discriminate between candidates, and use the predicted constituent's surface projection to determine the extent of an argument. Like for explicit connective identification, all classifiers trained for argument identification use the SVM^{light} toolkit and are tuned by ten-fold cross-validation against the training set.

Predicting Arg1 Location For non-explicit relations we make the simplifying assumption that `Arg1` occurs in the sentence immediately preceding that of `Arg2` (PS). However, the `Arg1`s of explicit relations frequently occur in the same sentence (SS), so, following Wang & Lan (2015), we attempt to learn a classification function to predict whether these are in SS or PS. Considering all features proposed by Wang & Lan, but under cross-validation on the training set, we found that the significantly informative features were limited to: the connective form, the syntactic path from connective to root, the connective position in sentence (tertiles), and a bigram of the connective and following token part-of-speech.

Candidate Generation and Ranking Candidates are limited to clausal constituents as these account for the majority of arguments, offering substantial coverage while restricting the ambiguity (i.e., the mean number of candidates per argument; see Table 1). Candidates whose projection corresponds to the true extent of the argument are labeled as correct; others are labeled as incorrect.

We experimented with various feature types to describe candidates, using the implementation of ordinal ranking in SVM^{light} (Joachims, 2002). These types comprise both the candidate's surface projection (including: bigrams of tokens in candidate, connective, connective category (Knott, 1996), connective part-of-speech, connective precedes the candidate, connective position in sentence, initial token of candidate, final token of candidate, size of candidate projection relative to the sentence, token immediately following the candidate, token immediately preceding the candidate, tokens in candidate, and verbs in candidate) and the candidate's position in the sentence's parse tree (including: path to connective, path to connective via root, path to initial token, path to root, path between initial and preceding tokens, path between final and following tokens, and production rules of the candidate subtree).

An exhaustive search of all permutations of the above feature types requires significant resources. Instead we iteratively build a pool of feature types, at each stage assessing the contribution of each feature type when added to the pool, and only add a feature type if its contribution is statistically significant (using a Wilcoxon signed-rank test, $p < .05$). The most informative feature types thus selected are syntactic in nature, with a small but significant contribution from surface features. Table 2 lists the specific feature types found to be optimal for each particular type of argument.

Constituent Editing Our approach to argument identification is based on the assumption that arguments correspond to syntactically meaningful units, more specifically we require arguments to be

clausal constituents (S/SBAR/SQ). In order to test this assumption, we quantify the alignment of arguments with constituents in `en.train`, see Table 1. We find that the initial alignment (Align w/o edits) is rather low, in particular for Explicit arguments (.48 for `Arg1` and .54 for `Arg2`). We therefore formulate a set of *constituent editing* heuristics, designed to improve on this alignment by including or removing certain elements from the candidate constituent. We apply the following heuristics, with conditions by argument type (`Arg1` vs. `Arg2`), connective type (explicit vs. non-explicit) and position (SS vs. PS) in parentheses.

- add conjunction (CC) preceding constituent (`Arg1`)
- cut clause headed by connective (`Arg1`, explicit, SS)
- cut constituent-final CC (`Arg1`)
- cut constituent-final wh-determiner (`Arg1`)
- cut constituent-initial CC (`Arg2`, explicit)
- cut relative clause, i.e. SBAR initiated by WHNP/WHADVP
- cut connective
- cut initial and final punctuation

Following editing, the alignment of arguments with the edited constituents improves considerably for explicit `Arg1`s (.81) and `Arg2`s (.84), see Table 1.

Limitations The assumptions of our approach mean that the system upper-bound is limited in three respects. Firstly, some arguments span sentence boundaries (see Sent. Span in Table 1) meaning there can be no single aligned constituent. Secondly, not all arguments correspond with clausal constituents (approximately 1.7% of arguments in `en.train` align with a constituent of some other type). Finally, as reported in Table 1, several `Arg1`s occur in neither the same sentence nor the immediately preceding sentence. Table 1 provides system upper-bounds taking each of these limitations into account.

5 Relation Sense Classification

In order to assign senses to the predicted relations, we apply an ensemble-classification approach. In particular, we use two separate groups of classifiers: one group for predicting the senses of explicit relations and another one for analyzing the senses of non-explicit relations. Each of these groups comprises the same types of predictors (presented below) but uses different feature sets.

Majority Class Senser The first classifier included in both of our ensembles is a simplistic system which, given an input connective (`none` for non-explicit relations), returns a vector of conditional probabilities of its senses computed on the training data.

W&L$_{\text{LSVC}}$ Another prediction module is a reimplementation of the Wang & Lan (2015) system—the winner of the previous iteration of the ConNLL Shared Task on shallow discourse parsing. In contrast to the original version, however, which relies on the Maximum Entropy classifier for predicting the senses of explicit relations and utilizes the Naïve Bayes approach for classifying the senses of the non-explicit ones, both of our components (explicit and non-explicit) use the `LIBLINEAR` system (Fan et al., 2008)—a speed-optimized SVM (Boser et al., 1992) with linear kernel. In our derived classifier, we adopt all features[1] of the original implementation up to the Brown clusters, where instead of taking the differences and intersections of the clusters from both arguments, we use the Cartesian product (CP) of the Brown groups similarly to the token-CP features of the UniTN system from last year (Stepanov et al., 2015). Additionally, in order to reduce the number of possible CP attributes, we take the set of 1,000 clusters provided by the organizers of the Task instead of differentiating between 3,200 Brown groups as was done originally by Wang & Lan (2015).

Unlike the upstream modules in our pipeline, whose model parameters are tuned by 10-fold cross-validation on the training set, the hyper-parameters of the sense classifiers are tweaked towards the development set, while using the entire training data for computing the feature weights. This decision is motivated by the wish to harness the full range of the training set, since the number of the target classes to predict is much bigger than in the preceding sub-tasks and because some of the senses, e.g. Expansion.Exception, only appear a dozen of times in the provided dataset. For training the final system, we use the Crammer-Singer multi-class strategy (Crammer & Singer, 2001) with $L2$-loss,

[1] A detailed description of these features can be found in the original paper by Wang & Lan (2015) and their code posted on github: `https://github.com/lanmanok/conll2015_discourse`.

	WSJ Test Set					Blind Test Set				
	2015	2016		OPT		2015	2016		OPT	
	F_1	F_1	P	R	F_1	F_1	F_1	P	R	F_1
Explicit Connectives	94.8	**98.9**	96.4	92.5	94.4	91.9	**98.4**	93.5	90.1	91.8
Explicit Arg1 Extraction	50.7	**53.8**	53.1	50.9	52.0	49.7	52.4	53.4	51.5	**52.4**
Explicit Arg2 Extraction	**77.4**	76.7	74.1	71.1	72.6	74.3	75.2	76.6	73.8	**75.2**
Explicit Both Extraction	45.2	**45.3**	44.9	43.0	43.9	41.4	44.0	44.9	43.2	**44.0**
Explicit Sense Micro-Average			38.6	40.2	39.4			33.9	35.1	34.5
Non-Explicit Arg1 Extraction	67.2	69.9	72.0	68.0	**69.9**	60.9	**66.8**	63.7	65.5	64.6
Non-Explicit Arg2 Extraction	68.4	71.5	73.5	69.5	**71.5**	74.6	**79.1**	75.3	77.5	76.4
Non-Explicit Both Extraction	53.1	53.5	55.0	52.0	**53.5**	50.4	**58.1**	51.3	52.8	52.0
Non-Explicit Sense Micro-Average			17.5	18.6	18.0			22.0	21.6	21.9
All Both Extraction	49.4	**49.6**	50.2	47.8	48.9	46.4	**50.6**	48.3	48.1	48.2
Overall Parser Performance	29.7	**30.7**	27.5	28.9	28.2	24.0	27.8	27.8	27.8	**27.8**

Table 3: Per-component breakdown of system performance, compared to top performers in 2015/16.

optimizing the primal objective and setting the error penalty term C to 0.3.

W&L$_{\text{XGBoost}}$ Even though linear SVM systems achieve competitive results on many important classification tasks, these systems can still experience difficulties with discerning instances that are not separable by a hyperplane. In order to circumvent this problem, we use a third type of classifier in our ensembles—a forest of decision trees learned by gradient boosting (XGBoost; Friedman, 2000). For this part, we take the same set of features as in the previous component and optimize the hyperparameters of this module on the development set as described previously. In particular, we set the maximum tree depth to 3 and take 300 tree estimators for the complete forest.

Prediction Merging To compute the final predictions, we first obtain vectors of the estimated sense probabilities for each input instance from the three classifiers in the respective ensemble and then sum up these vectors, choosing the sense with the highest final score. More formally, we compute the prediction label $\hat{y}_i$ for the input instance x_i as $\hat{y}_i = \arg\max \sum_{j=1}^{n} \vec{v}_j$, where n is the number of classifiers in the ensemble (in our case three), and $\vec{v}_j$ denotes the output probability vector of the j-th predictor. Since the XGBoost implementation we use, however, can only return classifications without actual probability estimates, we obtain a probability vector for this component by assigning the score $1-\epsilon$ to the predicted sense class (with the ϵ-term determined on the development and set to 0.1) and uniformly distributing the ϵ-weight among the remaining senses.

6 Experimental Results

Overall Results Table 3 summarizes `OPT` system performance in terms of the metrics computed by the official scorer for the Shared Task, against both the WSJ and 'blind' test sets. To compare against the previous state of the art, we include results for the top-performing systems from the 2015 and 2016 competitions (as reported by Xue et al., 2015, and Xue et al., 2016, respectively). Where applicable, best results (when comparing F_1) are highlighted for each sub-task and -metric. The highlighting makes it evident that the `OPT` system is competitive to the state of the art across the board, but particularly so on the argument identification sub-task and on the 'blind' test data: In terms of the WSJ test data, `OPT` would have ranked second in the 2015 competition, but on the 'blind' data it outperforms the previous state of the art on all but one metric for which contrastive results are provided by Xue et al.. Where earlier systems tend to drop by several F_1 points when evaluated on the non-WSJ data, this 'out-of-domain' effect is much smaller for `OPT`. For comparison, we also include the top scores for each submodule achieved by any system in the 2016 Shared Task.

Non-Explicit Relations In isolation, the stipulation of non-explicit relations achieves an F_1 of 93.2 on the WSJ test set ($P = 89.9$, $R = 96.8$). Since this sub-module does not specify full argument spans, we match gold and predicted relations based on the sentence identifiers of the arguments only. False positives include `NoRel` and missing relations. About half of the false negatives are relations within the same sentence (across a semicolon).

	WSJ Test Set			Blind Set		
	Arg1	**Arg2**	**Both**	**Arg1**	**Arg2**	**Both**
Explicit (SS)	.683	.817	.590	.647	.783	.519
Explicit (PS)	.623	.663	.462	.611	.832	.505
Explicit (All)	.572	.753	.474	.586	.782	.473
Non-explicit (All)	.744	.743	.593	.640	.758	.539
Overall	.668	.749	.536	.617	.769	.509

Table 4: Isolated argument extraction results (PS refers to the immediately preceding sentence only).

System	WSJ Test Set			Blind Set		
	Exp	**Non-Exp**	**All**	**Exp**	**Non-Exp**	**All**
2015	**90.79**	34.45	61.27	76.44	**36.29**	**54.76**
Majority	89.30	21.40	54.02	75.91	30.46	51.39
W&L$_{\text{LSVC}}$	89.63	**37.18**	**62.29**	**77.86**	33.05	53.66
W&L$_{\text{XGB}}$	89.41	34.12	60.64	76.27	34.42	53.62
OPT	89.95	33.53	60.64	76.81	33.66	53.54
LSTM*	89.90	33.76	60.78	77.63	33.69	53.29
OPT*	90.01	**41.12**	**64.70**	77.06	**37.20**	**55.55**

Table 5: Isolated results for sense classification (the bottom* model was not part of the submission).

Arguments Table 4 reports the isolated performance for argument identification. Most results are consistent across types of arguments, the two data sets, and the upper-bound estimates in Table 1, with Arg1 harder to identify than Arg2. However an anomaly is the extraction of Arg2 in explicit relations where the Arg1 is in the immediately preceding sentence, which is poor in the WSJ Test Set but better in the blind set. This may be due to variance in the number of PS Arg1s in the respective sets, but will be investigated further in future work on error analysis.

Sense Classification The results of the sense classification subtask without error propagation are shown in Table 5. As can be seen from the table, the LIBLINEAR reimplementation of the Wang & Lan system was the strongest component in our ensemble, outperforming the best results on the WSJ test set from the previous year by 0.89 F_1. The XGBoost variant of that module typically achieved the second best scores, being slightly better at predicting the sense of non-explicit relations on the blind test set. The majority class predictor is the least competitive part, which, however, is made up for by the simplicity of the model and its relative robustness to unseen data.

Finally, we report on a system variant that was not part of the official OPT submission, shown in the bottom rows of Table 5.

In this configuration, we added more features (types of modal verbs in the arguments, occurrence of negation, as well as the form and part-of-speech tag of the word immediately following the connective) to the W&L-based classifier of explicit relations, re-adjusting the hyper-parameters of this model afterwards; increased the ϵ-term of the XGBoost component from 0.1 to 0.5; and, finally, replaced the majority class predictor with a neural LSTM model (Hochreiter & Schmidhuber, 1997), using the provided Word2Vec embeddings as input. This ongoing work shows clear promise for substantive improvements in sense classification.

7 Conclusion & Outlook

The most innovative aspect of this work, arguably, is our adaptation of constituent ranking and editing from negation and speculation analysis to the sub-task of argument identification in discourse parsing. Premium performance (relatively speaking, comparing to the previous state of the art) on this sub-problem is in no small part the reason for overall competitive performance of the OPT system, despite its relatively simplistic architecture. The constituent ranker (and to some degree also the 'targeted' features in connective disambiguation) embodies a strong commitment to syntactic analysis as a prerequisite to discourse parsing. This is an interesting observation, in that it (a) confirms tight interdependencies between intra- and inter-utterance analysis and (b) offers hope that higher-quality syntactic analysis should translate into improved discourse parsing. We plan to investigate these connections through in-depth error analysis and follow-up experimentation with additional syntactic parsers and types of representations. Another noteworthy property of our OPT system submission appears to be its relative resilience to minor differences in text type between the WSJ and 'blind' test data. We attribute this behavior at least in part to methodological choices made in parameter tuning, in particular cross-validation over the training data—yielding more reliable estimates of system performance than tuning against the much smaller development set—and selective, step-wise inclusion of features in model development.

Acknowledgments

We are indebted to Te Rutherford of Brandeis University for his effort in preparing data and infrastructure for the Task, as well as for shepherding our team and everyone else through its various stages. We are grateful to two anonymous reviewers for comments on an earlier version of this manuscript.

References

Boser, B. E., Guyon, I., & Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory* (p. 144 – 152). Pittsburgh, PA, USA.

Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, *2*, 265 – 292.

Fan, R., Chang, K., Hsieh, C., Wang, X., & Lin, C. (2008). LIBLINEAR. A library for large linear classification. *Journal of Machine Learning Research*, *9*, 1871 – 1874.

Friedman, J. H. (2000). Greedy function approximation. A gradient boosting machine. *Annals of Statistics*, *29*, 1189 – 1232.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735 – 1780.

Joachims, T. (1999). Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in kernel methods. Support vector learning.* Cambridge, MA, USA: MIT Press.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (p. 133 – 142). Edmonton, Canada.

Knott, A. (1996). *A data-driven methodology for motivating a set of coherence relations*. Unpublished doctoral dissertation, University of Edinburgh.

Lin, Z., Ng, H. T., & Kan, M.-Y. (2014). A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, *20*(2), 151 – 184.

Pitler, E., & Nenkova, A. (2009). Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the 47th Meeting of the Association for Computational Linguistics* (p. 13 – 16). Singapore.

Read, J., Velldal, E., Øvrelid, L., & Oepen, S. (2012). UiO1. Constituent-based discriminative ranking for negation resolution. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics* (p. 310 – 318). Montréal, Canada.

Stepanov, E. A., Riccardi, G., & Bayer, A. O. (2015). The UniTN discourse parser in CoNLL 2015 Shared Task. Token-level sequence labeling with argument-specific models. In *Proceedings of the 19th Conference on Natural Language Learning* (p. 25 – 31). Bejing, China.

Velldal, E., Øvrelid, L., Read, J., & Oepen, S. (2012). Speculation and negation: Rules, rankers and the role of syntax. *Computational Linguistics*, *38*(2), 369 – 410.

Wang, J., & Lan, M. (2015). A refined end-to-end discourse parser. In *Proceedings of the 19th Conference on Natural Language Learning* (p. 17 – 24). Bejing, China.

Xue, N., Ng, H. T., Pradhan, S., Prasad, R., Bryant, C., & Rutherford, A. (2015). The CoNLL-2015 shared task on shallow discourse parsing. In *Proceedings of the 19th Conference on Natural Language Learning: Shared task* (p. 1 – 16). Bejing, China.

Xue, N., Ng, H. T., Pradhan, S., Webber, B., Rutherford, A., Wang, C., & Wang, H. (2016). The CoNLL-2016 shared task on multilingual shallow discourse parsing. In *Proceedings of the 20th Conference on Natural Language Learning: Shared task.* Berlin, Germany.