A minimalist approach to shallow discourse parsing and implicit relation recognition

Christian Chiarcos, Niko Schenk

Angaben zur Veröffentlichung / Publication details:

Chiarcos, Christian, and Niko Schenk. 2015. “A minimalist approach to shallow discourse parsing and implicit relation recognition.” In *CoNLL 2015: Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task, July 30-31, 2015, Beijing, China*, edited by Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Rashmi Prasad, Christopher Bryant, and Attapol Rutherford, 42–49. Stroudsburg, PA: Association for Computational Linguistics. https://doi.org/10.18653/v1/k15-2006.

A Minimalist Approach to Shallow Discourse Parsing and Implicit Relation Recognition

Christian Chiarcos and **Niko Schenk**
Applied Computational Linguistics Lab
Goethe University Frankfurt am Main
{chiarcos,n.schenk}@em.uni-frankfurt.de

Abstract

We describe a minimalist approach to shallow discourse parsing in the context of the CoNLL 2015 Shared Task.[1] Our parser integrates a rule-based component for argument identification and data-driven models for the classification of explicit and implicit relations. We place special emphasis on the evaluation of implicit sense labeling, we present different feature sets and show that (i) word embeddings are competitive with traditional word-level features, and (ii) that they can be used to considerably reduce the total number of features. Despite its simplicity, our parser is competitive with other systems in terms of sense recognition and thus provides a solid ground for further refinement.

1 Introduction

Comprehending sentences and other textual units requires capabilities beyond capturing the lexical semantics of their components. Contextual information is needed, i.e., a semantically coherent representation of the logical structure of a text—be it written or spoken discourse, unidirectional or bidirectional communication, etc. Different formalisms have been proposed to model these assumptions in frameworks of coherence relations and discourse structure (Mann and Thompson, 1988; Lascarides and Asher, 1993; Webber, 2004). In a more applied NLP context, the goal of *shallow discourse parsing* (SDP) is to automatically detect relevant discourse units and to label the relations that hold between them. Unlike *deep discourse parsing*, a stringent logical formalization or the establishment of a global data structure, say, a tree, is not required.

With the release of the Penn Discourse Treebank (Prasad et al., 2008, PDTB), annotated training data for SDP has become available and, as a consequence, the field has considerably attracted researchers from the NLP and IR community. Informally, the PDTB annotation scheme describes a discourse unit as a syntactically motivated character span in the text, and augments with relations pointing from argument 2 (*Arg2*, prototypically, a discourse unit associated with an explicit discourse marker) to its antecedent, i.e., the discourse unit *Arg1*. Relations are labeled with a relation type (its *sense*) and the associated discourse marker (either as found in the text or as inferred by the annotator). PDTB distinguishes *explicit* and *implicit* relations depending on whether such a connector or cue phrase (e.g., *because*) is present, or not.[2] As an illustration, consider the following example from the PDTB:

> **Arg1**: *Solo woodwind players have to be creative if they want to work a lot*
> **Connector**: *because*
> **Arg2**: *their repertoire and audience appeal are limited*

In this explicit relation, Arg1 and Arg2 are directly connected by the cue word; the relation type is *Contingency.Cause.Reason*—one out of roughly 20 three-level senses marking the relation sense between any given argument pair in the PDTB.

We participate in the CoNLL 2015 Shared Task (Xue et al., 2015) with a minimalist end-to-end shallow discourse parser developed from scratch. It was, however, originally not specifically developed for this purpose, but created in preparation of more elaborate experiments on implicit inter-sentential relations in discourse, an aspect not explicitly addressed by the evaluation of the Shared Task.

[1] http://www.cs.brandeis.edu/~clp/conll15st/index.html

[2] The set of relation types is completed by alternative lexicalization (*AltLex*, discourse marker rephrased), entity relation (*EntRel*, i.e., anaphoric coherence), resp. the absence of any relation (*NoRel*).

Proceedings of the Nineteenth Conference on Computational Natural Language Learning: Shared Task, pages 42–49,
Beijing, China, July 26-31, 2015. ©2014 Association for Computational Linguistics

The remainder of the paper describes the architecture and functionality of our system: A rule-based component identifies explicit and implicit argument-pairs and two statistical, data-driven models classify senses. Our system suffers from the surface-based definition of argument spans and their evaluation as string ranges, but with respect to sense disambiguation (in particular, in terms of precision), it is competitive with other systems in the task. Inspired by the diversity of different approaches to handle the more challenging—and more interesting—non-explicit relations, our description focuses on inferring implicit senses and benefits from abstracting from traditional surface-based features in favor of distributional representations of the argument spans.

2 Related Work

At the moment, few full-fledged end-to-end discourse parsers exist, but they use different theories of discourse, e.g., PDTB (Lin et al., 2010), or RST (duVerle and Prendinger, 2009; Feng and Hirst, 2012). Most of the literature on automated discourse analysis has focused on specialized subtasks:

Argument identification is approached by, e.g., Ghosh et al. (2012) on the word and inter-sentential level, using a CRF-based approach including local and global features. Kong et al. (2014) tackle argument span detection on the constituent-level with features for subtrees and special constraints.

Explicit relation classification Classifying the senses of explicit relations is rather straightforward, given the cue phrase. Pitler and Nenkova (2009) introduce a refinement using syntactic features to disambiguate explicit connectives which increases performance close to a human baseline.

Implicit relation classification In the early attempt by Marcu and Echihabi (2002), implicit relation classification was grounded on synthetic training data (relation patterns with explicit cue phrases removed) and a Naive Bayes model trained on word-pair features. Aggregation over such word-pairs was described by Biran and McKeown (2013), while Park and Cardie (2012) optimized feature sets through feature selection, preprocessing and special binning techniques.

Out of these, implicit relation classification remains the most problematic subtask, and attracted considerable interest: Pitler et al. (2009) present an extensive evaluation of mostly linguistically motivated features for implicit sense labeling in a 4-way classification experiment. Useful indicators, among others, are verb information, polarity labels and the first and last three words of an argument. Lin et al. (2009) refine their work by introducing contextual and dependency information from the argument pairs and show that syntactic phrase-structure features help in level-2 relation type classifications. Moreover, Zhou et al. (2010) use a language model to “predict” explicit connectives from implicit relations. Our approach is most similar to the one by Rutherford and Xue (2014), who successfully integrate distributional representations to substitute word-pair features.

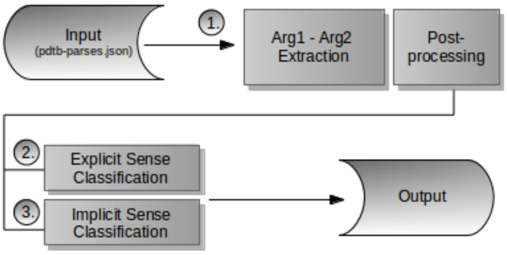

Figure 1: Our three-component SDP pipeline.

3 Approach

Our SDP system participates in the *closed track* of the Shared Task.[3] Its components are illustrated in Figure 1. Input is tokenized text in the provided JSON format including meta information about parts-of-speech and sentence boundaries.

3.1 Argument Identification

The SDP pipeline processes the documents sentence by sentence. Due to the strict time constraints of the Shared Task, we have set up a rule-based detector for both Arg1 and Arg2 spans as follows:

- Extract an *explicit* Arg1–Arg2 pair, where Arg2 is a complete sentence starting with an explicit connective.[4] The previous sentence serves as Arg1.

[3] http://www.cs.brandeis.edu/~clp/conll15st/dataset.html

[4] An exhaustive list of explicit cue words was obtained from the training section of the PDTB, ranging from unigrams to 7-grams.

- Refining step 1, we extract sentence-internal *explicit* Arg1–Arg2 pairs by applying the pattern `BOS-Arg1-cue word-punctuation-Arg2-EOS`.[5] Note that we require a punctuation symbol between both arguments to prevent the template from extracting, e.g., coordinated NPs such as *chairman and chief executive*.

- We take special care of *explicit* temporal Arg1–Arg2 relations and extract patterns of the form `BOS-cue word-Arg2-comma-Arg1-EOS`. Cue words are, e.g., *while*, *although*, *unless*.

- More complicated *explicit* patterns split the second argument into two parts by the cue word as with *however* in: *Argument identification is tough. Writing patterns, however, is easy.*

- Finally, we extract all relations between adjacent, complete sentences as Arg1 and Arg2 spans as *implicit*, iff Arg1–Arg2 is not already an explicit relation and Arg1–Arg2 does not cross a paragraph boundary.

- EntRel and AltLex relations are beyond the scope of our current parser as both taken together make up only 14.3% of all relations in the training section of the PDTB.

Post processing A rule-based *post-processor* is applied on top of the previous component. Its purpose is to fix token lists for argument spans according to the guidelines of the Shared Task as no partial credit is given for non-exact matches. For example, a leading or trailing punctuation, quote or attribution spans must not be part of any of the arguments.

This rule-based model had specifically to be developed for the Shared Task; it replaced a more elaborate argument identifier based on structured representations rather than character spans to represent the arguments of discourse relations.

3.2 Labeling Explicit Senses

Given two argument spans and an explicit connective, we aim to predict the correct relation type (sense). To this end, we trained a simple statistical model[6] in a supervised setting on all explicit relations whose only feature is the cue word itself. An exhaustive list of cue words (features) was obtained from the training section of the PDTB data. Moreover, we restricted the set of labels to those eight senses that appear only frequently enough, i.e. we excluded those whose proportion is less than 5% of all explicit senses in the training section.

3.3 Labeling Implicit Senses

A third component handles the classification of *implicit senses* for any implicit Arg1–Arg2 pair. Similar to the previous subtask, we restrict the label set (here to six senses). We trained various models only on implicit relations. Inspired by the previous literature on implicit sense classification, we experimented with different surface-based word-pair feature sets for Arg1 and Arg2, as well as more abstract representations for the word forms, such as embeddings and word vectors:[7]

1. Word-pair (WP) token features of Arg1 and Arg2: (i) normal-case (N) as encountered in the text and (ii) after lower-case normalization (l), both with frequency thresholds.

2. Similar to (1.) but using word stems (Porter, 1980) instead.

3. Similar to (1.) but using a Brown cluster 3200 representation (Turian et al., 2010) for each word form if it exists. Otherwise, we use the word form as feature.

A subsequent experiment is concerned with finding a more compact representation of both Arg1 and Arg2 spans: For each argument pair, we computed two real-valued vectors (600 features in total), in which each argument is represented by a 300-dimensional feature vector. These were obtained by summing over all skip-gram neural word embeddings (Mikolov et al., 2013) present in each argument weighted by the respective number of elements (embeddings) found in each argument. The normalization is necessary to handle sentences of different lengths.

[5] `BOS` and `EOS` mark the beginning and the end of sentence, respectively.

[6] In all our experiments, we made use of the JAVA implementation of *libsvm* (Chang and Lin, 2011) with linear kernel and default parameters.

[7] A word-pair is defined as the cross product of any combination of words in both Arg1 and Arg2. Punctuation symbols were removed before processing. All features are treated as boolean if present (true) or absent (false).

Testing the effect of both Brown clusters and neural word embeddings, a final experiment combines them into one feature set for each implicit argument pair.

4 Evaluation

4.1 Argument Identification

In the overall task (based on the blind test set), our system is ranked at position 13 – rather poorly compared to 17 submitted systems in total (including a baseline). This is due to the imperfect argument identification, and in particular due to the erroneous recognition of explicit cue phrases. The system suffers from low overall recall of the identified explicit argument spans, including the connective.[8] A simple error analysis reveals that patterns in which cue phrases do not directly start the second argument are hard to identify by our rule-based system. Moreover, punctuation symbols pose problems to the system as well (cf. our discussion in Section 4.3). A separate evaluation shows that post-processing argument pairs improves F-score by 2%.

Despite these obvious drawbacks, we would like to draw special attention to our statistical components for sense classification: for the argument pairs which were correctly recognized, our system is ranked at position 4 for sense precision, even outperforming the best three systems. We will elaborate more on these models in the next subsection.

4.2 Explicit and Implicit Senses

The classification of explicit senses with only the connector word as single feature reaches an accuracy of 80.48% using the PDTB training–development split. This is still below state-of-the art (94% in Pitler and Nenkova, 2009)[9]—yet satisfying for our lightweight system with its original emphasis on implicit relations.

Table 1 shows the results for implicit sense classification (472 instances in total) using different feature sets.[10] First, models trained on any of the feature sets significantly outperform the majority class baseline (25.4%, *Expansion.Conjunction*).[11] Applying lower-case normalization to the input tends to improve classifier performance, but using a frequency threshold on the minimum number of occurrences of a feature does not: This is an interesting observation and not in line with the previous literature on implicit sense classification; Lin et al. (2009), for example, use a frequency cutoff of 5 for feature selection. Also, stemming as another type of normalization seems not to be useful either and yields slightly lower accuracies.

Noticeably, substituting surface-level word-pair features by the Brown Cluster 3200 embeddings yields a better performance. The difference is, however, not statistically significant.[12] More important, however, may be the positive side effect of a smaller feature space ($\approx$1.4 million) which is reduced by 23%.

We expect the skip-gram neural word embeddings (word vectors) to perform even better than Brown clusters: They are comparable in their contextual features but preserve the topology of the original feature space. Indeed, these are competitive with the low-frequency word-pair features and even significantly better than the configurations l_3, l_4, l_5. Their greatest benefit can be seen in the overall number of real-valued features per instance (which is only 600 in our setting). Finally, a combination of Brown clusters and skip-gram embeddings yields the best results for the classification of implicit senses. This gain over using the embeddings alone may possibly be attributed to non-linearities in the feature space which may be partially captured in the Brown clusters, but not with embeddings in a SVM.[13] We report detailed scores for this best-performing classifier in Table 2.

4.3 Discussion & Open Issues

4.3.1 Argument Span Identification

Exact argument identification is a crucial preprocessing step for any SDP pipeline. Our shallow

[8] Ranks for expl. Arg1-Arg2 prec., recall, F_1: 12, 10, 11. Ranks for expl. connective prec., recall, F_1: 15, 16, 15.

[9] Note, however, that this is 4-way sense classification.

[10] We also tested a broad band-width of sentiment and phrase-structure features, but with the resulting accuracies not outperforming the current experiments, these are omitted for reasons of brevity.

[11] In all experiments, we applied the χ^2 test statistic to assess significance.

[12] We have tested the other Brown cluster representations provided, as well, but 100, 320 and 1000 cluster sets yielded lower accuracies.

[13] All results reported above were obtained with linear kernels. These experiments have also been conducted with RBF and polynomial kernels, whose performance was not reported here, as it did not yield an improvement. However, truly non-linear models would be possible with multi-layered neural networks. While this may yield better results for word embeddings as features, such an experiment is left for future research.

	N_0/l_0	N_1/l_1	N_2/l_2	N_3/l_3	N_4/l_4	N_5/l_5
WP / Tokens	36.65/38.14	36.23/34.53	33.68/32.84	32.84/33.05	31.57/32.63	30.08/32.63
WP / Stems	– /36.23	– /33.89	– /32.84	– /31.99	– /33.05	– /30.72
WP / Brown Cluster 3200	36.86/38.77	35.38/35.17	33.90/36.07	35.38/34.11	34.96/33.47	32.63/33.89
Word Vectors	36.23/37.28					
WP / Brown Cluster + Word Vectors	37.28/**39.41**					

Table 1: Accuracies for 6-way implicit sense labeling and different feature sets when tokens are treated in normal-case (N) or after lower-case preprocessing (l). Subscripts indicate frequency thresholds for feature selection (0 means no threshold applied). Majority class baseline: 25.4%.

	Prec	Rec	F_1
Expansion.Conjunction	43.09	67.50	52.59
Expansion.Restatement	32.68	49.50	39.37
Comparison.Contrast	42.85	18.29	25.64
Contingency.Cause.Reason	41.26	35.61	38.23
Contingency.Cause.Result	40.00	16.32	23.18
Expansion.Instantiation	46.15	12.76	20.00

Table 2: Detailed classification scores for the best-performing classifier combining Brown Cluster 3200 and skip-gram embeddings.

discourse parser suffers from low overall recall of the correctly recognized (explicit) spans, which we see as the main source of poor performance in the task evaluation.

Even though a system description may not be the right place for a general discussion about the appropriate representation of how arguments of discourse relations are to be defined and represented, we would like to point out that we see a potential issue in the rather strict evaluation of exact matches within the Shared Task (which does not allow for partial matches). Likewise problematic is an arguable definition of gold spans for Arg1 and Arg2 in the provided training data. As an illustration consider the following example:[14]

> **Gold**:
> Arg1: *At any rate India needs the sugar*
> Arg2: *it will be in sooner or later to buy it*
>
> **Our System Output**:
> Arg1: *At any rate, she added, "India needs the sugar*
> Arg2: *it will be in sooner or later to buy it.*

At least on a general basis, both argument spans are correctly identified by our system. The only difference is that punctuation symbols and attribution spans (*she added*) are not present in the gold data. Note, however, that a rule-based removal of such patterns is far from trivial, as syntactic patterns are complex and the PDTB gold data reveals many inconsistencies, especially regarding leading and trailing punctuation symbols. In this particular example, our system is capable of

(i) identifying the correct explicit connective (*so*), and

(ii) classifying its correct sense (*Contingency.Cause.Result*).

Nevertheless, it is not given any credit, as the system's token lists do not match the gold data. Very much related to the span identification problem sketched above is the detection of discontinuous argument spans and cases in which our system adds a subordinate clause to the argument, which is not present in the gold data. We believe that—in line with the annotation guidelines of the PDTB—these are relevant factors to consider when implementing a SDP, but that it should not affect the overall evaluation in such a strict and rigid manner. We would therefore encourage future evaluations to

- *either* employ additional metrics permitting partial matches, e.g., using sliding-window metrics such as Pevzner and Hearst (2002),
- *or* to ground argument definitions in psycholinguistically more plausible models of propositions, cf. Lascarides and Asher (1993) or Kintsch (1998), resp.—their more operationalizable approximation in terms of, say, frame semantics as previously annotated for the PDTB data in the context of PropBank

[14] Document ID: `wsj_2265`, Relation ID: 36896.

and NomBank (Palmer et al., 2005; Meyers et al., 2004).

The latter idea may be challenging, as it involves efficient handling of multi-layer annotations for different major annotation projects, yet, experiments in this direction have successfully been conducted (Pustejovsky et al., 2005). This integrative direction of research has been the original focus of our system.

4.3.2 Frequency Cutoffs for Word-Pair Feature Selection

Our experiments indicate that frequency cutoffs to select word-pair features for implicit relation recognition do not seem to improve classifier performance. While some previous approaches (most notably Lin et al., 2009) incorporate cutoffs in their experiments, others do not. But if a frequency filter is applied, the specific value for the threshold is usually not motivated.

We see a possible explanation for the negative impact of cutoffs in the extremely sparse feature space: Many word-pair features which are present in the training section of the PDTB are not found in the development set and vice versa, and with frequency cutoffs applied, sparsity even grows further. Closely related to our observation are earlier findings that using even a small stop word list has adverse effects on performance, which seems implausible at first sight (Blair-Goldensohn et al., 2007).

Biran and McKeown (2013) address this issue in closer detail by replacing the sparse lexical word-pair features by more dense, aggregated score features. Based on their experiments, the authors argue that the most powerful features are mainly function words. Yet, their lack of semantic content whatsoever still calls for an explanation why they are useful in distinguishing the different types of implicit relations—except through overfitting the data.

As a side experiment, we performed 10-fold cross validation on the PDTB, and again trained implicit relations by varying the cutoff. The results are in line with our experiments reported in Table 1 showing the same trend, which reinforces the aforementioned sparsity issue.

Overall, we believe that more aggregated types of features have advantages over sparse features and that they are better in representing the underlying semantic relationship between argument pairs. We elaborate on this in our final subsection.

4.3.3 Abstracting from Surface-Level Features

Our experiments for implicit relation classification have shown that is is beneficial to abstract from surface-level (token) features for two reasons:

(i) word embeddings seem to express a more general, semantic representation of the underlying relationship between two arguments in the discourse and

(ii) the number of features involved in a classification can be significantly reduced which has a positive effect on the computational side.

Future research should be concerned with a closer inspection of how combinations of word embeddings can be used to increase classification results, especially when no explicit connectives are available. Instead of vector addition, as applied in our setting, we think that traditional vector-based similarity measures comparing both arguments spans seem to be highly promising in approaching their underlying semantic relationship.

5 Conclusion

In the context of the CoNLL 2015 Shared Task, we have described a minimalist approach to shallow discourse parsing with an emphasis on implicit relation recognition.

Our system combines task-specific adaptations, i.e., rule-based discourse unit identification via templates, with data-driven models to infer senses of (esp. implicit) discourse relations.

We described the system architecture and experiments conducted on implicit sense labeling. In this context, we motivated the need to model the relationship between arguments in a more abstract way using distributional representations instead of surface-based features. Our experiments are in line with previous work (most notably by Rutherford and Xue, 2014), while having shown that more abstract representations are at least equally powerful in predicting the correct senses and, also, that sparsity issues can be overcome. A slight improvement in performance has yielded a combination of distributional profiles for argument spans (Brown clusters and skip-gram neural word embeddings) which is promising and should be addressed in closer detail in future work.

References

Or Biran and Kathleen McKeown. 2013. Aggregated Word Pair Features for Implicit Discourse Relation Disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 69–73.

Sasha Blair-Goldensohn, Kathleen McKeown, and Owen Rambow. 2007. Building and Refining Rhetorical-Semantic Relation Models. In Candace L. Sidner, Tanja Schultz, Matthew Stone, and ChengXiang Zhai, editors, *HLT-NAACL*, pages 428–435. The Association for Computational Linguistics.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

David A. duVerle and Helmut Prendinger. 2009. A Novel Discourse Parser Based on Support Vector Machine Classification. 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*, ACL '09, pages 665–673, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vanessa Wei Feng and Graeme Hirst. 2012. Text-level Discourse Parsing with Rich Linguistic Features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 60–68, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sucheta Ghosh, Giuseppe Riccardi, and Richard Johansson. 2012. Global Features for Shallow Discourse Parsing. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 150–159.

Walter Kintsch. 1998. *Comprehension: A Paradigm for Cognition*. Cambridge University Press, Cambridge.

Fang Kong, Tou Hwee 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 (EMNLP)*, pages 68–77. Association for Computational Linguistics.

Alex Lascarides and Nicholas Asher. 1993. Temporal Interpretation, Discourse Relations and Commonsense entailment. *Linguistics and Philosophy*, 16(5):437–493.

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*, EMNLP '09, pages 343–351, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2010. A PDTB-Styled End-to-End Discourse Parser. *CoRR*, abs/1011.0835.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Daniel Marcu and Abdessamad Echihabi. 2002. An Unsupervised Approach to Recognizing Discourse Relations. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 368–375, Stroudsburg, PA, USA. Association for Computational Linguistics.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. Annotating Noun Argument Structure for NomBank. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. European Language Resources Association (ELRA).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Comput. Linguist.*, 31(1):71–106, March.

Joonsuk Park and Claire Cardie. 2012. Improving Implicit Discourse Relation Recognition Through Feature Set Optimization. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 108–112, Seoul, South Korea, July. Association for Computational Linguistics, Association for Computational Linguistics.

Lev Pevzner and Marti A Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

Emily Pitler and Ani Nenkova. 2009. 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*, pages 13–16.

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*, ACL '09, pages 683–691, Stroudsburg, PA, USA. Association for Computational Linguistics.

M.F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *In Proceedings of LREC*.

James Pustejovsky, Adam Meyers, Martha Palmer, and Massimo Poesio, 2005. *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, chapter Merging PropBank, NomBank, TimeBank, Penn Discourse Treebank and Coreference, pages 5–12. 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. Association for Computational Linguistics.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word Representations: A Simple and General Method for Semi-Supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden, July. Association for Computational Linguistics.

Bonnie L. Webber. 2004. D-LTAG: extending lexicalized TAG to discourse. *Cognitive Science*, 28(5):751–779.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Rashmi Prasad, Christopher Bryant, and Attapol Rutherford. 2015. The CoNLL-2015 Shared Task on Shallow Discourse Parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning: Shared Task*, Beijing, China.

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: Posters*, COLING '10, pages 1507–1514, Stroudsburg, PA, USA. Association for Computational Linguistics.