The CLaC Discourse Parser at CoNLL-2016

Majid Laali[*]   **Andre Cianflone**[*]   **Leila Kosseim**

Department of Computer Science and Software Engineering
Concordia University, Montreal, Quebec, Canada
`{Laali, Cianflone, Kosseim}@encs.concordia.ca`

Abstract

This paper describes our submission (*CLaC*) to the CoNLL-2016 shared task on shallow discourse parsing. We used two complementary approaches for the task. A standard machine learning approach for the parsing of explicit relations, and a deep learning approach for non-explicit relations. Overall, our parser achieves an F_1-score of 0.2106 on the identification of discourse relations (0.3110 for explicit relations and 0.1219 for non-explicit relations) on the blind CoNLL-2016 test set.

1 Introduction

Shallow discourse parsing is defined as the identification of two discourse units, or discourse arguments, and labeling their relation. Although the topic of shallow discourse parsing has received much interest in the past few years (e.g. (Zhang et al., 2015; Weiss, 2015; Ji et al., 2015; Rutherford and Xue, 2014; Kong et al., 2014; Feng et al., 2014)), the performance of the state-of-the-art discourse parsers is not yet adequate to be used in other downstream Natural Language Processing applications. For example, the best parser submitted at CoNLL-2015 (Wang and Lan, 2015) achieved an F_1 score of 0.2400 on the blind test dataset.

For the CoNLL 2016 task of shallow discourse parsing, four types of discourse relations have to be annotated in texts (more details of the task can be found in (Xue et al., 2016)):

1. *Explicit Discourse Relations*: explicit discourse relations are explicitly signalled within the text through *discourse connectives* such as *because, however, since,* etc.

2. *Implicit Discourse Relations*: implicit discourse relations are inferred by the reader and no discourse connective is used within the text to convey the relation. As a reader, implicit discourse relations can be inferred by inserting a discourse connective (called an *implicit discourse connective*) in the text that best expresses the inferred relation.

3. *AltLex Discourse Relations*: Similarly to implicit discourse relations, AltLex are not signalled through the presence of discourse connectives in the text. However, the relation is alternatively lexicalized by some non-connective expression, hence inserting an *implicit discourse connective* to express the inferred relation would lead to a redundancy.

4. *EntRel Discourse Relations*: EntRel discourse relations are defined between two discourse arguments where only an entity-based coherence relation could be perceived.

In this paper, we report on the development and results of our discourse parser for the CoNLL 2016 shared task. As shown in Figure 1, our parser, named *CLaC Discourse Parser*, consists of two main components: the Explicit Discourse Relation Annotator and the Non-Explicit Discourse Relation Annotator .

The Explicit Discourse Relation Annotator is based on the parser that we submitted last year to CoNLL 2015 (Laali et al., 2015). For this year's submission, we improved its components by (1) adding new features (see Section 2 for more details), (2) using a sequence classifier instead of a multiclass classifier in the Discourse Argument Segmenter, and (3) defining a new component, the Discourse Argument Trimmer, to identify attributes and prune discourse arguments.
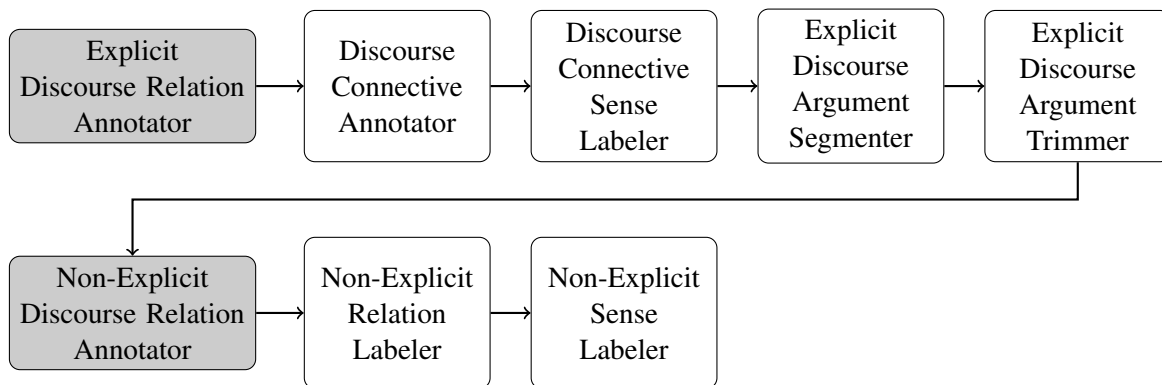
---

[*]Both authors contributed equally

Figure 1: Pipeline of the CLaC Discourse Parser

Last year's system did not address the annotation of non-explicit discourse relations (i.e. implicit, AltLex and EntRel discourse relations). For this year, we therefore built this module from scratch. The Non-Explicit Discourse Relation Annotator first uses a binary Convolutional Neural Network (ConvNet) to detect whether a relation exists in a text devoid of a discourse connective, then uses a multiclass ConvNet to label the relation.

2 Explicit Discourse Relation Annotator

Figure 1 shows the pipeline of the CLaC parser. The top row in Figure 1 focuses on the Explicit Discourse Relation Annotator. This pipeline consists of four main components: (1) *Discourse Connective Annotator*, (2) *Discourse Connective Sense Labeler*, (3) *Explicit Relation Argument Segmenter* and (4) *Discourse Argument Trimmer*.

Modules 1, 2 and 3 are based on last year's system (Laali et al., 2015) while module 4 has been newly developed to address a weak issue from last year.

2.1 Discourse Connective Annotator

The *Discourse Connective Annotator* annotates discourse connectives within a text. To label discourse connectives, the annotator first searches the input texts for terms that match any of the 100 discourse connectives listed in the Penn Discourse Treebank (Prasad et al., 2008a). Inspired by (Pitler et al., 2009), a C4.5 decision tree binary classifier (Quinlan, 1993) is used to detect if each discourse connective is used in a discourse usage or not. In addition to the six features proposed by (Pitler et al., 2009), this year we also used four of the features proposed by (Lin et al., 2014). In total 10

features were used:

1. The discourse connective text in lowercase.

2. The categorization of the case of the connective: *all lowercase* or *initial uppercase*.

3. The highest node (called the *SelfCat* node) in the parse tree that covers the connective words but nothing more.

4–6. The parent, the left sibling and the right sibling of the *SelfCat*.

7–10. The left and the right word of discourse connective and their parts of speech.

2.2 Discourse Connective Sense Labeler

Once discourse connectives have been classified as discourse usage or not, the *Discourse Connective Sense Labeler* labels the discourse relation signalled by the annotated discourse connectives with one of the 14 labels specified by the task. This component also uses a C4.5 decision tree classifier (Quinlan, 1993) with the same 10 features used by the *Discourse Connective Annotator* (see Section 2.1).

2.3 Discourse Argument Segmenter

The goal of the *Discourse Argument Segmenter* is to detect the discourse argument boundaries. This module first assumes that both discourse arguments (i.e. ARG1 and ARG2) are located in the same sentence that contains the discourse connective. If ARG1 is not found in the sentence, then the Discourse Argument Segmenter selects the immediately preceding sentence as ARG1.

We used a similar approach proposed by (Kong et al., 2014) to identify discourse arguments that appear in the same sentence. That is to say, we

first select all the constituents in the parse tree that are directly connected to one of the nodes in the path from the discourse connective to the root of the sentence and classify them into to one of three categories: *part-of-*ARG1, *part-of-*ARG2 or NON (i.e. not part of any discourse argument). Then, all constituents which are tagged as part of ARG1 or as part of ARG2 are merged to obtain the actual boundaries of ARG1 and ARG2.

Instead of using integer programming as proposed by Kong et al. (2014), we used a Conditional Random Field (CRF) in order to leverage global information (i.e. information across all constituent candidates). CRFs have been previously used for discourse argument identification (Ghosh et al., 2011) but at the token level. Kong et al. (2014)'s approach generates a sequence of constituents and therefore, CRFs can be applied at the constituent level.

We used the following categories of features for the CRF:

1. *Discourse connective features:* This category includes all 10 features used in the Discourse Connective Annotator (see Section 2.1).

2. *Constituent features:* Motivated by Kong et al. (2014)'features, we defined the following five features:

   (a) The constituents in the path from the current constituent to the *SelfCat* node in the parse tree.

   (b) The length of the path between the current constituent and the *SelfCat* node.

   (c) The context of the current constituent in the parse tree. The context of a constituent is defined by its label, the label of its parent and the label of its left and right siblings in the parse tree.

   (d) The position of the current constituent relative to the *SelfCat* node (i.e. left or right).

   (e) The syntactic production rule of the current constituent.

3. *Lexical features:* This year, we also used lexical features including the head of the current constituent and four tokens that appear in the constituent boundary (the first token of the constituent and its previous token and the last token of the constituent and its following token).

2.4 Discourse Argument Trimmer

According to the PDTB manual (Prasad et al., 2008b), annotators should keep the span of two discourse arguments as small as possible and should remove any extra information that is not necessary for the discourse relation. Following this idea, the *Discourse Argument Trimmer* is a classifier that excludes any constituent from the discourse argument span that is not related to the discourse relations.

To do so, we developed a binary classifier that labels all the constituents and tokens in the annotated discourse arguments with either *part-of-Argument* or *Not-part-of-Argument* to exclude tokens that are not part of the discourse argument. Once the classifier has labeled all the tokens and constituents, we remove from the discourse arguments all tokens that are labeled as *Not-Part-of-Argument* or part of a constituent with the *Not-Part-of-Argument* label.

A C4.5 decision tree binary classifier was developed using the following features:

1. The head of the constituent or the text of the token.

2. The label of the constituent in the syntax tree or the POS of token.

3. The position of the constituent/token (i.e whether it appears at the beginning, inside or at the end of the discourse argument).

4. The syntactic production rule of the constituent's parent and grand parent or "null" for tokens.

5. The type of the argument (i.e. ARG1 or ARG2)

6. The node label/POS of the left and right siblings of the constituent/token in the syntactic tree.

3 Non-Explicit Discourse Relation Annotator

As mentioned in Section 1, last year, the CLaC Discourse Parser did not address non-explicit relations. Therefore, for this year's participation we developed this module from scratch. Because Convolutional Neural Networks (ConvNets) have been successful at several sentence classification tasks (e.g. (Zhang and Wallace, 2016; Kim,

2014)), we wanted to investigate if similar networks could be used to address the task of non-explicit discourse relation recognition.

The Non-Explicit Discourse Relation Annotator begins where the Explicit Discourse Relation Annotator ends. The Explicit Discourse Relation Annotator only analyzes texts which contain a discourse connective; all other segments are sent to the Non-Explicit Discourse Relation Annotator.

Because these text segments may or may not contain a discourse relation, the Non-Explicit Discourse Relation Annotator first sends each text segment to a binary ConvNet to identify which segments contain a discourse relations and which do not. The Non-Explicit Discourse Relation Annotator trims trailing discourse punctuation as per the shared task requirement. Only discourses with two consecutive arguments are considered as possible non-explicit discourses. Non-discourse segments are removed from the pipeline. Sense labelling is then performed on the remaining segments using a multiclass ConvNet.

3.1 Input

The two ConvNets have an identical setup. The input to the models are pretrained word embeddings from the Google News set, as trained with Word2Vec[1]. Words not in the Google News set are randomly initialized. Word embeddings are non-static, meaning that they are allowed to change during training.

Each input to the networks is composed of the two padded discourse arguments. ARG1 is padded to the length of the longest ARG1, and ARG2 is similarly padded to the length of the longest ARG2. Since the training set contains a few unusually long arguments, we limited the argument size to the size of the 99.5[th] percentile. This reduced the length of ARG1 from 1000 to 60 words, and that of ARG2 from 400 to 61 words. This dramatically decreased the model complexity with insignificant impact on performance. The two arguments are then concatenated to form a single input. Each word is then replaced with their embedded vector representation.

Let l be the length of a single input (the number of words in the discourse plus padding, 121). Let d be the dimensionality of a word vector (300 for our pretrained embedding). Then the input to the networks, the matrix of discourse embedding, can

---

be denoted $Q \in \mathbb{R}^{l \times d}$.

3.2 Network

The network configuration is largely based on (Kim, 2014). We applied a narrow convolution over Q with height w (i.e. w words) and width d (the entire word vector) defined as region $h \in \mathbb{R}^{d \times w}$. We added a bias b and applied a nonlinear function f on the convolution to give us features c_i, where i is the ith word in the discourse input. This is shown in Formula 1.

$$c_i = f(h \cdot Q_{i:i+w-1} + b) \qquad (1)$$

The nonlinear function f in our case was the exponential linear unit (ELU) (Clevert et al., 2016), indicated in Formula 2.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(exp(x) - 1) & \text{if } x \leq 0 \end{cases} \qquad (2)$$

Since the convolution is narrow, there are $l - w + 1$ such features, giving us a *feature map* $c \in \mathbb{R}^{l-w+1}$. We applied max-over-time pooling on c to extract the most "important" feature as in Formula 3.

$$y = max(c) \qquad (3)$$

We applied 128 feature maps and pooled each one of these. We repeated the entire process 3 times for $w = 3, 4$ and 5, and concatenated them together. This gave us a final matrix $M \in \mathbb{R}^{3 \times 128}$. We reshaped M to a flat vector and applied dropout as our regularization (Srivastava et al., 2014), giving us vector $u \in \mathbb{R}^{384}$. u is fully connected to a softmax output layer where loss is measured with cross-entropy. The network was trained in mini-batches and optimized with the *Adam* algorithm (Kingma and Ba, 2015).

4 Results and Analysis

Table 1 shows the F_1 scores of the CLaC Discourse Parser and the best parser at CoNLL 2015 (Wang and Lan, 2015) for different datasets. The overall F_1 score of the CLaC parser is 0.2106 with the blind test dataset which is lower than the F_1 score of the best parser at CoNLL 2015 (i.e. 0.2400). For explicit relations, the performance of our parser (F_1=0.3110) is higher than the performance of last year's best parser (F_1=0.3038); however, for non-explicit relations there is gap between the performance of our parser (F_1=0.1219) and the performance of last year's best parser (F_1=0.1887).

	Development Dataset (PDTB)		Test Dataset (PDTB)		Blind Test Dataset (Wikinews)	
	CLaC	Best (2015)	CLaC	Best (2015)	CLaC	Best (2015)
Full Parsing						
Overall	**0.3260**	0.3851	**0.2442**	0.2499	**0.2106**	0.2400
Explicit	0.4457	0.4977	0.3572	0.3447	0.3110	0.3038
Non-Explicit	0.2167	0.2876	0.1395	0.1511	0.1219	0.1887
Identification of Explicit Discourse Connective						
Explicit	0.9203	0.9514	0.9100	0.9421	**0.9020**	0.9186
Argument Identification						
Overall	0.4929	0.5704	0.4173	0.4377	0.3912	0.4637
Explicit	0.4867	0.5352	0.4023	0.3882	**0.3989**	0.4135
Non-Explicit	0.4987	0.6014	0.4311	0.4881	**0.3844**	0.5041
Sense Labeling (Supplementary task)						
Overall	0.6222	-	0.5736	0.6802	0.5000	0.6327
Explicit	0.9074	-	0.8948	0.9079	**0.7622**	0.7685
Non-Explicit	0.3712	-	0.2813	0.4734	**0.2772**	0.5176

Table 1: F_1-score of the CLaC Discourse Parser and the best parser of 2015 with Different Datasets.

4.1 Explicit Discourse Relation Annotator

Table 1 shows that the argument segmentation component is the bottleneck of the Explicit Discourse Relation Annotator. While the CLaC Discourse parser achieves competitive results in the identification of explicit discourse connectives (F_1=0.9020) and labeling the sense signalled by the discourse connectives (F_1=0.7622) with the blind test dataset, its performance is rather low (F_1=0.3989) for the identification of the discourse argument boundaries.

Our results show that the CLaC Discourse Parser has difficulty in detecting ARG1. As Table 2 shows, the precision and recall for the identification of ARG1 (i.e. P=0.4928 and R=0.4749) are significantly lower than for ARG2 (i.e. P=0.7194 and R=0.6932). ARG2 is syntactically bound to discourse connectives and therefore, it is easier to detect its boundaries. Moreover, as mentioned in Section 2.3, our approach does not account for arguments that appear in non-adjacent sentences. However, according to Prasad et al. (2008a), 9.02% of ARG1 in the PDTB do not appear in the sentence adjacent to the discourse connective.

The exact match of CoNLL is a strict evaluation measure for the argument identification. For example, in Sentence (1), our parser did not detect the word '*it*' (boxed) and therefore, accordingly to the exact match scoring schema, the boundaries of the discourse arguments are incorrect.

	P	R	F_1
Arg1	0.4928	0.4749	0.4837
Arg2	0.7194	0.6932	0.7061
Arg1 & Arg2	0.4065	0.3917	0.3989

Table 2: Results of the CLaC Discourse parser for the identification of discourse arguments with the blind test dataset (exact match).

(1) *The law does allow the RTC to borrow from the Treasury up to $5 billion at any time.* Moreover, it **says the RTC's total obligations may not exceed $50 billion, but that figure is derived after including notes and other debt, and subtracting from it the market value of the assets the RTC holds.**[2]

Such cases where the CLaC parser misses the argument boundaries by only a few words (added or deleted) are frequent. For example, as Table 3 shows, if we evaluate the argument boundaries with the partial match metric defined in the CoNLL evaluator, the performance increases significantly. The partial match metric accepts the argument boundaries if 70% of the tokens of the identified discourse arguments are correct. Using this metric, the F_1 score of the identification of ARG1 and ARG2 increases by 0.1917 and 0.0777 respectively.

---

[2]This example is taken from the CoNLL development dataset.

	P	R	F$_1$
Arg1	0.6740	0.6768	0.6754
Arg2	0.7695	0.7986	0.7838
Arg1 & Arg2	0.6386	0.6667	0.6523

Table 3: Results of the CLaC Discourse parser for the identification of discourse argument with the blind test dataset (partial match).

We also observed that the Explicit Discourse Argument Trimmer has a difficulty detecting what parts of the texts are related to discourse relations especially if multiple events appear in the text with a TEMPORAL discourse relation. For example, in Sentence (2) the parser identified the boxed words as ARG1 and missed required information. On the other hand in Sentence (3) the parser included extra information in ARG1. This type of error appears more frequently for ARG1 which explains why the partial match metric improves the identification of ARG1 more than the identification of ARG2.

(2) *We would have to wait* ‖until we have collected on those assets‖ before **we can move forward**.

(3) ‖But the RTC also requires "working" capital *to maintain the bad assets of thrifts that are sold*‖, until **the assets can be sold separately**.

4.2 Non-Explicit Discourse Relation Annotator

Table 1 shows that for the task of non-explicit sense labelling the Non-Explicit Discourse Relation Annotator achieves an F$_1$-score of 0.2813 on the test dataset and 0.2772 on the blind dataset, versus 0.3712 on the developement dataset. The similar performance on the test and blind datasets and the 10% difference with the development dataset suggest overfitting of our neural network.

For argument segmentation, just removing tailing punctuations from consecutive sentences achieves an F$_1$-score of 0.3884. According to Prasad et al. (2008a), non-explicit relations are present between successive pairs of sentences within paragraphs, but also intra-sententially between complete clauses separated by a semicolon or a colon. Our simple argument segmentation heuristic ignores intra-sentential arguments. We

believe that this accounts for its poor performance on the identification of discourse arguments.

When looking more closely at the sense labelling performance (data not shown), it seems that our network tends to overweight a few high prior probability senses, notably *EntRel* and *Expansion.Conjunction*. *EntRel* is predicted for 46% of samples, whereas it only represents 29% of the development dataset. *Expansion.Conjunction* is predicted for 24% of samples, whereas it represents only 17% of the development dataset.

We believe that one of the key issues for the Non-Explicit Discourse Relation Annotator is the size of the training set for non-explicit discourse. 17,813 samples is limited for a ConvNet, hence reducing the possible complexity of our model. The Non-Explicit Discourse Relation Annotator underperformed the best parser from CoNLL-2015 on sense labeling by 24.04% for the blind dataset, showing the advantage of non-neural network machine learning techniques when training data is scarce.

5 Conclusion and Future Work

A major area of concern in our system is the argument identification, both for explicit and non-explicit discourse relations. If we compare the results of the Supplementary task and Full Parsing task in Table 1, we can see that the Full Parsing F$_1$-scores are about half of the Supplementary task F$_1$-scores due to mis-identification of arguments.

It is necessary to consider cases where ARG1 appears in non-adjacent sentences to improve the identification of discourse arguments for explicit relations. We believe that by considering co-references in texts, we can expand our approach to address non-adjacent discourse arguments. Furthermore, it would be interesting to define new features by using ARG2 to detect what information can be added to or removed from ARG1. Finally, we believe that new ways to identify discourse arguments, such as Recurrent Neural Networks (Long Short Term Memory), could enhance the performance of the argument identification. To improve the identification of discourse arguments for non-explicit relations, we plan to expand the Explicit Discourse Argument Trimmer for non-explicit relations.

For non-explicit sense labeling, we would like to experiment with a larger training set possibly by automatically expanding it.

References

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (elus). In *Proceeding of the 2016 International Conference on Learning Representation (ICLR 2016)*, San Juan, Puerto Rico, May.

Vanessa Wei Feng, Ziheng Lin, and Graeme Hirst. 2014. The Impact of Deep Hierarchical Discourse Structures in the Evaluation of Text Coherence. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING-2014)*, Dublin, Ireland.

Sucheta Ghosh, Richard Johansson, and Sara Tonelli. 2011. Shallow discourse parsing with conditional random fields. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 1071–1079, Chiang Mai, Thailand, November.

Yangfeng Ji, Gongbo Zhang, and Jacob Eisenstein. 2015. Closing the Gap: Domain Adaptation from Explicit to Implicit Discourse Relations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 2219–2224, Lisbon, Portugal, September.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1746–1751, Doha, Qatar, October.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceeding of the 2015 International Conference on Learning Representation (ICLR 2015)*, San Diego, California.

Fang Kong, Hwee Tou 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 2014)*, pages 68–77, Doha, Qatar, October.

Majid Laali, Elnaz Davoodi, and Leila Kosseim. 2015. The CLaC Discourse Parser at CoNLL-2015. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task (CoNLL 2015)*, pages 56–60, Beijing, China, July.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20(2):151–184.

Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP 2009)*, page 683–691, Suntec, Singapore, August.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K. Joshi, and Bonnie L. Webber. 2008a. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 28–30, Marrakech, Morocco, May.

Rashmi Prasad, Eleni Miltsakaki, Nikhil Dinesh, Alan Lee, Aravind Joshi, Livio Robaldo, and Bonnie L. Webber. 2008b. The Penn Discourse Treebank 2.0 annotation manual. Technical Report IRCS-08-01, Institute for Research in Cognitive Science, University of Pennsylvania.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Attapol T. 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 (EACL 2014)*, pages 645–654, Gothenburg, Sweden, April.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

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 (CoNLL 2015)*, pages 17–24, Beijing, China, July.

Gregor Weiss. 2015. Learning representations for text-level discourse parsing. In *Proceedings of the ACL-IJCNLP 2015 Student Research Workshop*, pages 16–21, Beijing, China, July.

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

Ye Zhang and Byron Wallace. 2016. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *Computing Research Repository, abs/1510.03820.*

Biao Zhang, Jinsong Su, Deyi Xiong, Yaojie Lu, Hong Duan, and Junfeng Yao. 2015. Shallow Convolutional Neural Network for Implicit Discourse Relation Recognition. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 2230–2235, Lisbon, Portugal, September.