Joint Syntacto-Discourse Parsing and the Syntacto-Discourse Treebank [*]

Kai Zhao[†] and **Liang Huang**
School of Electrical Engineering and Computer Science
Oregon State University
Corvallis, Oregon, USA
`{kzhao.hf, liang.huang.sh}@gmail.com`

Abstract

Discourse parsing has long been treated as a stand-alone problem independent from constituency or dependency parsing. Most attempts at this problem are pipelined rather than end-to-end, sophisticated, and not self-contained: they assume gold-standard text segmentations (Elementary Discourse Units), and use external parsers for syntactic features. In this paper we propose the first end-to-end discourse parser that jointly parses in both syntax and discourse levels, as well as the first syntacto-discourse treebank by integrating the Penn Treebank with the RST Treebank. Built upon our recent span-based constituency parser, this joint syntacto-discourse parser requires no preprocessing whatsoever (such as segmentation or feature extraction), achieves the state-of-the-art end-to-end discourse parsing accuracy.

1 Introduction

Distinguishing the semantic relations between segments in a document can be greatly beneficial to many high-level NLP tasks, such as summarization (Louis et al., 2010; Yoshida et al., 2014), sentiment analysis (Voll and Taboada, 2007; Somasundaran et al., 2009; Bhatia et al., 2015), question answering (Ferrucci et al., 2010; Jansen et al., 2014), and textual quality evaluation (Tetreault et al., 2013; Li and Jurafsky, 2016).

There has been a variety of research on discourse parsing (Marcu, 2000a; Soricut and Marcu, 2003; Pardo and Nunes, 2008; Hernault et al.,

2010; da Cunha et al., 2012; Joty et al., 2013; Joty and Moschitti, 2014; Feng and Hirst, 2014; Ji and Eisenstein, 2014; Li et al., 2014a,b; Heilman and Sagae, 2015; Wang et al., 2017). But most of them suffer from the following limitations:

1. *pipelined rather than end-to-end*: they assume pre-segmented discourse, and worse yet, use gold-standard segmentations, except Hernault et al. (2010);

2. *not self-contained*: they rely on external syntactic parsers and pretrained word vectors;

3. *complicated*: they design sophisticated features, including those from parse-trees.

We argue for the first time that discourse parsing should be viewed as an extension of, and be performed in conjunction with, constituency parsing. We propose the first *joint syntacto-discourse treebank*, by unifying constituency and discourse tree representations. Based on this, we propose the first *end-to-end* incremental parser that jointly parses at both constituency and discourse levels. Our algorithm builds up on the span-based parser (Cross and Huang, 2016); it employs the strong generalization power of bi-directional LSTMs, and parses efficiently and robustly with an extremely simple span-based feature set that does not use any tree structure information.

We make the following contributions:

1. We develop a combined representation of constituency and discourse trees to facilitate parsing at both levels without explicit conversion mechanism. Using this representation, we build and release a joint treebank based on the Penn Treebank (Marcus et al., 1993) and RST Treebank (Marcu, 2000a,b) (Section 2).

2. We propose a novel joint parser that parses at both constituency and discourse levels. Our

---
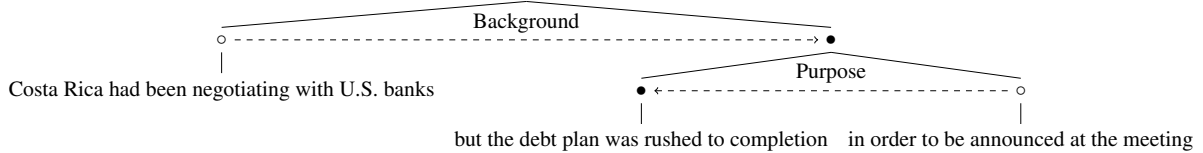
(a) A discourse tree with 3 EDUs (●: nucleas; ○: satellite) in the RST treebank (Marcu, 2000b)
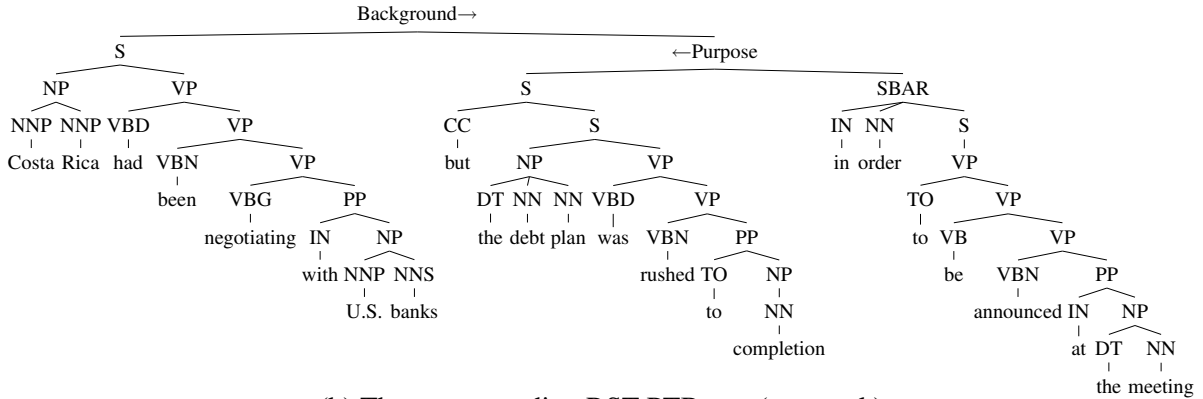
(b) The corresponding RST-PTB tree (our work)

Figure 1: Examples of the RST discourse treebank and our syntacto-discourse treebank (PTB-RST).

parser performs discourse parsing in an end-to-end manner, which greatly reduces the efforts required in preprocessing the text for segmentation and feature extraction, and, to our best knowledge, is the first end-to-end discourse parser in literature (Section 3).

3. Even though it simultaneously performs constituency parsing, our parser does *not* use any explicit syntactic feature, nor does it need any binarization of discourse trees, thanks to the powerful span-based framework of Cross and Huang (2016) (Section 3).

4. Empirically, our end-to-end parser outperforms the existing pipelined discourse parsing efforts. When the gold EDUs are provided, our parser is also competitive to other existing approaches with sophisticated features (Section 4).

2 Combined Representation & Treebank

We first briefly review the discourse structures in Rhetorical Structure Theory (Mann and Thompson, 1988), and then discuss how to unify discourse and constituency trees, which gives rise to our syntacto-discourse treebank PTB-RST.

2.1 Review: RST Discourse Structures

In an RST discourse tree, there are two types of branchings. Most of the internal tree nodes are binary branching, with one *nucleus* child containing the core semantic meaning of the current node, and one *satellite* child semantically decorating the nucleus. Like dependency labels, there is a *relation* annotated between each satellite-nucleus pair, such as "Background" or "Purpose". Figure 1(a) shows an example RST tree. There are also non-binary-branching internal nodes whose children are conjunctions, e.g., a "List" of semantically similar EDUs (which are all nucleus nodes); see Figure 2(a) for an example.

2.2 Syntacto-Discourse Representation

It is widely recognized that lower-level lexical and syntactic information can greatly help determining both the boundaries of the EDUs (i.e., discourse segmentation) (Bach et al., 2012) as well as the semantic relations between EDUs (Soricut and Marcu, 2003; Hernault et al., 2010; Joty and Moschitti, 2014; Feng and Hirst, 2014; Ji and Eisenstein, 2014; Li et al., 2014a; Heilman and Sagae, 2015). While these previous approaches rely on pre-trained tools to provide both EDU segmentation and intra-EDU syntactic parse trees, we instead propose to directly determine the low-level segmentations, the syntactic parses, and the high-level discourse parses using a single joint parser. This parser is trained on the combined trees of constituency and discourse structures.

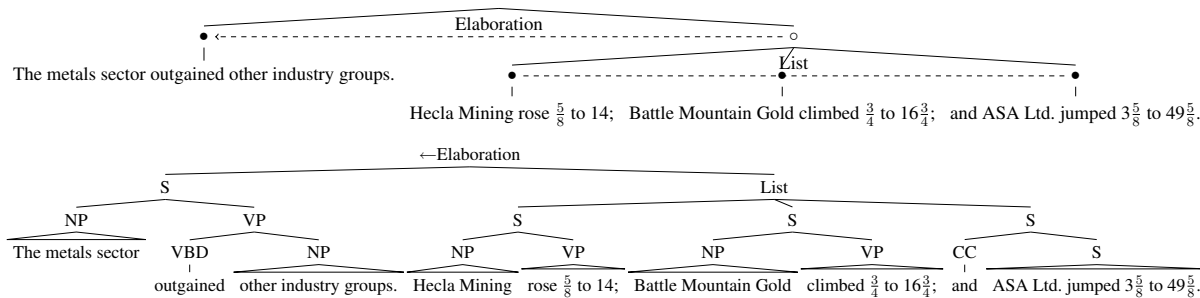We first convert an RST tree to a format similar

2118

Figure 2: Another example of RST vs. PTB-RST, demonstrating a discourse tree over two sentences and a non-binary relation (List). The lower levels of the PTB-RST tree are collapsed due to space contraints.

to those constituency trees in the Penn Treebank (Marcus et al., 1993). For each binary branching node with a nucleus child and a satellite child, we use the relation as the label of the converted parent node. The nucleus/satellite relation, along with the direction (either ← or →, pointing from satellite to nucleus) is then used as the label. For example, at the top level in Figure 2, we convert
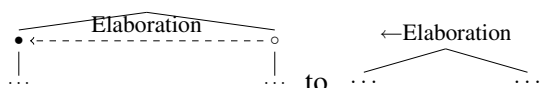
For a conjunctive branch (e.g. "List"), we simply use the relation as the label of the converted node.

After converting an RST tree into the constituency tree format, we then replace each leaf node (i.e., EDU) with the corresponding syntactic (sub)tree from PTB. Given that the sentences in the RST Treebank (Marcu, 2000b) is a subset of that of PTB, we can always find the corresponding constituency subtrees for each EDU leaf node. In most cases, each EDU corresponds to one single (sub)tree in PTB, since the discourse boundaries generally do not conflict with constituencies. In other cases, one EDU node may correspond to multiple subtrees in PTB, and for these EDUs we use the lowest common ancestor of those subtrees in the PTB as the label of that EDU in the converted tree. E.g., if C–D is one EDU in the PTB tree
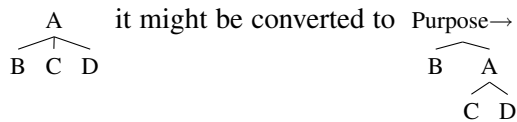
if the relation annonated in RST is B $\xrightarrow{\text{Purpose}}$ C–D.

Figures 1–2 are two examples of discourse trees and their combined syntacto-discourse trees.

2.3 Joint PTB-RST Treebank

Using the conversion strategy described above we build the first joint syntacto-discourse treebank
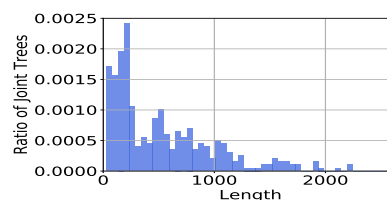

Figure 3: PTB-RST: length distribution (# tokens).

based on the Penn Treebank and RST Treebank. This PTB-RST treebank is released as a set of tools to generate the joint trees given Penn Treebank and RST Treebank data. During the alignment between the RST trees and the PTB trees, we only keep the common parts of the two trees.

We follow the standard training/testing split of the RST Treebank. In the training set, there are 347 joint trees with a total of 17,837 tokens, and the lengths of the discourses range from 30 to 2,199 tokens. In the test set, there are 38 joint trees with a total of 4,819 tokens, and the lengths vary from 45 to 2,607. Figure 3 shows the distribution of the discourse lengths over the whole dataset, which on average is about 2x of PTB sentence length, but longest ones are about 10x the longest lengths in the Treebank.

3 Joint Syntacto-Discourse Parsing

Given the combined syntacto-discourse treebank, we now propose a joint parser that can perform end-to-end discourse segmentation and parsing.

3.1 Extending Span-based Parsing

As mentioned above, the input sequences are substantially longer than PTB parsing, so we choose linear-time parsing, by adapting a popular greedy constituency parser, the span-based constituency parser of Cross and Huang (2016).

As in span-based parsing, at each step, we maintain a a stack of spans. Notice that in conventional incremental parsing, the stack stores the subtrees

input $\quad w_0 \dots w_{n-1}$

axiom $\quad \langle {}_{-1}\square_0 \rangle : (0, \emptyset) \quad$ **goal** $\quad \langle {}_{-1}\square_0\triangle_n \rangle : (\_, t)$

sh $\quad \dfrac{\langle \dots {}_i\triangle_j \rangle : (c, t)}{\langle \dots {}_i\triangle_j\square_{j+1} \rangle : (c + sc_{\mathsf{sh}}(i, j), t)} \; j < n$

comb $\quad \dfrac{\langle \dots {}_i\triangle_k\triangle_j \rangle : (c, t)}{\langle \dots {}_i\angle{k}\triangle_j \rangle : (c + sc_{\mathsf{comb}}(i, k, j), t)}$

label$_X$ $\quad \dfrac{\langle \dots {}_i\angle{k}\triangle_j \rangle : (c, t)}{\langle \dots {}_i\triangle_j \rangle : (c + sc_{\mathsf{label}_X}(i, k, j), t \cup \{{}_iX_j\})}$

nolabel $\quad \dfrac{\langle \dots {}_i\angle{k}\triangle_j \rangle : (c, t)}{\langle \dots {}_i\triangle_j \rangle : (c + sc_{\mathsf{nolabel}}(i, k, j), t)}$

Figure 4: Deductive system for joint syntactic and discourse parsing. $sc_{\mathsf{sh}}(\cdot, \cdot)$, $sc_{\mathsf{comb}}(\cdot, \cdot, \cdot)$, $sc_{\mathsf{label}_X}(\cdot, \cdot, \cdot)$, and $sc_{\mathsf{nolabel}}(\cdot, \cdot, \cdot)$ are scoring functions evaluated in the neural network.

constructed so far, but in span-based constituency parsing, the stack only stores the boundaries of subtrees, which are just a list of indices $\dots {}_i\triangle_k\triangle_j$. In other words, quite shockingly, no tree structure is represented anywhere in the parser. Please refer Cross and Huang (2016) for details.

Similar to span-based constituency parsing, we alternate between structural (either shift or combine) and label (label$_X$ or nolabel) actions in an odd-even fashion. But different from Cross and Huang (2016), after a structural action, we choose to keep the last branching point k, i.e., ${}_i\angle{k}\triangle_j$ (mostly for combine, but also trivially for shift). This is because in our parsing mechanism, the discourse relation between two EDUs is actually determined after the previous combine action. We need to keep the splitting point to clearly find the spans of the two EDUs to determine their relations. This midpoint k disappears after a label action; therefore we can use the shape of the last span on the stack (whether it contains the split point, i.e., ${}_i\angle{k}\triangle_j$ or ${}_i\triangle_j$) to determine the parity of the step and thus no longer need to carry the step z in the state as in Cross and Huang (2016).

The nolabel action makes the binarization of the discourse/constituency tree unnecessary, because nolabel actually combines the top two spans on the stack σ into one span, but without annotating the new span a label. This greatly simplifies the pre-processing and post-processing efforts needed.

	Prec.	Recall	F1
Constituency	87.6	86.9	87.2
Discourse	46.5	40.2	43.0
Overall	83.5	81.6	82.5

Table 1: Accuracies on PTB-RST at constituency and discourse levels.

3.2 Recurrent Neural Models and Training

The scoring functions in the deductive system (Figure 4) are calculated by an underlying neural model, which is similar to the bi-directional LSTM model in Cross and Huang (2016) that evaluates based on span boundary features. Again, it is important to note that no discourse or syntactic tree structures are represented in the features.

During the decoding time, a document is firstl passed into a two-layer bi-directional LSTM model, then the outputs at each text position of the two layers of the bi-directional LSTMs are concatenated as the positional features. The spans at each parsing step can be represented as the feature vectors at the boundaries. The span features are then passed into fully connected networks with softmax to calculate the likelihood of performing the corresponding action or marking the corresponding label.

We use the "training with exploration" strategy (Goldberg and Nivre, 2013) and the dynamic oracle mechanism described in Cross and Huang (2016) to make sure the model can handle unseen parsing configurations properly.

4 Empirical Results

We use the treebank described in Section 2 for empirical evaluation. We randomly choose 30 documents from the training set as the development set.

We tune the hyperparameters of the neural model on the development set. For most of the hyperparameters we settle with the same values suggested by Cross and Huang (2016). To alleviate the overfitting problem for training on the relative small RST Treebank, we use a dropout of 0.5.

One particular hyperparameter is that we use a value β to balance the chances between training following the exploration (i.e., the best action chosen by the neural model) and following the correct path provided by the dynamic oracle. We find that $\beta = 0.8$, i.e., following the dynamic oracle with a probability of 0.8, achieves the best performance. One explanation for this high chance to follow the oracle is that, since our combined trees are signif-

	description	syntactic feats.	segmentation	structure	+nuclearity	+relation
Bach et al. (2012)	segmentation only	Stanford	95.1	-	-	-
Hernault et al. (2010)	end-to-end pipeline	Penn Treebank	94.0	72.3	59.1	47.3
joint syntactic & discourse parsing		-	**95.4**	**78.8**	**65.0**	**52.2**

Table 2: F1 scores of end-to-end systems. "+nuclearity" indicates scoring of tree structures with nuclearity included. "+relation" has both nuclearity and relation included (e.g., ←Elaboration).

		syntactic feats	structure	+nuclearity	+relation
human annotation (Ji and Eisenstein, 2014)		-	88.7	77.7	65.8
sparse	Hernault et al. (2010)	Penn Treebank	83.0	68.4	54.8
	Joty et al. (2013)	Charniak (retrained)	82.7	68.4	55.7
	Joty and Moschitti (2014)	Charniak (retrained)	-	-	57.3
	Feng and Hirst (2014)	Stanford	85.7	71.0	58.2
	Heilman and Sagae (2015)	ZPar (retraied)	83.5	68.1	55.1
	Wang et al. (2017)	Stanford	**86.0**	**72.4**	59.7
neural	Li et al. (2014a)	Stanford	82.4	69.2	56.8
	+ sparse features		84.0	70.8	58.6
	Ji and Eisenstein (2014)	MALT	80.5	68.6	58.3
	+ sparse features		81.6	71.1	**61.8**
span-based discourse parsing		-	84.2	67.7	56.0

Table 3: Experiments using gold segmentations. The column of "syntactic feats" shows how the syntactic features are calculated in the corresponding systems. Note that our parser predicts solely based on the span features from bi-directionaly LSTM, instead of any explicitly designed syntactic features.

icantly larger than the constituency trees in Penn Treebank, lower β makes the parsing easily divert into wrong trails that are difficult to learn from.

Since our parser essentially performs both constituency parsing task and discourse parsing task. We also evaluate the performances on sentence constituency level and discourse level separately. The result is shown in Table 1. Note that in constituency level, the accuracy is not directly comparable with the accuracy reported in Cross and Huang (2016), since: a) our parser is trained on a much smaller dataset (RST Treebank is about 1/6 of Penn Treebank); b) the parser is trained to optimize the discourse-level accuracy.

Table 2 shows that, in the perspective of end-to-end discourse parsing, our parser first outperforms the state-of-the-art segmentator of Bach et al. (2012), and furthermore, in end-to-end parsing, the superiority of our parser is more pronounced comparing to the previously best parser of Hernault et al. (2010).

On the other hand, the majority of the conventional discourse parsers are not end-to-end: they rely on gold EDU segmentations and pre-trained tools like Stanford parsers to generate features. We perform an experiment to compare the per-

formance of our parser with them given the gold EDU segments (Table 3). Note that most of these parsers do not handle multi-branching discourse nodes and are trained and evaluated on binarized discourse trees (Feng and Hirst, 2014; Li et al., 2014a,b; Ji and Eisenstein, 2014; Heilman and Sagae, 2015), so their performances are actually not directly comparable to the results we reported.

5 Conclusion

We have presented a neural-based incremental parser that can jointly parse at both constituency and discourse levels. To our best knowledge, this is the first end-to-end parser for discourse parsing task. Our parser achieves the state-of-the-art performance in end-to-end parsing, and unlike previous approaches, needs little pre-processing effort.

References

Ngo Xuan Bach, Nguyen Le Minh, and Akira Shimazu. 2012. A reranking model for discourse segmentation using subtree features. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 160–168. Association for Computational Linguistics.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from rst discourse parsing. *arXiv preprint arXiv:1509.01599*.

James Cross and Liang Huang. 2016. Incremental parsing with minimal features using bi-directional lstm. *arXiv preprint arXiv:1606.06406*.

Iria da Cunha, Eric SanJuan, Juan-Manuel Torres-Moreno, M Teresa Cabré, and Gerardo Sierra. 2012. A symbolic approach for automatic detection of nuclearity and rhetorical relations among intra-sentence discourse segments in spanish. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 462–474. Springer.

Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *ACL (1)*, pages 511–521.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.

Yoav Goldberg and Joakim Nivre. 2013. Training deterministic parsers with non-deterministic oracles. *Transactions of the association for Computational Linguistics*, 1:403–414.

Michael Heilman and Kenji Sagae. 2015. Fast rhetorical structure theory discourse parsing. *arXiv preprint arXiv:1505.02425*.

Hugo Hernault, Helmut Prendinger, David A DuVerle, Mitsuru Ishizuka, and Tim Paek. 2010. Hilda: a discourse parser using support vector machine classification. *Dialogue and Discourse*, 1(3):1–33.

Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for nonfactoid answer reranking. In *ACL (1)*, pages 977–986.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *ACL (1)*, pages 13–24.

Shafiq Joty and Alessandro Moschitti. 2014. Discriminative reranking of discourse parses using tree kernels. *a) A*, 4(5):6.

Shafiq R Joty, Giuseppe Carenini, Raymond T Ng, and Yashar Mehdad. 2013. Combining intra-and multi-sentential rhetorical parsing for document-level discourse analysis. In *ACL (1)*, pages 486–496.

Jiwei Li and Dan Jurafsky. 2016. Neural net models for open-domain discourse coherence. *arXiv preprint arXiv:1606.01545*.

Jiwei Li, Rumeng Li, and Eduard H Hovy. 2014a. Recursive deep models for discourse parsing. In *EMNLP*, pages 2061–2069.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014b. Text-level discourse dependency parsing. In *ACL (1)*, pages 25–35.

Annie Louis, Aravind Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summarization. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 147–156. Association for Computational Linguistics.

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

Daniel Marcu. 2000a. The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational linguistics*, 26(3):395–448.

Daniel Marcu. 2000b. *The theory and practice of discourse parsing and summarization*. MIT press.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Thiago Alexandre Salgueiro Pardo and Maria das Graças Volpe Nunes. 2008. On the development and evaluation of a brazilian portuguese discourse parser. *Revista de Informática Teórica e Aplicada*, 15(2):43–64.

Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. 2009. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 170–179. Association for Computational Linguistics.

Radu Soricut and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 149–156. Association for Computational Linguistics.

ETS Tetreault et al. 2013. Holistic discourse coherence annotation for noisy essay writing.

Kimberly Voll and Maite Taboada. 2007. Not all words are created equal: Extracting semantic orientation as a function of adjective relevance. In *Australasian Joint Conference on Artificial Intelligence*, pages 337–346. Springer.

Yizhong Wang, Sujian Li, and Houfeng Wang. 2017. A two-stage parsing method for text-level discourse analysis. In *Proceedings of ACL*.

Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. 2014. Dependency-based discourse parser for single-document summarization. In *EMNLP*, pages 1834–1839. Citeseer.