A Two-Stage Parsing Method for Text-Level Discourse Analysis

Yizhong Wang **Sujian Li** **Houfeng Wang**
Key Laboratory of Computational Linguistics, Peking University, MOE, China
{yizhong, lisujian, wanghf}@pku.edu.cn

Abstract

Previous work introduced transition-based algorithms to form a unified architecture of parsing rhetorical structures (including span, nuclearity and relation), but did not achieve satisfactory performance. In this paper, we propose that transition-based model is more appropriate for parsing the naked discourse tree (i.e., identifying span and nuclearity) due to data sparsity. At the same time, we argue that relation labeling can benefit from naked tree structure and should be treated elaborately with consideration of three kinds of relations including within-sentence, across-sentence and across-paragraph relations. Thus, we design a pipelined two-stage parsing method for generating an RST tree from text. Experimental results show that our method achieves state-of-the-art performance, especially on span and nuclearity identification.

1 Introduction

A typical document is usually organized in a coherent way that each text unit is relevant to its context and plays a role in the entire semantics. Text-level discourse analysis tries to identify such discourse structure of a document and its success can benefit many downstream tasks, such as sentiment analysis (Polanyi and van den Berg, 2011) and document summarization (Louis et al., 2010).

One most influential text-level discourse parsing theory is Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), under which a text is parsed to a hierarchical discourse tree. The leaf nodes of this tree correspond to Elementary Discourse Units (EDUs, usually clauses) and then leaf nodes are recursively connected by rhetorical relations to form larger text spans until the final tree is built. RST also depicts which part is more important in a relation by tagging *Nucleus* or *Satellite*. Generally, each relation at least includes a *Nucleus* and there are three nuclearity types: *Nucleus-Satellite (NS)*, *Satellite-Nucleus (SN)* and *Nucleus-Nucleus (NN)*. Therefore, the performance of RST discourse parsing can be evaluated from three aspects: span, nuclearity and relation.

To parse discourse trees, transition-based parsing model, which gains significant success in dependency parsing (Yamada and Matsumoto, 2003; Nivre et al., 2006) , was introduced to discourse analysis. Marcu (1999) first employed a transition system to derive a discourse parse tree. In such a system, action labels are designed by combining shift-reduce action with nuclearity and relation labels, so that one classifier can determine span, nuclearity and relation simultaneously via judging actions. More recent studies followed this research line and enhanced the performance by either tuning the models (Sagae, 2009) or using more effective features (Ji and Eisenstein, 2014; Heilman and Sagae, 2015). Though these transition-based models show advantages in the unified processing of span, nuclearity and relation, they report weaker performance than other methods, like CYK-like algorithms (Li et al., 2014, 2016) or greedy bottom-up algorithms that merge adjacent spans (Hernault et al., 2010; Feng and Hirst, 2014).

In such cases, we analyze that the labelled data can not sufficiently support the classifier to distinguish among the information-rich actions (e.g., Reduce-NS-Contrast) , since there exist very few labelled text-level discourse corpus available for training. The limited training data will cause unbalanced actions and lead to the problems of data sparsity and overfitting. Thus, we propose to use the transition-based model to parse a naked dis-

course tree (i.e., identifying span and nuclearity) in the first stage. The benefits are three-fold. First, we can still use the transition based model which is a good tree construction tool. Second, much fewer actions need to be identified in the tree construction process. Third, we could separately label relations, which needs careful consideration.

In the second stage, relation labels for each span are determined independently. Prior studies (Joty et al., 2013; Feng and Hirst, 2014) have found that rhetorical relations distribute differently intra-sententially vs. multi-sententially. They discriminate the two levels by training two models with different feature sets. We take a further step and argue that relations between paragraphs are usually more loosely connected than those between sentences within the same paragraph. Therefore we train three separate classifiers for labeling relations at three levels: within-sentence, across-sentence and across-paragraph. Different features are used for each classifier and the naked tree structure generated in the first stage is also leveraged as features. Experiments on the RST-DT corpus demonstrate the effectiveness of our pipelined two-stage discourse parsing model.

2 Our Method

Our discourse parsing process is composed of two stages: tree structure construction and relation labeling. In this work, we follow the convention to use the gold standard EDU segmentations and focus on building a tree with nuclearity and relation labels assigned for each inner node.

2.1 Tree Structure Construction

In a typical transition-based system for discourse parsing, the parsing process is modeled as a sequence of *shift* and *reduce* actions, which are applied to a stack and a queue. The stack is initialized to be empty and the queue contains all EDUs in the document. At each step, the parser performs either *shift* or *reduce*. *Shift* pushes the first EDU in the queue to the top of the stack, while *reduce* pops and merges the top elements in the stack to get a new subtree, which is then pushed back to the top of the stack. A parse tree can be finally constructed until the queue is empty and the stack only contains the complete tree. Only one classifier is learned to judge the actions at each step.

To derive a discourse tree in a unified framework, prior systems design multiple reduce actions with consideration of both nuclearity and relation types. With 3 nuclearity types and 18 relation types, the number of reduce actions exceeds 40, leading to the data sparsity problem.

In our parsing model, a transition-based system is responsible for building the naked tree without relation labels. We only design four types of actions, including: Shift, Reduce-NN, Reduce-NS, Reduce-SN. We identify span and nuclearity simultaneously in the transition-based tree construction, since nuclearity is actually closely related to the tree structure, just as the left-arc and right-arc action in dependency parsing. The number of the four actions on the training set of RST-DT corpus is shown in Table 1. Though the four actions still have an unbalanced distribution, the relatively large number of occurrences assures that the classifier in our system can be trained more sufficiently.

Shift	Reduce-NN	Reduce-NS	Reduce-SN
19443	4329	11702	3065

Table 1: Statistics of the Four Actions

2.2 Relation Labeling

The most challenging subtask of discourse parsing is relation labeling. In a binarized RST discourse tree, a relation label can be determined for each internal node, describing the relation between its left and right subtrees[1].

We conduct relation labeling after the naked tree structure has been constructed. On one hand, the naked tree structure can provide more information to support relation classification, verified in (Feng and Hirst, 2014). For example, some relations tend to appear around the tree root while other relaitons would like to keep away from the root. On the other hand, we can elaborately distinguish relations at different levels, including within-sentence, across-sentence, across-paragraph. We add across-paragraph level because some relations, like textual-organization and topic-change are observed to mainly occur between paragraphs.

Therefore, we adopt three classifiers for labeling relations at different levels. We first traverse the naked tree in post order and ignore leaf nodes, since we only need to judge relations for internal nodes. Next, for each internal node, we determine

[1]Relation label is actually assigned to the satellite subtree and a "Span" label is assigned to the nucleus substree.

whether its left and right subtrees are in different paragraphs, or the same paragraph, or the same sentence. For each level, we predict a relation label using the corresponding classifier.

2.3 Training

We use SVM classifiers for the four classification tasks (one action classifier and three relation classifiers). We take the linear kernel for fast training and use squared hinge loss with L_1 penalty on the error term. The penalty coefficient C is set to 1.

The four classifiers are learned with offline training. Training instances for the action classifier are generated by converting gold parse trees into a sequence of actions. Then we extract features for each action before it is performed. Training instances for relation classifiers are prepared by traversing the gold parse trees and extracting features for the relation of each internal node.

3 Features

This section details the features used in our model, which are a key to the four classifiers in discourse parsing.

For the action classifier, features are extracted from the top 2 elements S_1, S_2 in the stack and the first EDU Q_1 in the queue. We design the feature sets for the action classifier as follows:

- **Status features:** the previous action; number of elements in the stack and queue.
- **Position features:** whether S_1, S_2 or S_1, Q_1 are in the same sentence or paragraph; whether they are start or end of a sentence, paragraph or document; distance from S_1, S_2, Q_1 to the start and end of document.
- **Structural features:** nuclearity type (NN, NS or SN) of S_1, S_2; number of EDUs and sentences in S_1, S_2; length comparison of S_1, S_2 with respect to EDUs and sentences.
- **Dependency features:** whether dependency relations exist between S_1, S_2 or between S_1, Q_1; the dependency direction and relation type.
- **N-gram features:** the first and the last n words and their POS tags in the text of S_1, S_2, Q_1, where $n \in \{1, 2\}$.
- **Nucleus features:** the dependency heads of the nucleus EDUs[2] for S_1, S_2, Q_1 and their POS tags; brown clusters (Brown et al., 1992; Turian et al., 2010) of all the words in the nucleus EDUs of S_1, S_2, Q_1.

Next, we list all the features used for the three relation classifiers. Given an internal node P in the naked tree, we aim to predict the relation between its left child C_{left} and right child C_{right}. Dependency features, N-gram features and nucleus features discussed above are also needed, the only difference is that these features are applied to the left and right children. Other features include:

- **Refined Structural features:** nuclearity type of node P; distance from P, C_{left}, C_{right} to the start and end of the document / paragraph / sentence with respect to paragraphs / sentences / EDUs; number of paragraphs / sentences / EDUs in C_{left} and C_{right}; length comparison of C_{left} and C_{right} with respect to paragraphs / sentences / EDUs.
- **Tree features:** depth and height of the node P in the tree; nuclearity type of P and P's grandparent node, if they exist. This feature type benefits from our stagewise parsing method.

Relation labeling classifiers at different levels pick somewhat different features from all the features. N-gram and structural features work for the three classifiers. Dependency features are only used for within-sentence classifier. Nucleus features and tree features are only used for across-sentence and across-paragraph classifiers.

4 Experiments

We evaluate our parser on RST Discourse Treebank (RST-DT) (Carlson et al., 2003) and thoroughly analyze different components of our method. Results show our parsing model achieves state-of-the-art performance on the text-level discourse parsing task.

4.1 Setup

RST-DT annotates 385 documents (347 for training and 38 for testing) from the Wall Street Journal using Rhetorical Structure Theory (Mann and Thompson, 1988). Conventionally, we use 18 coarse-grained relations and binarize non-binary relations with right-branching (Sagae and Lavie, 2005). For preprocessing, we use the Stanford CoreNLP toolkit (Manning et al., 2014) to lemmatize words, get POS tags, segment sentences and syntactically parse them.

To directly compare with other discourse parsing systems, we employ the same evaluation met-

[2]Nucleus EDU is defined by recursively selecting the Nucleus in the binary tree until an EDU (leaf node) is reached.

rics, i.e. the precision, recall and F-score [3] with respect to span (S), nuclearity (N) and relation (R), as defined by Marcu (2000).

4.2 Results and Analysis

We compare our system against other state-of-the-art discourse parsers, shown in Table 2. Among them, Joty et al. (2013), Li et al. (2014) and Li et al. (2016) all employ CKY-like algorithms to search global optimal parsing result. Ji and Eisenstein (2014) and Heilman and Sagae (2015) use transition-based parsing systems with improvements on the feature representation. Feng and Hirst (2014) adopts a greedy approach that merges two adjacent spans at each step and two CRFs are used to predict the structure and the relation separately.

From Table 2, we can see that our method outperforms all the others with respect to span and nuclearity, and exceeds most systems on relation labeling. Especially, our method significantly outperforms other transition-based models (Ji and Eisenstein, 2014; Heilman and Sagae, 2015) on building the naked tree structure (span and nuclearity). This is mainly due to the proper design of actions in our transition-based system. The reason that Ji and Eisenstein (2014) achieve a high score of relation labeling may be that their latent representations are more advantageous in capturing semantics, which will inspire us to refine our features in future work.

Model	S	N	R
Joty et al. (2013)	82.7	68.4	55.7
Li et al. (2014)	84.0	70.8	58.6
Ji and Eisenstein (2014)	82.1	71.1	**61.6**
Feng and Hirst (2014)	85.7	71.0	58.2
Heilman and Sagae (2015)	83.5	69.3	57.4
Li et al. (2016)	85.8	71.1	58.9
Ours	**86.0**	**72.4**	59.7
Human [4]	88.7	77.7	65.8

Table 2: Performance comparison with state-of-the-art parsers.

To further explore the influence of different components in our model, we implement three simplified versions (i.e., *Simp*-1/2/3), as is shown in Table 3. **Stage** means whether two-stage strategy is adopted, **Level** denotes whether three kinds of relations (i.e., within-sentence, across-sentence, and across-paragraph) are differently classified, and **Tree** represents whether relation labeling uses tree features generated in the first stage.

The simplest model *Simp*-1 is almost the same as (Heilman and Sagae, 2015) except that we employ more features. That *Simp*-1 has a high performance also means that transition-based method has potentials for constructing discourse trees. *Simp*-2 adopts the two-stage strategy, but uses only one classifier to classify all the relations. We can observe that the pipelined two stages bring a significant improvement with respect to all the aspects, compared to *Simp*-1. The difference between *Simp*-3 and Ours is that *Simp*-3 does not exploit the tree structure features generated in the first stage. We can see that the three-level relation classification and tree features together bring an improvement of about 1 percent on relation labeling. Compared with prior work, this slight improvement is also valuable and more efficacious features need to be explored.

Model	**Stage**	**Level**	**Tree**	S	N	R
Simp-1	No	No	No	84.4	70.7	57.7
Simp-2	Yes	No	No	86.0	72.4	58.6
Simp-3	Yes	Yes	No	86.0	72.4	59.4
Ours	Yes	Yes	Yes	86.0	72.4	59.7

Table 3: Comparison with simplified versions.

Though the three-level relation labeling does not achieve prominent improvement, we get some interesting results via analyzing the performance on each relation. The *Attribution* and *Same-Unit* relations are the top 2 relations that we successfully classify with F-score as 0.87 and 0.83 respectively and over 90 percent of these two relations occur within sentences. This means that within-sentence relations are relatively easy to cope with. We also compare our final model with *Simp*-1 and results show that the Textual-Organization and Topic-Comment relaitons gain an increase by 20% and 8% respectively. Most of the Textual-Organization and Topic-Comment relations are loosely across paragraphs and their numbers (i.e., 148 and 130 instances in training data) are also relatively small. We can see that our method can improve on predicting infrequent relations and partly solve the data sparsity problem. At the same time, we infer that relations indeed belong to different levels and deserve fine treatment.

[3] Precision, recall and F-score are the same when manual segmentation is used.

[4] The human agreement on the annotations of RST corpus

5 Conclusion

In this paper, we design a novel two-stage method for text-level discourse analysis. The first stage adopts the transition-based algorithm to construct naked trees with consideration of span and nuclearity. The second stage categorizes relations into three levels and uses three classifiers for relation labeling. This pipelined design can mitigate the data sparsity problem in tree construction, and provide a new view of elaborately treating relations. Comprehensive experiments show the effectiveness of our proposed method.

Acknowledgments

We thank the anonymous reviewers for their insightful comments on this paper. This work was partially supported by National Natural Science Foundation of China (61572049 and 61333018). The correspondence author of this paper is Sujian Li.

References

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics* 18(4):467–479.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current and new directions in discourse and dialogue*, Springer, pages 85–112.

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

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.

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

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

Qi Li, Tianshi Li, and Baobao Chang. 2016. Discourse parsing with attention-based hierarchical neural networks pages 362–371.

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*. Association for Computational Linguistics, pages 147–156.

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.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60.

Daniel Marcu. 1999. A decision-based approach to rhetorical parsing. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, pages 365–372.

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

Joakim Nivre, Johan Hall, Jens Nilsson, Gülşen Eryiit, and Svetoslav Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pages 221–225.

Livia Polanyi and Martin van den Berg. 2011. Discourse structure and sentiment. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. IEEE, pages 97–102.

Kenji Sagae. 2009. Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. In *Proceedings of the 11th International Conference on Parsing Technologies*. Association for Computational Linguistics, pages 81–84.

Kenji Sagae and Alon Lavie. 2005. A classifier-based parser with linear run-time complexity. In *Proceedings of the Ninth International Workshop on Parsing Technology*. Association for Computational Linguistics, pages 125–132.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *ACL*. Association for Computational Linguistics, pages 384–394.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*. volume 3, pages 195–206.