Modeling Discourse Cohesion for Discourse Parsing via Memory Network

Yanyan Jia, Yuan Ye, Yansong Feng, Yuxuan Lai, Rui Yan and **Dongyan Zhao**
Institute of Computer Science and Technology, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{jiayanyan,pkuyeyuan,fengyansong,erutan,ruiyan,zhaody}@pku.edu.cn

Abstract

Identifying long-span dependencies between discourse units is crucial to improve discourse parsing performance. Most existing approaches design sophisticated features or exploit various off-the-shelf tools, but achieve little success. In this paper, we propose a new transition-based discourse parser that makes use of memory networks to take discourse cohesion into account. The automatically captured discourse cohesion benefits discourse parsing, especially for long span scenarios. Experiments on the RST discourse treebank show that our method outperforms traditional featured based methods, and the memory based discourse cohesion can improve the overall parsing performance significantly [1].

1 Introduction

Discourse parsing aims to identify the structure and relationship between different element discourse units (EDUs). As a fundamental topic in natural language processing, discourse parsing can assist many down-stream applications such as summarization (Louis et al., 2010), sentiment analysis (Polanyi and van den Berg, 2011) and question-answering (Ferrucci et al., 2010). However, the performance of discourse parsing is still far from perfect, especially for EDUs that are distant to each other in the discourse. In fact, as found in (Jia et al., 2018), the discourse parsing performance drops quickly as the dependency span increases. The reason may be twofold:

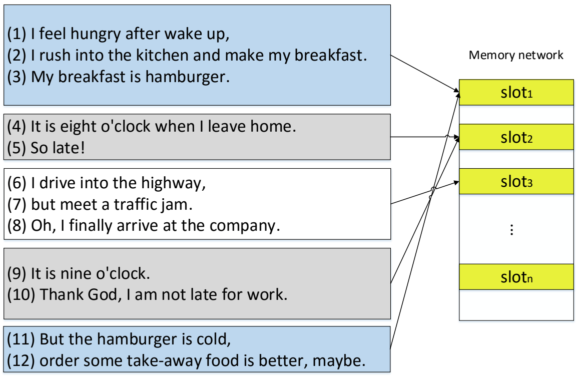
Firstly, as discussed in previous works (Joty et al., 2013), it is important to address discourse structure characteristics, e.g., through modeling lexical chains in a discourse, for discourse parsing, especially in dealing with long span scenarios. However, most existing approaches mainly focus on studying the semantic and syntactic aspects of EDU pairs, in a more local view. Discourse cohesion reflects the syntactic or semantic relationship between words or phrases in a discourse, and, to some extent, can indicate the topic changing or threads in a discourse. Discourse cohesion includes five situations, including reference, substitution, ellipsis, conjunction and lexical cohesion (Halliday and Hasan, 1989). Here, lexical cohesion reflects the semantic relationship of words, and can be modeled as the recurrence of words, synonym and contextual words.

However, previous works do not well model the discourse cohesion within the discourse parsing task, or do not even take this issue into account. Morris and Hirst (1991) proposes to utilize Roget thesauri to form lexical chains (sequences of semantically related words that can reflect the topic shifts within a discourse), which are used to extract features to characterize discourse structures. (Joty et al., 2013) uses lexical chain feature to model multi-sentential relation. Actually, these simplified cohesion features can already improve parsing performance, especially in long spans.

Secondly, in modern neural network methods, modeling discourse cohesion as part of the networks is not a trivial task. One can still use off-the-shell tools to obtain lexical chains, but these tools can not be jointly optimized with the main neural network parser. We argue that characterizing discourse cohesion implicitly within a unified framework would be more

[1] Code for replicating our experiments is available at https://github.com/PKUYeYuan/ACL2018_CFDP.

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 438–443
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics



(1) I feel hungry after wake up,
(2) I rush into the kitchen and make my breakfast.
(3) My breakfast is hamburger.

(4) It is eight o'clock when I leave home.
(5) So late!

(6) I drive into the highway,
(7) but meet a traffic jam.
(8) Oh, I finally arrive at the company.

(9) It is nine o'clock.
(10) Thank God, I am not late for work.

(11) But the hamburger is cold,
(12) order some take-away food is better, maybe.

Figure 1: An illustration for modelling discourse cohesion with memory network. The example discourse includes 12 EDUs and talks about 3 different threads (food, time and traffic), which are colored by blue, gray and white, respectively.

straightforward and effective for our neural network based parser. As shown in Figure 1, the 12 EDUs in the given discourse talk about different topics, marked with 3 different colors, which could be captured by a memory network that maintains several memory slots. In discourse parsing, such an architecture may help to cluster topically similar or related EDUs into the same memory slot, and each slot could be considered as a representation that maintains a specific topic or thread within the current discourse. Intuitively, we could also treat such a mechanism as a way to capture the cohesion characteristics of the discourse, just like the lexical chain features used in previous works, but without relying on external tools or resources.

In this paper, we investigate how to exploit discourse cohesion to improve discourse parsing. Our contribution includes: 1) we design a memory network method to capture discourse cohesion implicitly in order to improve discourse parsing. 2) We choose bidirectional long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) with an attention mechanism to represent EDUs directly from embeddings, and use simple position features to capture shallow discourse structures, without relying on off-the-shelf tools or resources. Experiments on the RST corpus show that the memory based discourse cohesion model can help better capture discourse structure information and lead to significant improvement over traditional feature based discourse parsing methods.

2 Model overview

Our parser is an arc-eager style transition system (Nivre, 2003) with 2 stacks and a queue as shown in Figure 2, which is similar in spirit with (Dyer et al., 2015; Ballesteros et al., 2015). We follow the conventional data structures in transition-based dependency parsing, i.e., a queue (B) of EDUs to be processed, a stack (S) to store the partially constructed discourse trees, and a stack (A) to represent the history of transitions (actions combined with discourse relations).

In our parser, the transition actions include *Shift*, *Reduce*, *Left-arc* and *Right-arc*. At each step, the parser chooses to take one of the four actions and pushes the selected transition into A. *Shift* pushes the first EDU in queue B to the top of the stack S, while *Reduce* pops the top item of S. *Left-arc* connects the first EDU (head) in B to the top EDU (dependent) in S and then pops the top item of S, while *Right-arc* connects the top EDU (head) of S to the first EDU (dependent) in B and then pushes B's first EDU to the top of S. A parse tree can be finally constructed until B is empty and S only contains a complete discourse tree. For more details, please refer to (Nivre, 2003).

As shown in Figure 2, at time t, we characterize the current parsing process by preserving the top two elements in B, top three elements in A and the root EDU in the partially constructed tree at the top of S. We first concatenate the embeddings of the preserved elements in each data structure to obtain the embeddings of S, B and A. We then append the three representations with the $position_2$ features (introduced in Section 2.1), respectively. We pass them through one ReLU layer and two fully connected layers with ReLU as their activation functions to obtain the final state representation p_t at time t, which will be used to determine the best transition to take at t.

Next, we apply an affine transformation to p_t and feed it to a softmax layer to get the distribution over all possible decisions (actions combined with discourse relations). We train our model using the automatically generated oracle action sequences as the gold-standard annotations, and utilize cross entropy as the loss function. We perform greedy search during decoding.

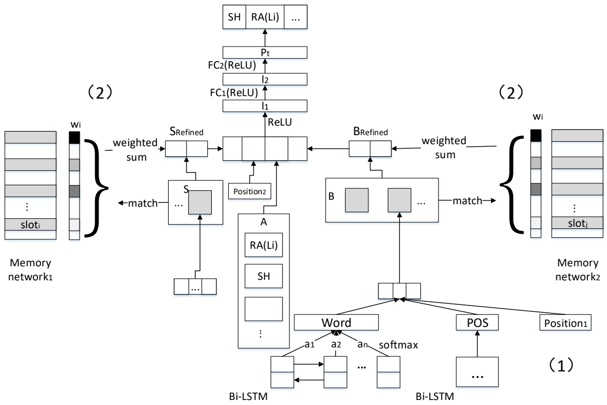


Figure 2: Our discourse parsing framework: (1) Basic EDU representation module; (2) Memory networks to capture the discourse cohesion so as to obtain the refined representations of S and B. RA(Li) means that the chosen action is *Right-arc* and its relation is *List*. SH means *Shift*. a_1 to a_n are weights for the attention mechanism of the bidirectional LSTM.

2.1 Discourse Structures

As mentioned in previous work (Jia et al., 2018), when the top EDUs in S and B are far from each other in the discourse, i.e., with a long span, the parser will be prone to making wrong decisions. To deal with these long-span cases, one should take discourse structures into account, e.g., extracting features from the structure of a long discourse or analyzing and characterizing different topics discussed in the discourse.

We, therefore, choose two kinds of position features to reflect the structure information, which can be viewed as a shallow form of discourse cohesion. The first one describes the position of an EDU alone, while the second represents the spatial relationship between the top EDUs of S and B. (1) $Position_1$: the positions of the EDU in the sentence, paragraph and discourse, respectively. (2) $Position_2$: whether the top EDUs of S and B are in the same sentence/paragraph or not, and the distance between them.

3 Memory based Discourse Cohesion

Basic EDU representation: In our model, the EDUs in both S and B follow the same representation method, and we take an EDU in B as an example as shown in Figure 2. The basic representation for an EDU is built by concatenating three components, i.e., *word*, *POS* and $Position_1$. Regarding *word*, we feed the sequence of words in the EDU to a bi-directional Long Short Term Memory (LSTM) with attention mechanism and obtain the final *word* representation by concatenating the two final outputs from both directions. Here, we use pre-trained Glove (Pennington et al., 2014) as the word embeddings. We get the POS tags from Stanford CoreNLP toolkit (Manning et al., 2014), and similarly, send the POS tag sequence of the EDU to a bi-directional LSTM with attention mechanism to obtain the final *POS* representation. For concise, we omit the bi-directional LSTM network structure for *POS* in Figure 2, which is the same as the one for *word*. The $Position_1$ feature vectors are randomly initialized and we expect them to work as a proxy to capture the shallow discourse structure information.

Memory Refined Representation: Besides the shallow structure features, we design a memory network component to cluster EDUs with similar topics to the same memory slot to alleviate the long span issues, as illustrated in Figure 1. We expect these memory slots can work as lexical chains, which can maintain different threads within the discourse. Such a memory mechanism has the advantage that it can perform the clustering automatically and does not rely on extra tools or resources to train.

Concretely, we match the representations of S and B with their corresponding memory networks, respectively, to get their discourse cohesion clues, which are used to improve the original representations. Take B as an example, we first compute the similarity between the representation of B (V_b) and each memory slot m_i in B's memory. We adopt the cosine similarity as our metric as below:

$$Sim[x, y] = \frac{x \cdot y}{\|x\| \cdot \|y\|} \tag{1}$$

Then, we use this cosine similarity to produce a normalized weight w_i for each memory slot. We introduce a strength factor λ to improve the focus.

$$w_i = \frac{\exp(\lambda Sim[V_b, m_i])}{\sum_j \exp(\lambda Sim[V_b, m_j])} \tag{2}$$

Finally, we get the discourse cohesion clue of B (denoted by B_{Coh}) from its memory according to the weighted sum of m_i.

$$B_{Coh} = \sum_i w_i m_i \tag{3}$$

We concatenate B_{Coh} (the discourse cohesion clue of B) and the original embedding of B to get the refined representation $B_{refined}$ for B. Similarly, we concatenate S_{Coh} and the embedding of S to get the refined representation $S_{refined}$ for S, as shown in Figure 2. In our experiments, each memory contains 20 slots, which are randomly initialized and optimized during training.

4 Evaluation and Results

Dataset: We use the RST Discourse Treebank (Carlson et al., 2001) with the same split as in (Li et al., 2014), i.e., 312 for training, 30 for development and 38 for testing. We experiment with two set of relations, the 111 types of fine-grained relations and the 19 types of coarse-grained relations, respectively.

Evaluation Metrics: In the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), *head* is the core of a discourse, and a *dependent* gives supporting evidence to its head with certain relationship. We adopt unlabeled accuracy UAS (the ratio of EDUs that correctly identify their heads) and labeled accuracy LAS (the ratio of EDUs that have both correct heads and relations) as our evaluation metrics.

Baselines: We compare our method with the following baselines and models: (1) **Perceptron**: We re-implement the perceptron based arc-eager style dependency discourse parser as mentioned in (Jia et al., 2018) with coarse-grained relation. The **Perceptron** model chooses words, POS tags, positions and length features, totally 100 feature templates, with the *early update* strategy (Collins and Roark, 2004). (2) **Jia18**: Jia et al. (2018) implement a transition-based discourse parser with stacked LSTM, where they choose a two-layer LSTM to represent EDUs by encoding four kinds of features including words, POS tags, positions and length features. (3) Basic EDU representation (**Basic**): Our discourse parser with the basic EDU representation method mentioned in Section 3. (4) Memory refined representation (**Refined**): Our full parser equipped with the basic EDU representation method and the memory networks to capture the discourse cohesion mentioned in Section 3. (5) **MST-full** (Li et al., 2014): a graph-based dependency discourse parser with carefully selected 6 sets of features including words, POS tags, positions, length, syntactic and semantic similarity features, which achieves the state-of-art performance on the RST Treebank.

4.1 Results

We list the overall discourse parsing performance in Table 1. Here, **Jia18**, a stack LSTM based method (Jia et al., 2018), outperforms the traditional **Perceptron** method, but falls behind our **Basic** model with *word*, *POS* tags and *Position* features. The reason may be that representing EDUs directly from the sequence of word/POS embeddings could probably capture the semantic meaning of EDUs, which is especially useful for taking into account synonyms or paraphrases that often confuse traditional feature-based methods. We can also see that **Basic**(word+pos+position) significantly outperforms **Basic**(word+pos), as the *Position* features may play a crucial role in providing useful structural clues to our parser. Such position information can also be considered as a shallow treatment to capture the discourse cohesion, especially for long span scenarios. When using the memory network, our **Refined** method achieves better performance than the **Basic**(word+pos+position) in both UAS and LAS. The reason may come from the ability of the memory networks in simulating the lexical chains within a discourse, where the memory networks can model the discourse cohesion so as to provide topical or structural clues to our parser. We use SIGF V2 (Padó, 2006) to perform significance test for the discussed models. We find that the **Basic**(word+pos+position) method significantly outperforms (Jia et al., 2018), and our **Refined** model performs significantly better than **Basic**(word+pos+position) (with $p < 0.1$).

However, when compared with **MST-full** (Li et al., 2014), our models still fall behind this state-of-the-art method. The main reason might be that **MST-full** follows a global graph-based dependency parsing framework, where their high order methods (in cubic time complexity) can directly analyze the relationship between any EDUs pairs in the discourse, while, we choose the transition-based local method with linear time complexity, which can only investigate the top EDUs in S and B according to the selected actions, thus usually has a lower performance than the global graph-based methods, but with a lower (linear) time complexity. On the other hand, the neural network components help us maintain much fewer features than **MST-full**, which carefully selects 6 different sets of features that are usually obtained using extra tools and resources. And, the neural network design is flexible enough to incorporate various clues into a uniform framework, just like how we introduce the memory networks as a proxy to capture discourse cohesion.

In the RST corpus, when the distance between two EDUs is larger, there are usually fewer numbers of such EDU pairs, but the parsing performance for those long span cases drops more significantly. For example, the LAS is even lower than 5% for those dependencies that have a range of 6 EDUs. We take a detailed look at the parsing performance for dependencies at different lengths (from 1 to 6 as an example) using coarse-grained relations. As shown in Table 2, compared with the **Basic** method, both UAS and LAS of the **Refined** method are improved significantly in almost all spans, where we observe more prominent improvement for the UAS in larger spans such as **span 5** and *span 6*, with about 8.70% and 6.38%, respectively.

Method	UAS	LAS (Fine)	LAS (Coarse)
Perceptron	0.5422	0.3231	0.3777
Jia18	0.5852	0.3286	0.4037
Basic (word+pos)	0.5588	0.367	0.3985
Basic (word+pos+position)	0.5933	0.3832	0.4305
Refined (20 slots)	0.6197	0.3947	0.4445
MST-full	0.7331	0.4309	0.4851

Table 1: Overall discourse parsing performance in the RST dataset.

span	Basic(word+pos+position)		Refined (20)	
(count)	UAS	LAS	UAS	LAS
1(1225)	0.7796	0.618	**0.8261**	**0.6261**
2 (405)	**0.6198**	0.4	0.6025	**0.4124**
3 (212)	0.434	0.2217	**0.4576**	**0.2642**
4 (125)	0.256	0.112	**0.296**	**0.128**
5 (69)	0.1739	0.0725	**0.2609**	**0.1015**
6 (47)	0.1064	0.0426	**0.1702**	**0.0638**

Table 2: Performance in different discourse spans.

Finally, let us take a detailed comparison between **Refined** and **Basic** to investigate the advantages of capturing discourse cohesion. Note that, our **Refined** method wins **Basic** in almost all relations. Here, we discuss one typical relation *List*, which often indicates a long span dependency between a pair of EDUs. In the test set of RST, the average span for *List* is 7.55, with the max span of 69. Our **Refined** can successfully identify 55 of them, with an average span of 9.02 and the largest one of 63, while, the **Basic** method can only identify 41 edges labeled with *List*, which are mostly shorter cases, with an average span of 1.32 and the largest one of 5. More detailedly, there are 18 edges that are correctly identified by our **Refined** but missed by the **Basic** method. The average span of those dependencies is 25.39. It is easy to find that without further considerations in discourse structures, the **Basic** method has limited ability in correctly identifying longer span dependencies. And those comparisons prove again that our **Refined** can take better advantage of modeling discourse cohesion, which enables our model to perform better in long span scenarios.

5 Conclusions

In this paper, we propose to utilize memory networks to model discourse cohesion automatically. By doing so we could capture the topic change or threads within a discourse, which can further improve the discourse parsing performance, especially for long span scenarios. Experimental results on the RST Discourse Treebank show that our proposed method can characterize the discourse cohesion efficiently and archive significant improvement over traditional feature based discourse parsing methods.

Acknowledgments

We would like to thank our anonymous reviewers, Bingfeng Luo, and Sujian Li for their helpful comments and suggestions, which greatly improved our work. This work is supported by National High Technology R&D Program of China (Grant No.2015AA015403), and Natural Science Foundation of China (Grant No. 61672057, 61672058). For any correspondence, please contact Yansong Feng.

References

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon,* *Portugal, September 17-21, 2015*. pages 349–359. http://aclweb.org/anthology/D/D15/D15-1041.pdf.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the SIGDIAL 2001 Workshop, The 2nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, Saturday, September 1, 2001 to Sunday, September 2, 2001, Aalborg, Denmark*. http://aclweb.org/anthology/W/W01/W01-1605.pdf.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain.*. pages 111–118. http://aclweb.org/anthology/P/P04/P04-1015.pdf.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. pages 334–343. http://aclweb.org/anthology/P/P15/P15-1033.pdf.

David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. 2010. Building watson: An overview of the deepqa project. *AI Magazine* 31(3):59–79. http://www.aaai.org/ojs/index.php/aimagazine/article/view/2303.

M.A.K. Halliday and Ruqaiya Hasan. 1989. *Language, Context, and Text: Aspects of Language in a Social-Semiotic Perspective*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Yanyan Jia, Yansong Feng, Yuan Ye, Chao Lv, Chongde Shi, and Dongyan Zhao. 2018. Improved discourse parsing with two-step neural transition-based model. *ACM Trans. Asian & Low-Resource Lang. Inf. Process.* 17(2):11:1–11:21. https://doi.org/10.1145/3152537.

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 *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*. pages 486–496. http://aclweb.org/anthology/P/P13/P13-1048.pdf.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*. pages 25–35. http://aclweb.org/anthology/P/P14/P14-1003.pdf.

Annie Louis, Aravind K. Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summarization. In *Proceedings of the SIGDIAL 2010 Conference, The 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 24-15 September 2010, Tokyo, Japan*. pages 147–156. http://www.aclweb.org/anthology/W10-4327.

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 Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*. pages 55–60. http://aclweb.org/anthology/P/P14/P14-5010.pdf.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* 17(1):21–48.

J Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Iwpt-2003 : International Workshop on Parsing Technology*. pages 149–160.

Sebastian Padó. 2006. *User's guide to `sigf`: Significance testing by approximate randomisation*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1532–1543. http://aclweb.org/anthology/D/D14/D14-1162.pdf.

Livia Polanyi and Martin van den Berg. 2011. Discourse structure and sentiment. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, Vancouver, BC, Canada, December 11, 2011*. pages 97–102. https://doi.org/10.1109/ICDMW.2011.67.