A Multi-stage Strategy for Chinese Discourse Tree Construction

Tishuang Wang, Peifeng Li, Qiaoming Zhu
Natural Language Processing Lab, School of Computer Science and Technology
Soochow University
Suzhou, China
tswang@stu.suda.edu.cn, {pfli, qmzhu}@suda.edu.cn

***Abstract*—Building discourse tree is crucial to improve the performance of discourse parsing. There are two issues in previous work on discourse tree construction, i.e., the error accumulation and the influence of connectives in transition-based algorithms. To address above issues, this paper proposes a tensor-based neural network with the multi-stage strategy and connective deletion mechanism. Experimental results on both CDTB and RST-DT show that our model achieves the state-of-the-art performance.**

Keywords-discourse parsing; tree construction; chinese discourse treebank

I. INTRODUCTION

According to the Rhetorical Structure Theory (RST) [1], discourse usually composed of a series of Element Discourse Units (EDUs, e.g., words, phrases, sentences or paragraphs), which is an organized, hierarchical whole. Discourse parsing aims to identify the structures and relationships with semantic connection and combines adjacent EDUs with rhetorical relations in a hierarchical way to represent an entire document as a discourse tree. As a subtask of discourse parsing, discourse tree construction can assist to analyze and understand the information of discourse, and it is widely used in many down-steam NLP tasks, such as information extraction [2], summarization [3], and question answering [4].

Discourse (structure) tree construction is to recursively connect EDUs by rhetorical relation to larger text spans until the final tree is built. To make a clearer explanation of the discourse tree, take the chtb_0013 as an example, which is a typical news article from Chinese Treebank 8.0 [5].

Example1: 大运河作为一条水运大动脉，为沿岸企业提供了运输和给排水之便 $_a$，成为企业发展的生命通道 $_b$。据江苏苏钢集团公司负责人介绍，苏钢每年要靠大运河运输原料、成品一百五十万吨 $_c$。其中煤炭从徐州运来，走运河比走陆路运费每吨便宜十五元 $_d$，仅此一项大运河每年就为厂家节约成本二千多万元 $_e$。运河整治后，苏钢在运河对岸建了新厂区 $_f$，并自筹资金建设了一座跨运河大桥 $_g$，把新厂、老厂连为一体 $_h$。As the main artery of water transport, the Grand Canal provides transportation and water supply and drainage for coastal enterprises $_a$, become the life channel of enterprise development $_b$. According to the person in charge of Jiangsu Sugang Group Co., Ltd., Sugang relies on the Grand Canal to transport raw materials and finished products up to 1.5 million tons each year $_c$. Among them, coal is transported from Xuzhou, and it's 15 yuan cheaper per ton than land freight by canal $_d$. Only this one, Grand Canal saves more than 20 million yuan for manufacturers each year $_e$. After the canal was rehabilitated, Sugang built a new factory on the opposite side of the canal $_f$, and they raised funds to build a bridge across the canal $_g$, Connect the new factory and the old factory $_h$.

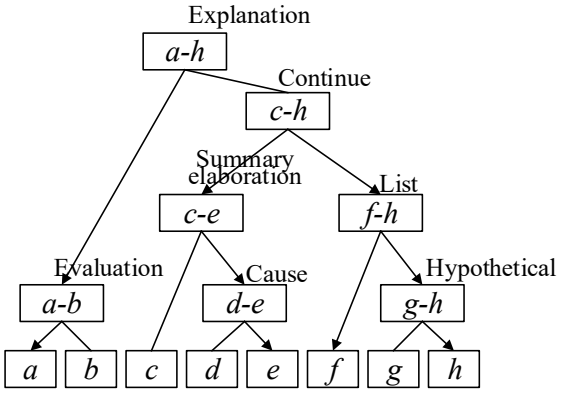

Figure1. Discourse tree of chtb_0013

This whole discourse tree with 15 spans is shown as Figure 1 and there are eight EDUs (*a-h*). The leaf of this tree is EDU and the internal node is relational node. The arrow points to the important span. In this paper, we only focus on how to construct the naked Chinese discourse tree, ignoring the relations (e.g., *Evaluation*) and the nuclearity (e.g., *Nucleus* and *Satellite*).

Although there are many studies on discourse parsing due to its vital role in NLP, only a few address discourse tree construction [6, 7, 8, 9, 10, 11]. Among them, only two studies [10, 11] explored discourse tree construction in Chinese due to the lack of annotated corpus and the abstract nature of Chinese itself. In addition, those studies heavily relied on manual feature engineering.

Those transition-based algorithms are widely used in discourse tree construction and there are two issues in Chinese discourse tree construction using shift-reduce algorithm. The first is the error accumulation. The error of the previous prediction in the shift-reduce algorithm will directly lead to the deviation of the subsequent prediction results. We found that the performance of discourse tree construction drops quickly with the increase of the EDU number. The second is the influence of connectives. The connectives have a greater impact on action prediction in the shift-reduce algorithm. It tends to connect two EDUs to a new span when there is a connective between them. For example, the connective (并 and) between two spans f and g-h in example1 will make the algorithm connect them to a span.

In this paper, to solve the above two issues in Chinese discourse tree construction, we propose a tensor-based neural network with the multi-stage strategy and connective deletion mechanism. Experimental results on CDTB, a Chinese discourse corpus, show that our model achieves the state-of-the-art performance.

II. RELATED WORK

Most of previous work focused on constructing English discourse tree. The algorithms of discourse tree construction on Rhetorical Structure Theory Discourse Treebank (RST-DT) [12], a English corpus, can mainly be categorized as shift-reduce algorithms [6, 7, 13], probabilistic CKY-like algorithms [8, 14, 15] and greedy bottom-up algorithms [9]. Wang et al. [7] used a transition-based system to build discourse trees with nuclearity labels and then used Support Vector Machines (SVM) to determine the discourse relation at different text levels. Joty et al. [8] used the sequence labeling instead of classification, and used Dynamic Conditional Random Field (DCRF) model combining structure recognition with relationship recognition. Li et al. [15] proposed an attention-based hierarchical Bi-LSTM network to learn the representations of the text spans and used a tensor-based transformation function to capture interactions among the features of the text spans. Feng et al. [9] adopts a greedy bottom-up approach, with two linear-chain CRFs applied in cascades as local classifiers. To enhance the accuracy of the pipe line, they add additional constraints in the Viterbi decoding of the first CRF, and used the novel approach of post-editing, which modifies a fully-built tree by considering information from constituents on the upper levels.

As for Chinese discourse tree construction, there are only two studies [10,11] on the Chinese Discourse Treebank (CDTB) [16]. Kong and Zhou [10] proposed a CDT-styled End-to-End discourse parser, which can automatically detect discourse units in a free text, generates the discourse parse tree in a bottom-up way, and determines the sense and centering attributions for all nonterminal nodes by traversing the discourse parse tree. Sun and Kong [11] used shift-reduce algorithm and then used Convolutional Neural Networks (CNN) with different convolution windows, proposed a complete Chinese discourse structure generating framework which can be used to generate the tree-like structure from plain texts. Furthermore, those studies heavily relied on manual feature engineering.

III. CONSTRUCTING CHINESE DISCOURSE TREE WITH MULTI-STAGE STRATEGY AND CONNECTIVE DELETION MECHANISM

In this section, we first introduce the basic model, a tensor-based neural network, to construct Chinese discourse tree, and then apply the multi-stage strategy and connective deletion mechanism to further help the basic model to improve the performance.

A. Basic Model

We also use transition-based (shift-reduce) algorithm to build the Chinese discourse trees and this process is modeled as a sequence of shift and reduce action with a stack and a queue. The stack is initialized to be empty and the queue contains all EDUs in a document. At each step, a tensor-based neural network is to perform either shift or reduce. The action shift pushes the first EDU in the queue on the top of the stack, while the reduce action 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. Finally, a discourse tree can be constructed until the queue is empty and the stack contains only the root node of the discourse tree.

For the two elements DU_{s2}, DU_{s1} at the top of the stack and the first element DU_{q1} in the queue, a tensor-based neural network, as showed in Figure 2, is introduced to judge whether the relationship between DU_{s1} and DU_{s2} is closer or more closely related to DU_{q1}. If it is closer to DU_{s2}, the reduce action is performed; otherwise, the shift action is performed. Our tensor-based neural network consists of three parts: 1) Input and Coding; 2) Tensor-based Matching; and 3) Classification.

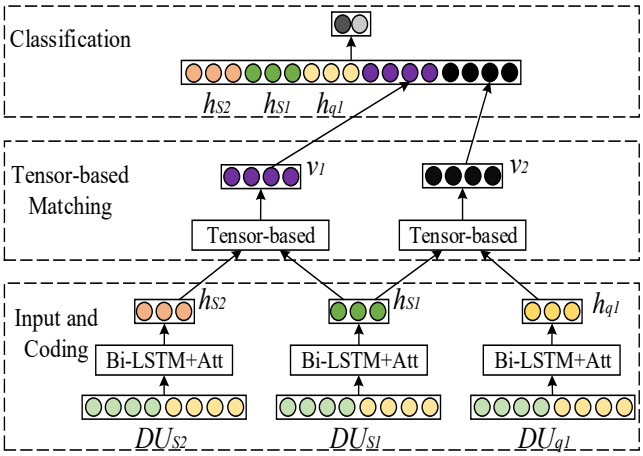

Figure 2. The tensor-based neural network model

Input and Coding

In the module Input and Coding, the input are the word sequences and POSs (Part-Of-Speeches) of the first two elements DU_{s1} and DU_{s2} in the top of the stack and the first element DU_{q1} in the queue, and then this module encodes the units DU_{s2}, DU_{s1}, and DU_{q1} using Bi-LSTM and attention mechanism to obtain their semantic vectors h_{s1}, h_{s2} and h_{q1}.

We combine the last outputs of the Bi-LSTM to be $h_s = [\overrightarrow{h_{e_n}}, \overleftarrow{h_{e_1}}]$. We also combine the outputs of the Bi-LSTM at every step: $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$ and thus get a matrix $H = [h_1; h_2; \ldots; h_n]$. Taking H and h_s as inputs, we get a vector α standing for weights and use it to get representation of the DU r:

$$M = \tanh(\boldsymbol{W}_y H + \boldsymbol{W}_l h_s \otimes e_n) \quad (1)$$

$$\alpha = \mathrm{softmax}(\boldsymbol{W}_\alpha^T M) \quad (2)$$

$$r = H\alpha \quad (3)$$

where $\otimes$ denotes Cartesian product, e_n is a n dimensional vector of all 1s, and $\boldsymbol{W}_y, \boldsymbol{W}_l, \boldsymbol{W}_\alpha$ are parameters.

Finally, we synthesize the information of r and h_s to get the final representation of the DU:

$$w_h = \sigma(\boldsymbol{W}_{hr} r + \boldsymbol{W}_{hh} h_s) \quad (4)$$

$$h = w_h \odot h_s + (1 - w_h) \odot r \quad (5)$$

where ⊙ denotes Hadamard product, $\boldsymbol{W_{hr}}, \boldsymbol{W_{hh}}$ are parameters, w_h is the representation of weight vector calculated by r and h_s. h is the final representation of the *DU* by the Bi-LSTM and Attention.

Tensor-based Matching

Based on the discourse unit DU_{s1}, DU_{s2} and DU_{q1}, we obtain the semantic representation vectors h_{s1}, h_{s2} and h_{q1} from the Input and Coding module. Then these semantic representations are fed into the Tensor-based Matching module, which uses tensor-based transformation function [15] to incorporate the interaction between DU_{s1} and DU_{s2} and the interaction between DU_{s1} and DU_{q1}.

$$v_1 = Relu(\boldsymbol{W_h}[h_{s_1}, h_{s_2}] + [h_{s_1}, h_{s_2}]^T \boldsymbol{P_h Q_h}[h_{s_1}, h_{s_2}] + \boldsymbol{b_h}) \quad (6)$$

$$v_2 = Relu(\boldsymbol{W_h}[h_{s_1}, h_{q_1}] + [h_{s_1}, h_{q_1}]^T \boldsymbol{P_h Q_h}[h_{s_1}, h_{q_1}] + \boldsymbol{b_h}) \quad (7)$$

where $\boldsymbol{W_h}, \boldsymbol{P_h}, \boldsymbol{Q_h}, \boldsymbol{b_h}$ are parameters to incorporate the interaction between DU_{s1} and DU_{s2} and the interaction between DU_{s1} and DU_{q1}, and we choose $Relu$ as the activation function.

Classification

Finally, three semantic vectors ($h_{s1}, h_{s2}$and h_{q1}) and two interactive vectors (v_1 and v_2) are combined and then sent to the output layer, i.e., the Classification module, through a nonlinear transformation.

$$c = [h_{s_1}, h_{s_2}, h_{q_1}, v_1, v_2] \quad (8)$$

$$y = \text{softmax}(\boldsymbol{W_y} c + \boldsymbol{b_y}) \quad (9)$$

where $\boldsymbol{W_y}$ and $\boldsymbol{b_y}$ are the parameters. During the training, we use the Adam optimizer to optimize the network parameters by maximizing the log-likelihood loss function.

B. Multi-stage Strategy

The shift-reduce algorithm will cause error accumulation, and the error of the previous prediction will directly lead to the deviation of the subsequent prediction results. We found that the performance of discourse tree construction drops quickly with the increase of the EDU number. To reduce the cascading errors in shift-reduce algorithm, we propose a multi-stage strategy to construct discourse tree, as shown in Figure 3. It first constructs the sentence-level subtree, and then constructs the paragraph-level subtree based on its sentence-level subtrees. Finally, it constructs the document-level tree on its paragraph-level subtrees if this document has more than one paragraph.

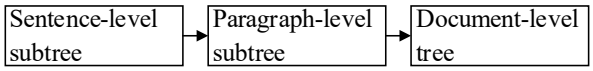

Figure3. Multi-stage in discourse tree construction.

The article in Figure 1 has four sentences, as showed in Figure 4(a) (each dashed box contains a sentence). It is easy to understand that each sentence can form a subtree in most cases. Therefore, we first construct four sentence-level subtrees for each sentence, as showed in Figure 4(a). Then we construct the paragraph-level subtree based on four sentence-level subtrees, as showed in Figure 4(b). If this article has more than one paragraph, we will continue to construct its document-level tree.

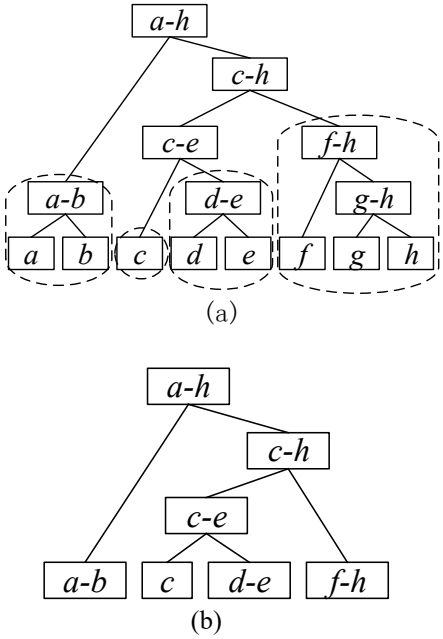

Figure 4. Two-stage to construct a discourse tree.

Feng and Hirst [9] also proposed a two-stage (sentence-level and document-level) bottom-up, greedy parser with linear-chain CRF models. However, their model used the sliding window to get contextual features and heavily relied on manual features engineering. Different from them, our model uses the shift-reduce algorithm to build discourse tree and uses the tensor-based neural network to classify actions. Besides, we do not use any additional manual features.

C. Connective Deletion

Connectives have a greater impact on action prediction in the shift-reduce algorithm. For example in chtb_0013, there is a connective (并 and) between two spans *f* and *g-h*. The current state is that the two elements in the stack are *f* and *g* and the first element in the queue is *h*. Our model is more inclined to connect *f* and *g* due to the above connective between them.

There are two types of connective in our model, intra-sentence connectives and inter-sentence connectives. Our multi-stage strategy can solve the impact of inter-sentence connective because each sentence must be a subtree, as the connective (其中 Among them) in chtb_0013, our multi-stage strategy to construct discourse tree will builds *c* and *d-e* into a subtree respectively, at this time, the connective (其中 Among them) is connected to the subtrees of *c* and *d-e*, and then our multi-stage strategy to construct discourse tree avoid connecting *c* and *d* directly. However, the intra-sentence connectives still have an adverse effect on constructing sentence-level subtree, as the connective (并 and) in chtb_0013 connecting is still *f* and *g*. Therefore, we use a simple mechanism to solve this issue, i.e., deleting the intra-sentence connectives and retaining the inter-sentence connectives.

IV. EXPERIMENTS

In this section, we first introduce the dataset and experimental setting, then report the experimental results on CDTB. Finally, we also evaluate our model on the English corpus RST-DT.

A. Experimental Setup

We conducted our experiments on the Chinese Discourse TreeBank (CDTB) [16]. This corpus is built on the Chinese Treebank (CTB) [5] with a connective-driven dependency tree scheme. Each paragraph in CDTB is marked as a tree and CTDB consists of 500 newswire articles, which are further divided into 2342 paragraphs with a tree representation for one paragraph. Hence, we only apply a two-stage strategy for CDTB because this corpus does not provide document-level tree. Besides, CDTB contains 10650 EDUs, and each EDU has 22 Chinese characters on average.

Following Sun and Kong [10], we choose 425 for training, 25 for development and 50 for testing. We follow Morey et al. [17] to report the micro-averaged and macro-averaged F1-scores. In addition, we also report the accuracy of the whole tree structure (Tree Acc). For comparison with previous studies, the experimental result returned on CDTB corpus does not contain leaf nodes of discourse tree, but on RST-DT the leaf nodes are included.

The dimension of the word embeddings is set to 300, and the dimension of the POS embeddings is set to 50. We pre-trained the word embeddings with Word2Vec on the Wikipedia Chinese corpus. For a fair comparison, all of the models in our experiments use the same parameters. The number of LSTM neurons is set to 150, and the number of Attention neurons is set to 50. We adopt the dropout strategy to avoid overfitting and set the dropout rate to 0.5.

B. Experimental Results

We compare our method with the following state-of-the-art Chinese baselines:

- **KZ17** [10]: used contextual features, lexical features and dependency tree features to build discourse tree by a maximum entropy (ME) classifier with a greedy bottom-up algorithm.
- **SK18** [11]: used shift-reduce algorithms and then used a stack-augmented parser-interpreter CNN model with the features of different size windows.
- **LLC16** [15]: A Bi-LSTM model with the attention mechanisms and the tensor-based transformation function and it used probabilistic CKY-like algorithms.

TABLE I. OVERALL PERFORMANCE IN CDTB

Model	Macro-F1	Micro-F1
LLC16	74.8	60.8
KZ17	67.3	57.1
SK18	84.0	-
Ours	**86.8**	**79.9**

Table 1 shows the performance of our model and three baselines on CDTB and it illustrates that our model outperforms all the others significantly both on the micro-averaged and macro-averaged F1-score. This result verifies that our tensor-based neural network and two-stage strategy are beneficial to Chinese discourse tree construction. Our Tensor-based Matching module is similar to LLC16, and the improvement of our model shows that our tensor-based neural network and two-stage strategy are more suitable for Chinese discourse tree construction.

To further explore the different influence of our two-stage and the mechanism of connective deletion in our model, we implement three simplified versions (i.e., *Simp1/2/3*) as show in table 2 where *Stage* means whether the two-stage strategy is adopted and *Del* represents whether the intra-sentence connective deletion mechanism is used.

TABLE II. COMPARISON WITH SIMPLIFIED VERSIONS

Model	Stage	Del	Macro-F1	Micro-F1	Tree Acc
Simp1	No	No	74.5	59.6	44.5
Simp2	No	Yes	77.3	64.0	48.5
Simp3	Yes	No	85.5	77.9	58.1
Ours	Yes	Yes	**86.8**	**79.9**	**60.3**

The simplest model Simp1 is like to Li et al. [15] except that we use the shift-reduce algorithms. Comparing with Simp1, we can find that Simp2 with the connective deletion mechanism achieves the improvements on all three metrics. This result ensures that this mechanism is helpful for discourse tree construction. We can also observe that Simp3 with the two-stage strategy improves the micro-averaged/macro-averaged F1-score and the whole tree accuracy by 11, 18.3 and 13.6, respectively. This result verifies that our two-stage strategy can reduce the complexity of discourse tree construction.

Table 3 shows the performance comparison of one-stage and two-stage strategy both on sentence-level subtree and paragraph-level subtree performance. We can find that our two-stage strategy can improve the Micro-F1 scores of sentence-level and paragraph-level subtree construction simultaneously.

TABLE III. COMPARISON OF THE EFFECT OF TWO-STAGE STRATEGY ON DIFFERENT LEVEL SUBTREE (MICRO-F1)

	one-stage	two-stage
sentence-level	60.7	80.8
paragraph-level	72.2	78.3

Table 4 shows the effectiveness of connective deletion mechanism on those explicit and implicit nodes, in which the explicit nodes mean that the adjacent DUs contain connectives. We can find that the connective deletion mechanism improves the Micro-F1 scores of the explicit and implicit nodes by 2.4% and 1.8%, respectively.

TABLE IV. THE EFFECTIVENESS OF CONNECTIVE DELETION MECHANISM ON EXPLICIT AND EXPLICIT NODES (MICRO-F1)

	w/o deletion	deletion
explicit	75.7	78.1
implicit	78.6	80.4

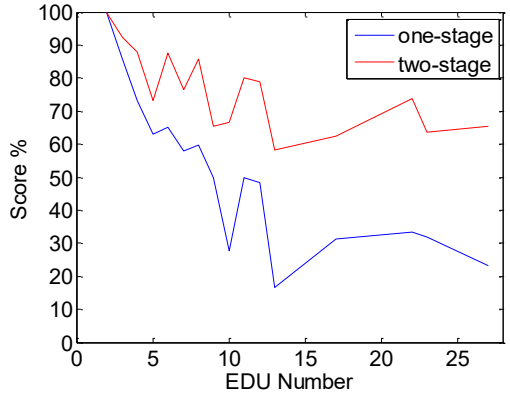

Figure 5. Micro-averaged F1 in different EDUs number.

Figure 5 shows the micro-averaged F1-scores with different EDUs numbers in a discourse tree. We can find that the micro-averaged F1-score of one-stage strategy drop significantly with the increase of the number of EDUs. However, this figure of the two-stage strategy is relatively stable. This result proves that our two-stage strategy is better than the one-stage strategy in all EDUs numbers.

We find that the construction performance of sentence-level subtree is similar to that of paragraph-level tree. The main reason is that the sentence-level subtree and paragraph-level tree are quite complex and they have a similar average number of EDUs (3.0 and 2.9). Besides, we can also find out that the sentence-level subtree and paragraph-level tree are relatively simple in CDTB for their small EDUs numbers.

C. *Experimental Result on RST-DT*

We also evaluate our model on an English corpus, RST-DT [12], and all parameter settings are the same as on CDTB. In CDTB, each paragraph forms one discourse tree and each paragraph contains 4.5 EDUs on average. In RST-DT, one document forms one discourse tree and each document contains 55.6 EDUs on average. Hence, these two corpora are different in many aspects. Since RST-DT provides a document-level discourse tree, a three-stage strategy is used in our model.

We use the same data split as in Li et al. [14], i.e., 312 for training, 30 for development and 38 for testing. Three state-of-the-art baselines are selected for comparison: 1) BCS17 [13]: a transition-based discourse parser using a feed-forward neural network and a shift-reduce algorithm; 2) LLC16 [15]: a Bi-LSTM model with the attention mechanisms and the tensor-based transformation function and it used probabilistic CKY-like algorithms; 3) FH14 [9]: a two stage (sentence-level and document-level) bottom-up, greedy parser with linear-chain CRF models. Table 5 shows the performance of three baselines and our model.

TABLE V. PERFORMANCE COMPARISON WITH STATE-OF-THE-ART DISCOURSE PARSERS.

Model	Macro-F1	Micro-F1
BSC17	85.1	81.3
LLC16	85.4	82.2
FH14	87.0	**84.3**
Ours	**87.2**	83.4

The preliminary experimental results show that our model achieves comparable performance to those state-of-the-art discourse parsers. Especially, our model outperforms the other two neural network models, a transition-based (shift-reduce algorithm) model (BSC17), and a tensor-based model (LLC16). However, compared with FH14, our model achieves the similar performance. It is worthy to note that FH14 uses many additional manual features, while we only use the sentence as the input.

Compared with CDTB, the structure of RST is more complex. The main reason is that RST annotates the structure between paragraphs. In addition, some sentence does not have a well-formed subtree, because some of its units attach to the left and some to the right. Vliet and Redeker [18] called these cases as 'leaky' boundaries. This situation accounts for 5% of RST-DT and only 0.1% of CDTB. Therefore, our multi-stage strategy for discourse tree construction works better on CDTB.

In CDTB, each paragraph forms one discourse tree and the EDU number of all discourse tree is less than 30. In RST-DT, one document forms one discourse tree, the largest discourse tree contains 304 EDUs. Figure 6 shows the micro-averaged F1 of discourse tree with less than 30 EDUs in CDTB and RST-DT. We can find that RST-DT performs better than CDTB for discourse tree with less than 30 EDUs, this is because the structure of English is clear and precise, and however, the expression in Chinese is freer. But due to the low performance of the discourse tree with more than 30 EDUs in English, the overall performance of CDTB is better than RST-DT.

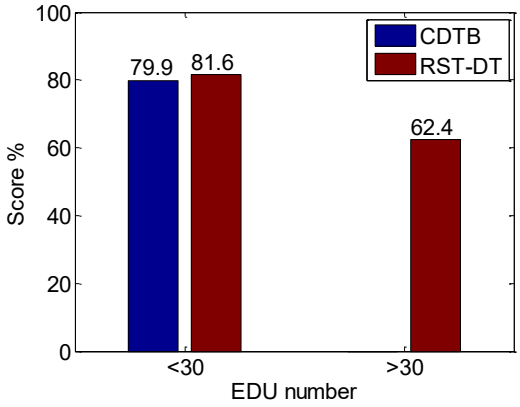

Figure 6. Micro-averaged F1 in different EDUs number on CDTB and RST-ST.

V. CONCLUSIONS

In this paper, we focus on two issues in naked Chinese discourse tree construction, the error accumulation and the influence of connectives in transition-based algorithms, and then propose a tensor-based neural networks with the multi-stage strategy and the connective deletion mechanism. Experimental results on both the Chinese and English corpora (CDTB and RST-DT) show that our model outperforms the state-of-the-art systems. In our future work, we will focus on how to provide more effective and language-independent methods to replace our simple mechanisms in this paper.

REFERENCES

[1] W. Mann, S. Thompson, "Rhetorical structure theory: Toward a functional theory of text organization." Text-Interdisciplinary Journal for the Study of Discourse, 1988, PP. 243-281.

[2] B. W. Zou, G. D. Zhou, Q. M. Zhu, "Negation focus identification with contextual discourse information." Proc. ACL 2014, pp. 522-532.

[3] A. Cohan, N. Goharian, "Scientific document summarization via citation contextualization and scientific discourse." International Journal on Digital Libraries, 2018, pp.187-303.

[4] M. Liakata, S. Dobnik, S. Saha, "A discourse-driven content model for summarising scientific articles evaluated in a complex question answering task." Proc. EMNLP 2013, pp. 747-757.

[5] N. Xue, F. Xia, F. Chiou, M. Palmer, "The Penn Chinese Treebank: Phrase structure annotation of a large corpus." Natural Language Engineering, 2005, pp. 207-238.

[6] Y. Ji, J. Eisenstein, "Representation learning for text-level discourse parsing." Proc. ACL 2014, pp. 13-24.

[7] Y. Wang, S. Li, H, Wang, "A two-stage parsing method for text-level discourse analysis." Proc. ACL 2017, pp. 184–188.

[8] S. Joty, G. Carenini, R. Ng, and Y. Mehdad, "Combining intra- and multi-sentential rhetorical parsing for document-level discourse analysis." Proc. ACL 2013, pp. 486–496.

[9] V. W. Feng, G. Hirst, "Text-level discourse parsing with rich linguistic features." Proc. ACL 2012, pp. 60–68.

[10] F. Kong, G. D. Zhou, "A CDT-styled end-to-end Chinese discourse parser." ACM Transactions on Asian and Low-Resource Language Information Processing, 2017, pp. 16-26.

[11] C. Sun, F. Kong, "A Transition-based Framework for Chinese Discourse Structure Parsing." Journal of Chinese Information Processing, 2018, pp. 48-56.

[12] L. Carlson, D. Marcu, M. Okurowski, "Building a discourse-tagged corpus in the framework of rhetorical structure theory." Current and new directions in discourse and dialogue, 2003, pp. 85-112.

[13] C. Braud, M. Coavoux, A. Søgaard, "Cross-lingual rst discourse parsing." Proc. EACL 2017, pp. 292–304.

[14] J. Li, R. Li, E. Hovy, "Recursive deep models for discourse parsing." Proc. EMNLP 2014, pp. 2061–2069.

[15] Q. Li, T. Li, and B. Chang, "Discourse parsing with attention-based hierarchical neural networks." Proc. EMNLP 2016, pp. 362–371.

[16] Y. C. Li, F. Kong, and G. D. Zhou, "Building Chinese discourse corpus with connective-driven dependency tree structure." Proc. EMNLP 2014, pp. 2105-2114.

[17] M. Morey, P. Muller, N. Asher, "How much progress have we made on RST discourse parsing? A replication study of recent results on the RST-DT." Proc. EMNLP 2017, pp. 1330–1335.

[18] V. Nynke, D. Vliet, and R. Gisela. "Complex Sentences as Leaky Units in Discourse Parsing." In Proceedings of Constraints in Discourse, Agay-Saint Raphael, 2011.