Context Tracking Network: Graph-based Context Modeling for Implicit Discourse Relation Recognition

Yingxue Zhang[1], Fandong Meng[1], Peng Li[1], Ping Jian[2], Jie Zhou[1]
[1]Pattern Recognition Center, WeChat AI, Tencent Inc, China
[2]Beijing Institute of Technology, China
{yxuezhang, fandongmeng, patrickpli, withtomzhou}@tencent.com
pjian@bit.edu.cn

Abstract

Implicit discourse relation recognition (IDRR) aims to identify logical relations between two adjacent sentences in the discourse. Existing models fail to fully utilize the contextual information which plays an important role in interpreting each local sentence. In this paper, we thus propose a novel graph-based Context Tracking Network (CT-Net) to model the discourse context for IDRR. The CT-Net firstly converts the discourse into the paragraph association graph (PAG), where each sentence tracks their closely related context from the intricate discourse through different types of edges. Then, the CT-Net extracts contextual representation from the PAG through a specially designed cross-grained updating mechanism, which can effectively integrate both sentence-level and token-level contextual semantics. Experiments on PDTB 2.0 show that the CT-Net gains better performance than models that roughly model the context.

1 Introduction

Implicit discourse relation recognition (IDRR) aims to identify logical relations between two adjacent sentences in discourse without the guidance of connectives (e.g., because, but), which is one of the major challenges in discourse parsing. With the rise of deep learning, lots of sentence-modeling based methods (Liu and Li, 2016; Rönnqvist et al., 2017; Bai and Zhao, 2018; Xu et al., 2019; Shi and Demberg, 2019) have emerged in the field of IDRR. These methods typically focus on modeling the local semantics of these two sentences, without considering wider discourse context.

Contextual information plays an important role in understanding sentences. Take the paragraph $P = \{S_1, S_2, S_3, S_4\}$ in Figure 1 as an example, the ground-truth relation between S_3 and S_4 is "Comparison". Combining the contextual information carried by S_1 and S_2, we can more easily identify the "Comparison" relation reflected by

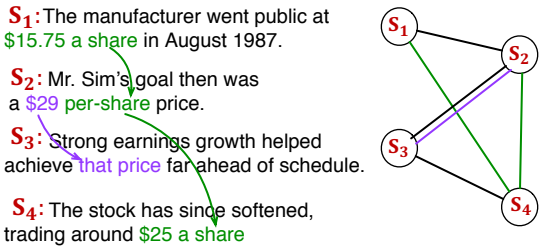

Figure 1: The paragraph association graph (PAG) (right) built for a case (left) of PDTB 2.0.

"achieve that price" (rising: "$15.75 a share" to "$29 per-share") and "softened" (falling: "$29 per-share" to "$25 a share"). Dai and Huang (2018) move one step on utilizing wider discourse context, where they use a hierarchical BiLSTM (H-LSTM) to model the whole paragraph rather than only the two sentences, to obtain context-aware sentence representation. However, there are still two limitations in their model. First, they roughly merge all the information in the paragraph, which dilutes the role of key context that closely related to the current sentence. Second, the H-LSTM suffers from the long-distance forgetting problem, which may fail to model the long-distance and non-continuous dependency across multiple sentences (like green lines in Figure 1).

To overcome these limitations, we propose a novel Context Tracking Network (CT-Net), which can track essential context for each sentence from the intricate discourse, without being affected by the spatial distance. The CT-Net computes contextual representation through two main steps. Firstly, it converts the paragraph into the paragraph association graph (PAG) (Figure 1), which contains three types of edges between sentences, namely (1) *adjacency edge* (black lines): connecting adjacent sentences, (2) *co-reference edge* (purple lines): connecting sentences with co-reference associations, and (3) *lexical chain edge* (green lines): connecting sentences containing related words. Each sentence can track closely related context along these

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1592–1599
June 6–11, 2021.

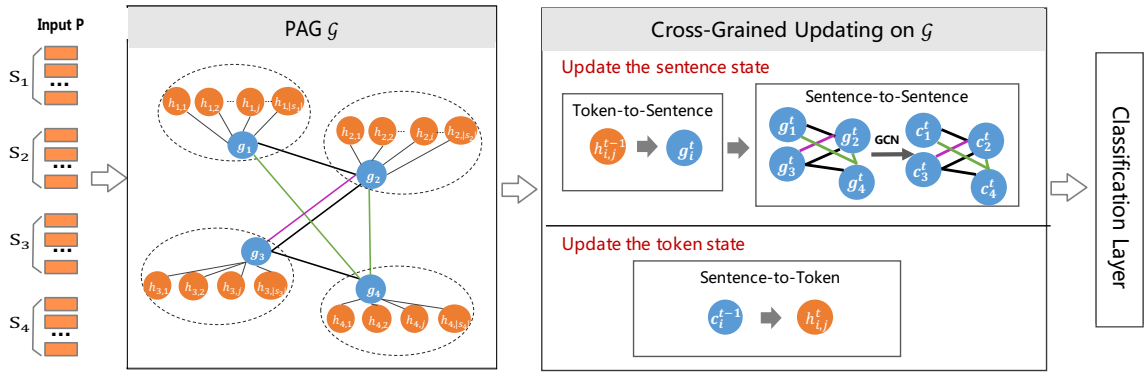

Figure 2: The overall architecture of the CT-Net. Given a paragraph $P = (S_1, S_2, S_3, S_4)$, it converts P into the PAG $\mathcal{G}$, then employs the cross-grained updating mechanism on $\mathcal{G}$ to get contextual representation for classification.

edges, including long-distance sentences involving the same object or topic. Secondly, the CT-Net extracts contextual representation over the PAG. To effectively incorporate fine-grained information carried by tokens, we propose the cross-grained updating mechanism, which will be executed multiple recurrent rounds. At each round, it performs semantic exchange via three processes:

- **Token-to-Sentence Updating**: updating the sentence representation with its tokens to grasp fine-grained semantics.
- **Sentence-to-Sentence Updating**: performing interaction between sentences on the PAG to get context-aware sentence representation.
- **Sentence-to-Token Updating**: using the context-aware sentence representation to update tokens, so that each token can also incorporate contextual information. The obtained context-aware token representation will be used for the computation of the next round.

After multiple rounds, the CT-Net obtains the contextual representation that fully combines sentence-level and token-level contextual semantics.

Our main contributions are two folds. [1] First, we propose a novel CT-Net for IDRR, which builds the PAG to track closely related context for each sentence in the intricate discourse, and incorporates multi-grained contextual semantics via the cross-grained updating mechanism. Second, experiments on PDTB 2.0 demonstrate that the CT-Net gains better performance than a variety of approaches that roughly model the discourse context.

[1]Code is available at: https://github.com/yxuezhang/CTNet

2 Model

The input of the CT-Net is a paragraph $P = (S_1, S_2, ..., S_{n-1}, S_n)$. Here, S_{n-1} and S_n are the adjacent sentences to be classified, while $S_1, ..., S_{n-2}$ are context with background information. Our goal is to identify the relation between S_{n-1} and S_n. We firstly build a paragraph association graph (PAG) for P (Section 2.1), then employ the cross-grained updating mechanism on the PAG to extract the contextual representation of S_{n-1} and S_n (Section 2.2). The contextual representation is then used for the final classification (Section 2.3).

2.1 Paragraph Association Graph

The CT-Net firstly converts the P into a PAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges respectively. As shown in Figure 2, the PAG contains sentence nodes (blue) and token nodes (orange). Each token node is connected with its corresponding sentence node. We carefully design the edges between sentence nodes so that each sentence only connects the ones that are closely related to it. Specifically, there are three types of edges between sentence nodes in the PAG:

- **Adjacency Edge** (black edges). Adjacent sentences tend to carry important contextual information. Therefore, we add adjacency edges between the neighbors in the discourse.
- **Co-reference Edge** (purple edges). Sentences with co-reference associations tend to involve the same object and be highly related, so we add a co-reference edge between them.
- **Lexical Chain Edge** (green edges). Lexical chain tracks related words that run through the whole paragraph. Sentences containing the

same words or synonyms (except stop words) tend to involve the same topic, therefore, we add a lexical chain edge between them.

We give more details of the PAG in Section 3.2.

2.2 Cross-Grained Updating Mechanism

The CT-Net then extracts contextual representation of S_{n-1} and S_n from the PAG $\mathcal{G}$ through cross-grained updating mechanism, which is executed T rounds. At the t-th round, we denote the state of the i-th sentence node as g_i^t, and the state of the j-th token node of the i-th sentence as $h_{i,j}^t$. The states transition from the (t-1)-th to the t-th round consists of three computation processes: token-to-sentence updating, sentence-to-sentence updating and sentence-to-token updating. The first two processes are responsible for updating sentence nodes, while the last one is for updating token nodes.

Node Initialization. When $t = 0$, we initialize token nodes with the concatenation of char, GloVe (Pennington et al., 2014) and ELMo (Peters et al., 2018) embeddings. And the dimension is reduced:

$$h_{i,j}^0 = x_{i,j} = W[x_{i,j}^{char}; x_{i,j}^{glove}; x_{i,j}^{elmo}] + b \quad (1)$$

where W, b are parameters. The sentence node g_i^0 is initialized as the average of its token nodes.

Token-to-Sentence Updating. This process updates the sentence state g_i^t with the token states of last round $h_{i,j}^{t-1}$. We employ Sentence-state LSTM (SLSTM) (Zhang et al., 2018) to achieve this. SLSTM is a novel graph RNN that converts a sentence into a graph with one global sentence node and several local word nodes, just like the sub-graph in the PAG (inside the dotted ellipse in Figure 2). At the t-th round, the hidden state of i-th sentence g_i^t is computed as follows:

$$g_i^t = \text{SLSTM}_{h\to g}(h_{i,0}^{t-1}, h_{i,1}^{t-1}..., h_{i,|S_i|}^{t-1}, g_i^{t-1}) \quad (2)$$

where $\text{SLSTM}_{h\to g}$ represents the process of updating the sentence state with token states by SLSTM, and its detailed equations are shown in Appendix A. $|S_i|$ is the number of tokens in S_i.

Sentence-to-Sentence Updating. After merging token semantics, sentences further grasp sentence-level contextual semantics through the interaction between sentence nodes on the PAG. Since there are three types of edges, we employ Multi-Relational GCN (Schlichtkrull et al., 2018) to get contextual sentence representation c_i^t of S_i:

$$c_i^t = \sigma(W_g g_i^t + \sum_{r\in R}\sum_{k\in N_i^r} \frac{1}{|N_i^r|} W_r g_k^t) \quad (3)$$

where W_g, W_r are model parameters. R is the set of edge types between sentence nodes. N_i^r denotes neighbours of the i-th sentence node of relation r, where $r \in R$. σ is the ReLU function.

Sentence-to-Token Updating. This process is for updating token states. It conveys the sentence-level contextual information c_i^{t-1} to the token, which is also achieved by the SLSTM. At the t-th round, the hidden state of each token $h_{i,j}^t$ is computed as follows:

$$h_{i,j}^t = \text{SLSTM}_{g\to h}(x_{i,j}, c_i^{t-1}, h_{i,j-1}^{t-1}, h_{i,j}^{t-1}, h_{i,j+1}^{t-1}) \quad (4)$$

where $x_{i,j}$ is the initial token embedding. We show the detailed equations of $\text{SLSTM}_{g\to h}$ in Appendix A. Then, the obtained $h_{i,j}^t$ is used for the token-to-sentence updating of the next round.

After T rounds, we get c_{n-1}^T and c_n^T as the final contextual representations of S_{n-1} and S_n, respectively, which fully combine token-level and sentence-level contextual semantics.

2.3 Classification Layer

After obtaining global contextual representations c_{n-1}^T and c_n^T, we use a one-layer BiLSTM (Hochreiter and Schmidhuber, 1997) to encode S_{n-1} into l_{n-1} by concatenating the last hidden states in two directions, and encode S_n into l_n in the same way. l_{n-1} and l_n are local representations without considering wider context. We then concatenate global and local features as follows:

$$X_{cls} = \text{concat}(l_{n-1}, l_n, c_{n-1}^T, c_n^T) \quad (5)$$

X_{cls} is then fed into a two-layer MLP (a fully-connected layer with ReLU activation followed by a softmax output layer) for classification.

Multi-Task Training. Following previous works (Dai and Huang, 2018; Nguyen et al., 2019), we apply multi-task learning to improve the performance. The main task is implicit discourse relation recognition (IDRR), while the auxiliary tasks are explicit discourse relation recognition (EDRR) and connective prediction (CP). These three tasks share the same encoder but use three different MLPs. The objective function is as follows:

$$\begin{aligned} L = &- \alpha \sum_{j=1}^{C_{idrr}} y_{idrr}^j \log \widehat{y}_{idrr}^j - \beta \sum_{j=1}^{C_{edrr}} y_{edrr}^j \log \widehat{y}_{edrr}^j \\ &- \gamma \sum_{j=1}^{C_{cp}} y_{cp}^j \log \widehat{y}_{cp}^j \end{aligned} \quad (6)$$

where α, β, γ are adjustable hyper-parameters. y_{idrr}, y_{edrr} and y_{cp} are ground-truth labels of IDRR, EDRR and CP respectively, while $\widehat{y}_{idrr}$, $\widehat{y}_{edrr}$ and $\widehat{y}_{cp}$ are corresponding predictions. C_{idrr}, C_{edrr} and C_{cp} represent the number of classes of IDRR, EDRR, and CP respectively.

3 Experiment

3.1 Dataset

We conduct experiments on PDTB 2.0 (Prasad et al., 2008), which contains $16,224$ implicit instances and $18,459$ explicit instances. We perform one-vs-others binary classification and 4-way classification on 4 top-level discourse relations: comparison (Comp.), contingency (Cont.), expansion (Exp.), and temporal (Temp.). Following Pitler et al. (2009), we use sections 2-20 for training, sections 21-22 for test and sections 0-1 for validation. The metric is F1 score, and for 4-way classification, we calculate the macro-average F1 score.

3.2 Implementation Details

Details of the PAG. We set the number of sentences to build PAGs as 6, and use zero padding when the text is less than 6 sentences. When building the PAG, we employ spaCy (`https://spacy.io/`) to identify co-reference chains, use simple matching to recognize the same words and use WordNet (Miller, 1995) to recognize synonyms. The WordNet covers 59.38% (7558/12632) training samples, 59.05% (699/1183) developing samples, and 56.98% (596/1046) testing samples. The average number of edges in the PAG is 11.

Details of Parameters and Training. For the node embedding initialization, we use 150-dimensional char embedding obtained by a CNN (Kim, 2014) with kernel window size of [1, 2, 3], 300-dimensional-GloVe embedding, and ELMo with 1024 dimension (the output of the second layer of BiLSTM). We reduce the dimension of node states as 512, so that the dimensions of SLSTM and MR-GCN are also 512. The iteration rounds of the cross-grained updating mechanism is set as 6. The size of the BiLSTM which is used to compute local features (Section 2.3) is 128. For multi-task learning, we set the α, β, γ as 1.0, 0.5, 0.5. The learning rate is 0.001 with batch size of 64. The number of parameters of the CT-Net is about 16M. We use the F1 score as the criterion when manually tuning the hyper-parameter values. The whole model is trained end to end with the ADAM optimizer (Kingma and Ba, 2014) on two Tesla P40s with 24GB GPU memory, and the average runtime is about 6 hours.

Model	Comp.	Cont.	Exp.	Temp.	4-way
NoContext	44.90	53.44	72.20	44.96	51.64
BiLSTM	45.25	53.75	72.66	45.38	51.02
H-LSTM	45.56	53.84	73.23	45.11	51.92
FCG-Net	46.42	54.75	72.43	45.57	52.45
CT-Net	**46.86**	**55.63**	**73.71**	**45.90**	**53.11**

Table 1: Comparison (F1, %) with models using different paragraph encoders (introduced in Section 3.3).

Row	Edge Type			Number of Sentences			F1 (%)
	Adj.	Coref.	Lex.	n=4	n=6	n=8	
1	✗	✓	✓		✓		52.14
2	✓	✗	✓		✓		52.58
3	✓	✓	✗		✓		52.61
4	✓	✓	✓	✓			52.33
5	✓	✓	✓			✓	52.78
6	✓	✓	✓		✓		**53.11**

Table 2: Results of CT-Net with different PAG settings on 4-way classification.

3.3 Results and Discussion

Main Results (Table 1). We carefully design four baselines with different paragraph encoders for a full comparison: (1) "NoContext", the model only using BiLSTM to get local features without considering wider context. (2) "BiLSTM", the model using BiLSTM to encode the paragraph. (3) "H-LSTM", the model using hierarchical BiLSTM as paragraph encoder. (4) "FCG-Net", the model replacing the PAG in the CT-Net with a fully-connected graph (FCG). Except for the way of encoding paragraph, the other settings of these models are the same as the CT-Net. We can draw the following three conclusions. First, "NoContext" obtains the worst performance in most cases, demonstrating the necessity of using contextual representations. Second, the CT-Net gains better performance than models with sequential paragraph encoders "BiLSTM" and "H-LSTM", which proves the superiority of our graph-based CT-Net. The reason is that the CT-Net can track and model closely related context for sentences including long-distance ones. Third, replacing the PAG in the CT-Net with the FCG (FCG-Net) brings a quality drop, which proves the PAG effectively pick out appropriate context that benefits on sentence understanding. We also performed paired t-test between CT-Net and these 4 baselines. The CT-Net is significantly

Model	Comp.	Cont.	Exp.	Temp.	4-way
Chen et al. (2016)	40.17	54.76	-	31.32	-
Qin et al. (2017)	40.87	54.56	72.38	36.20	-
Lan et al. (2017)	40.73	**58.96**	72.47	38.50	47.80
Dai and Huang (2018)	46.79	57.09	70.41	45.61	51.84*
Lei et al. (2018)	43.24	57.82	72.88	29.10	47.15
Bai and Zhao (2018)	47.85	54.47	70.60	36.87	51.06
Nguyen et al. (2019)	**48.44**	56.84	73.66	38.60	53.00
Dai and Huang (2019)	-	-	-	-	52.89
Guo et al. (2020)	42.92	57.67	73.45	36.33	47.90
Ours	46.86	55.63	**73.71**	**45.90**	**53.11**

Table 3: Comparison (F1, %) with existing models on binary and 4-way settings. $*$ means ensemble result.

Row	Multi-Task		F1 (%)
	EDRR	CP	
1	✓	✓	**53.11**
2	✗	✓	51.32
3	✓	✗	51.95

Table 4: Ablation study of multi-task learning on the 4-way classification.

better than all these baselines with $p < 0.05$.

Analysis of the PAG (Table 2). The PAG contains three types of edges: adjacency edge (Adj.), co-reference edge (Coref.) and lexical chain edge (Lex.). To understand the impact of these edges, we conduct ablation experiments on 4-way classification. Rows 1-3 report the results of removing "Adj.", "Coref.", and "Lex." respectively. Removing "Adj." brings the biggest drop (0.97%), which reflects that the adjacency edge plays the most important role in the PAG. We also explore the impact of the number of sentences in the PAG. Rows 4-6 report the results. The CT-Net gains the best performance when the PAG contains 6 sentences, and modeling a longer paragraph of 8 sentences causes a decline. We hypothesize that modeling a paragraph this is too long may introduce some irrelevant context, resulting in a reduction in performance.

Comparison with Existing Systems (Table 3). Table 3 shows the comparison with existing systems. Our method outperforms other models on 4-way classification, and also gains the best performance on the binary classifications of temporal (Temp.) and expansion (Exp.).

Ablation Study of Multi-task Learning (Table 4). Following Dai and Huang (2018) and Nguyen et al. (2019), we utilize the explicit discourse relation recognition (EDRR) and connective prediction (CP) as auxiliary tasks to help implicit discourse relation recognition (IDRR). We conduct ablation experiments of the two auxiliary tasks on 4-way classification (Table 4) to show their impact. Row 1 is the performance of the CT-Net. Rows 2-3 report the performance of removing the auxiliary task. As expected, the EDRR contributes more to the IDRR than the CP does, which is because that the EDRR is a more similar task with the IDRR.

4 Conclusion

We propose a novel graph-based Context Tracking Network (CT-Net) to model the context for implicit discourse relation classification. The CT-Net first converts the paragraph into the paragraph association graph (PAG), where each sentence tracks their appropriate context through different edges, then employs the cross-grained updating mechanism to combine sentence-level and token-level contextual information. Experiments on PDTB 2.0 demonstrate that the CT-Net captures more effective contextual information than carefully designed baselines with different context encoders.

References

Hongxiao Bai and Hai Zhao. 2018. Deep enhanced representation for implicit discourse relation recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 571–583, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jifan Chen, Qi Zhang, Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Implicit discourse relation detection via a deep architecture with gated relevance network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1726–1735, Berlin, Germany. Association for Computational Linguistics.

Zeyu Dai and Ruihong Huang. 2018. Improving implicit discourse relation classification by modeling inter-dependencies of discourse units in a paragraph. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 141–151, New Orleans, Louisiana. Association for Computational Linguistics.

Zeyu Dai and Ruihong Huang. 2019. A regularization approach for incorporating event knowledge and coreference relations into neural discourse parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2976–2987, Hong Kong, China. Association for Computational Linguistics.

Fengyu Guo, Ruifang He, Jianwu Dang, and Jr-Yan Wang. 2020. Working memory-driven neural networks with a novel knowledge enhancement paradigm for implicit discourse relation recognition. In *AAAI 2020*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

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

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Man Lan, Jianxiang Wang, Yuanbin Wu, Zheng-Yu Niu, and Haifeng Wang. 2017. Multi-task attention-based neural networks for implicit discourse relationship representation and identification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1299–1308, Copenhagen, Denmark. Association for Computational Linguistics.

Wenqiang Lei, Yuanxin Xiang, Yuwei Wang, Qian Zhong, Meichun Liu, and Min-Yen Kan. 2018. Linguistic properties matter for implicit discourse relation recognition: Combining semantic interaction, topic continuity and attribution. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yang Liu and Sujian Li. 2016. Recognizing implicit discourse relations via repeated reading: Neural networks with multi-level attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1233, Austin, Texas. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Linh The Nguyen, Linh Van Ngo, Khoat Than, and Thien Huu Nguyen. 2019. Employing the correspondence of relations and connectives to identify implicit discourse relations via label embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4201–4207, Florence, Italy. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric Xing. 2017. Adversarial connective-exploiting networks for implicit discourse relation classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1006–1017, Vancouver, Canada. Association for Computational Linguistics.

Samuel Rönnqvist, Niko Schenk, and Christian Chiarcos. 2017. A recurrent neural model with attention for the recognition of Chinese implicit discourse relations. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 256–262, Vancouver, Canada. Association for Computational Linguistics.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

Wei Shi and Vera Demberg. 2019. Next sentence prediction helps implicit discourse relation classification within and across domains. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5790–5796, Hong Kong, China. Association for Computational Linguistics.

Sheng Xu, Peifeng Li, Fang Kong, Qiaoming Zhu, and Guodong Zhou. 2019. Topic tensor network for implicit discourse relation recognition in Chinese. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 608–618, Florence, Italy. Association for Computational Linguistics.

Yue Zhang, Qi Liu, and Linfeng Song. 2018. Sentence-state LSTM for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 317–327, Melbourne, Australia. Association for Computational Linguistics.

A Sentence-state LSTM

Sentence-state LSTM (SLSTM) (Zhang et al., 2018) is a novel graph RNN. We denote the process of updating sentence states as $\text{SLSTM}_{h\to g}$, the process of updating token states as $\text{SLSTM}_{g\to h}$.

$\text{SLSTM}_{\mathbf{h\to g}}$. At the t-th round, the hidden state of i-th sentence g_i^t is computed based on the values $h_{i,j}^{t-1}$ for all $j \in [0, ..., |S_i|]$:

$$\begin{aligned}
\overline{h}_i &= avg(h_{i,0}^{t-1}, h_{i,1}^{t-1}, ..., h_{i,|S_i|}^{t-1}) \\
\widehat{f}_{g_i}^t &= \sigma(W_g g_i^{t-1} + U_g \overline{h}_i + b_g) \\
\widehat{f}_{i,j}^t &= \sigma(W_f g_i^{t-1} + U_f h_{i,j}^{t-1} + b_f) \\
o_i^t &= \sigma(W_o g_i^{t-1} + U_o \overline{h}_i + b_o) \\
f_{i,0}^t, ...f_{i,|S_i|}^t, f_{g_i}^t &= F_s(\widehat{f}_{i,0}^t, ..., \widehat{f}_{i,|S_i|}^t, \widehat{f}_{g_i}^t) \\
v_{g_i}^t &= f_{g_i}^t \odot v_{g_i}^{t-1} + \sum_j f_{i,j}^t \odot v_{i,j}^{t-1} \\
g_i^t &= o_i^t \odot \tanh(v_{g_i}^t)
\end{aligned} \tag{7}$$

where W_*, U_* and b_* are model parameters, here, $* \in \{g, f, o\}$. $|S_i|$ is the number of tokens of the i-th sentence. $f_{i,0}^t$, ..., $f_{i,|S_i|}$ and $f_{g_i}^t$ are gates controlling information from $v_{i,0}^{t-1}$, ..., $v_{i,|S_i|}^{t-1}$, $v_{g_i}^{t-1}$, respectively. o_i^t is an output gate from the recurrent cell $v_{g_i}^t$ to g_i^t. F_s represents the softmax function.

$\text{SLSTM}_{\mathbf{g\to h}}$. At the t-th round, the hidden state of each token $h_{i,j}^t$ is computed based on the initial input $x_{i,j}$, its hidden state of last round $h_{i,j}^{t-1}$, the hidden states of its neighbors of last round $h_{i,j-1}^{t-1}$, $h_{i,j+1}^{t-1}$ and the contextual representation c_i^{t-1}.

$$\begin{aligned}
\varepsilon_{i,j}^t &= [h_{i,j-1}^{t-1}, h_{i,j}^{t-1}, h_{i,j+1}^{t-1}], \\
\widehat{i}_{i,j}^t &= \sigma(W_i \varepsilon_{i,j}^t + U_i x_{i,j} + V_i c_i^{t-1} + b_i) \\
\widehat{l}_{i,j}^t &= \sigma(W_l \varepsilon_{i,j}^t + U_l x_{i,j} + V_l c_i^{t-1} + b_l) \\
\widehat{r}_{i,j}^t &= \sigma(W_r \varepsilon_{i,j}^t + U_r x_{i,j} + V_r c_i^{t-1} + b_r) \\
\widehat{f}_{i,j}^t &= \sigma(W_f \varepsilon_{i,j}^t + U_f x_{i,j} + V_f c_i^{t-1} + b_f) \\
\widehat{s}_{i,j}^t &= \sigma(W_s \varepsilon_{i,j}^t + U_s x_{i,j} + V_s c_i^{t-1} + b_s) \\
o_{i,j}^t &= \sigma(W_o \varepsilon_{i,j}^t + U_o x_{i,j} + V_o c_i^{t-1} + b_o) \\
u_{i,j}^t &= \tanh(W_u \varepsilon_{i,j}^t + U_u x_{i,j} + V_u c_i^{t-1} + b_u) \\
i_{i,j}^t, l_{i,j}^t, r_{i,j}^t, f_{i,j}^t, s_{i,j}^t &= F_s(\widehat{i}_{i,j}^t, \widehat{l}_{i,j}^t, \widehat{r}_{i,j}^t, \widehat{f}_{i,j}^t, \widehat{s}_{i,j}^t) \\
v_{i,j}^t &= l_{i,j}^t \odot v_{i,j-1}^{t-1} + f_{i,j}^t \odot v_{i,j}^{t-1} + r_{i,j}^t \odot v_{i,j+1}^{t-1} \\
&\quad + s_{i,j}^t \odot v_{g_i}^{t-1} + i_{i,j}^t \odot u_{i,j}^t \\
h_{i,j}^t &= o_{i,j}^t \odot \tanh(v_{i,j}^t)
\end{aligned} \tag{8}$$

where W_*, U_* and b_* are model parameters, here, $* \in \{i, l, r, f, s, o\}$. F_s represents the softmax function, and σ represents the sigmoid function. $\widehat{i}_{i,j}^t$, $\widehat{l}_{i,j}^t$, $\widehat{r}_{i,j}^t$, $\widehat{r}_{i,j}^t$,$\widehat{f}_{i,j}^t$ are gates conveying information from the $\varepsilon_{i,j}^t$ and $x_{i,j}$ to the cell state $v_{i,j}^t$, which are normalised. o_t^i is an output gate from the cell $v_{i,j}^t$ to the hidden state $h_{i,j}^t$.