Pair-Aware Neural Sentence Modeling for Implicit Discourse Relation Classification

Deng Cai[1,2] and Hai Zhao[1,2](✉)

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
thisisjcykcd@gmail.com, zhaohai@cs.sjtu.edu.cn
[2] Key Lab of Shanghai Education Commision for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China

Abstract. Implicit discourse relation recognition is an extremely challenging task, for it lacks of explicit connectives between two arguments. Currently, most methods to address this problem can be regarded as to solve it in two stages, the first is to extract features from two arguments separately, and the next is to apply those features to some standard classifier. However, during the first stage, those methods neglect the links between two arguments and thus are blind to find pair-specified clues at the very beginning. This paper therefore makes an attempt to model sentence with its targeted pair in mind. Concretely, an LSTM model with attention mechanism is adapted to accomplish this idea. Experiments on the benchmark dataset show that without the help of feature engineering or any external linguistic knowledge, our proposed model outperforms previous state-of-the-art systems.

1 Introduction

Discourse parsing has been shown helpful for many downstream natural language process (NLP) tasks, such as summarization, question answering. While recently the field of discourse parsing has been widely studied, implicit discourse relation classification remains a significant challenge and becomes a performance bottleneck of such systems [9,11]. The main reason that makes implicit discourse relation classification so difficult is that the absence of discourse connectives (e.g., *so*, *but* et al.), which can explicitly indicate the relation between its governed arguments with few ambiguousness [18,21]. In other words, implicit discourse relation classification requires semantic understanding of both two text arguments [6].

This paper was partially supported by Cai Yuanpei Program (CSC No. 201304490199 and No. 201304490171), National Natural Science Foundation of China (No. 61170114, No. 61672343 and No. 61272248), National Basic Research Program of China (No. 2013CB329401), Major Basic Research Program of Shanghai Science and Technology Committee (No. 15JC1400103), Artand Science Interdisciplinary Funds of Shanghai Jiao Tong University (No. 14JCRZ04), and Key Project of National Society Science Foundation of China (No. 15-ZDA041).

S. Benferhat et al. (Eds.): IEA/AIE 2017, Part II, LNAI 10351, pp. 458–466, 2017.
DOI: 10.1007/978-3-319-60045-1_47

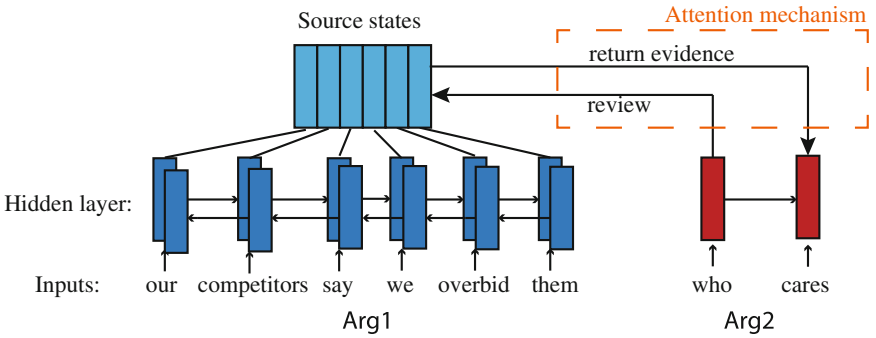

Fig. 1. Illustration of our model. The processes of modeling two **Arg**uments are connected via an attention mechanism.

Until now, related works focus on identifying an ideal set of features to better represent two arguments, such as exploiting handcrafted features obtained from external linguistic knowledge [9,12,14,16,20,27,30], feature combination optimization [19], and using neural network for automatic feature learning [32]. Data selection or augmentation is also applied [3,10,31].

However, in previous works, either of two arguments is simply transformed into vector representation according to feature designing individually [22–24]. The modeling processes of two arguments are parallel running and independent to each other. For one who is asked what the relation is between the current and the previous sentences, one highly possible strategy is to look back into the previous sentence and find some relevant evidence to make the decision, i.e., goal-directed observation makes better judgment. Moreover, as plain text of arguments can be so long that redundant and needless words may also be contained, conventional models can easily overfit on training corpus. Previous work has primarily applied attentive neural models to generating text for their capability of target-guided feature extraction, advancing several fields such as machine translation [1,15] and sentence summarization [26]. In this work, we extends these techniques to modeling text pair for discourse relation classification. Specifically, we adapt an LSTM model with attention mechanism for sentence modeling in implicit discourse relation classification, which intends to filter out useless information and capture critical evidence via pair-guided feature extraction.

2 Model

The overall model architecture is illustrated in Fig. 1. Without any feature engineering, we just use the original word information as input. Those symbolic data will first be transformed into distributed vectors (word embeddings) [2] through an embedding layer.

Bi-LSTM Sentence Modeling. LSTM [8] is a variant of recurrent neural networks (RNNs) which has been shown to be an effective tool for sequence modeling tasks [4,13,29]. Unlike classic bag-of-words model, LSTM constructs sentence representations as an order-sensitive function. At each time step t, LSTM uses a *memory cell* $\mathbf{c}_t \in \mathbb{R}^H$ to preserve history information and output a *hidden state* $\mathbf{h}_t \in \mathbb{R}^H$ as the current sentence representation (Due to the structure of LSTM, it tends to focus on more recent inputs). The transition equations are the following:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}^i\mathbf{x}_t + \mathbf{U}^i\mathbf{h}_{t-1} + \mathbf{b}^i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}^f\mathbf{x}_t + \mathbf{U}^f\mathbf{h}_{t-1} + \mathbf{b}^f) \\
\mathbf{o}_t &= \sigma(\mathbf{W}^o\mathbf{x}_t + \mathbf{U}^o\mathbf{h}_{t-1} + \mathbf{b}^o) \\
\hat{\mathbf{c}}_t &= \tanh(\mathbf{W}^c\mathbf{x}_t + \mathbf{U}^c\mathbf{h}_{t-1} + \mathbf{b}^c) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}$$

where x_t is the input at the current time step (e.g., t-th word representation), σ denotes the sigmoid function and $\odot$ denotes element-wise multiplication.

To fully capture the semantics of natural language, a bidirectional LSTM [7] is used to modeling the first argument (Arg1) which consists of two LSTMs: one takes the input word sequence in its original order and the other takes the sequence in the reverse order. Therefore, the outputs of bi-LSTM include a sequence of *forward hidden states* $(\overrightarrow{\mathbf{h}_1}, \ldots, \overrightarrow{\mathbf{h}_{T_1}})$ and a sequence of *backward hidden states* $(\overleftarrow{\mathbf{h}_1}, \ldots, \overleftarrow{\mathbf{h}_{T_1}})$, where T_1 is the length of Arg1. We then concatenate those two sequence into one sequence as $\bar{\mathbf{h}}_j = [\overrightarrow{\mathbf{h}_j}^T; \overleftarrow{\mathbf{h}_j}^T]^T$. In this way, each annotation $\bar{\mathbf{h}}_i$ contains summarized information about the whole input sentence, but with a strong attention to the details surrounding the i-th word. The resulted vectors $(\bar{\mathbf{h}}_1, \ldots, \bar{\mathbf{h}}_{T_1})$ not only stand as a representation of Arg1 but also serve as an information source, so called *source states*, to the followed modeling of the second argument (Arg2).

Attentive LSTM Sentence Modeling. To generate the representation of Arg2, one can also use an LSTM to achieve it (See Sect. 3). However, since we already have the representation of Arg1 and the ultimate goal is to classify the relation between those two arguments, it should be better to make more purposeful feature extraction so that more targeted evidence could be detected and focused. In our model, an attentive LSTM [1] is adapted to fulfill this requirement, which uses the source states as another input. Concretely, at each time step t, the actual input to feed the attentive LSTM is calculated as:

$$\mathbf{x}_t = g(\mathbf{c}_t, \mathbf{w}_t)$$

where $g(\cdot)$ is a nonlinear, potentially multi-layered function that mixes the information of $\mathbf{c}_t$ and $\mathbf{w}_t$, while $\mathbf{c}_t$ is the *collaborate vector* detected from source states and $\mathbf{w}_t$ is the word representation of the t-th word in Arg2.

The collaborate vector is computed as weighted sum of source states $\bar{\mathbf{h}}_i$:

$$\mathbf{c}_t = \sum_{j=1}^{T_1} \alpha_{tj} \bar{\mathbf{h}}_j$$

the weight α_{tj} of source state $\bar{\mathbf{h}}_j$ is computed by:

$$\alpha_{tj} = \frac{\exp(s(\mathbf{h}_{t-1}, \bar{\mathbf{h}}_j))}{\sum_{i=1}^{T_1} \exp(s(\mathbf{h}_{t-1}, \bar{\mathbf{h}}_i))}$$

where $s()$ is a function to score the importance of each source state based on previous hidden state $\mathbf{h}_{t-1}$:

$$s(\mathbf{h}_{t-1}, \bar{\mathbf{h}}_j) = \mathbf{v}_\alpha^T \tanh(\mathbf{W}_\alpha [\mathbf{h}_{t-1}^T; \bar{\mathbf{h}}_j^T]^T + \mathbf{b}_\alpha)$$

where $\mathbf{v}_\alpha, \mathbf{b}_\alpha \in \mathbb{R}^V$ and $\mathbf{W}_\alpha \in \mathbb{R}^{V \times (H_1 + H_2)}$ are trainable parameters, H_1, H_2 are the dimensionality of source states and hidden states, respectively. Intuitively, this score function implements a mechanism of attentive comparison between the both details of arguments.

After obtaining the hidden states of attentive LSTM, i.e., $(\mathbf{h}_1, \ldots, \mathbf{h}_{T_2})$, to deal with variable argument lengths and reduce the dimensionality for final classification, we average the vector representations of both arguments and concatenate them into one vector:

$$\mathbf{h}^* = [(\frac{1}{T_1} \sum_{i=1}^{T_1} \bar{\mathbf{h}}_i)^T; (\frac{1}{T_2} \sum_{j=1}^{T_2} \mathbf{h}_j)^T]^T$$

where $\mathbf{h}^*$ is the final hidden layer representation of both arguments. Upon the hidden layer, we stack a Softmax layer for relation classification. During training, the traditional cross-entropy error combined with an ℓ_2 regularization is used as the loss function:

$$J(\theta) = \frac{1}{m} \sum_{k=1}^{m} - \log \text{Softmax}_{y_{(k)}}(\mathbf{h}^*_{(k)}) + \frac{\lambda}{2} ||\theta||_2^2$$

where m is the size of training set, $y_{(k)}$, $\mathbf{h}^*_{(k)}$ are the golden relation and final representation for the k-th training instance respectively, λ is the regularization coefficient and θ is the parameter set in our model. The diagonal variant of AdaGrad [5] with minibatchs is used for the training procedure.

3 Experiments

To evaluate the proposed model, we conducted a series of experiments on the Penn Discourse Treebank (PDTB) dataset [25]. Following the conventions of most previous works, we used sections 2–20 in the PDTB as training set, sections 0–1 as development set for hyper-parameter tuning and sections 21–22 as test set.

Table 1. Distribution of the second level relation types of implicit relations from training sections.

Level 1 class	Level 2 type	Training instances	%
Comparison	Concession	184	1.43
	Contrast	1610	12.54
	Pragmatic concession	1	0.01
	Pragmatic contrast	4	0.03
Contingency	Cause	3277	25.53
	Condition	1	0.01
	Pragmatic cause	64	0.50
	Pragmatic condition	1	0.01
Expansion	Alternative	151	1.18
	Conjunction	2882	22.46
	Exception	2	0.02
	Instantiation	1102	8.59
	List	338	2.63
	Restatement	2458	19.15
Temporal	Asynchronous	555	4.32
	Synchrony	204	1.59

Setup. The PDTB [25] provides a multi-level hierarchy of discourse relations. The first level roughly categorizes the relations into four major classes. For each class, a second level of types is available to make more distinct and pragmatic description on the relation. However, most of recent works only concern about recognizing of the first level classes, in the "one-versus-all" binary classification setting (in fact, at present only two papers are available presenting results on second level classification). The neglect of deeper processing may be due to the following reasons: (1) the distribution of second level discourse relation is unbalanced; (2) the training instances for each relation type are relatively small. The distribution of 16 second level relation types of implicit relations from training sections is shown in Table 1.

In this paper, we attack the second level relation classification as it is more challenging but more relevant for the ultimate use of discourse parsing, and the more general multi-classes classification setting is adopted.

Implement Details. Pre-training the word embeddings on large unlabeled data has been found to benefit the performance of neural network models on many tasks. We therefore use word2vec[1] [17] toolkit to initialize the word embedding matrix $\mathbf{M}$. According to early experiments on development set, we empirically set $d = 300$, $H_1 = 300$, $H_2 = 300$ and $V = 100$. We also found that dropout

[1] http://code.google.com/p/word2vec/.

Table 2. Performance comparisons of baseline models.

Models	Accuracy (%)
Most common class	26.63
Arg1 only	36.38
Arg2 only	40.81
Arg1 + Arg2	42.93
Arg2 + attentive Arg1	45.14
Arg1 + attentive Arg2	45.81

Table 3. Comparisons with previous models.

Models	Accuracy (%)
[12]	–
+ Surface features	40.20
+ Brown cluster	40.66
[9]	36.98
+ Surface features	43.75
+ Entity semantics	37.63
+ Both	44.59
This work	**45.81**

[28] on the embedding layer with dropout rate 0.5 can significantly improve the overall performance.

Model Analysis. To reveal the effect of pair-aware sentence modeling, we re-implemented several simplified versions of our model as baseline models: one without the attention mechanism, one only uses features in Arg1 and the other only uses Arg2. We also tried swapping the positions of two arguments. The results are listed in Table 2. As we can see, the performance is significantly boosted by exploiting pair-aware sentence modeling. Moreover, it is interesting to see that the Arg2-only one yields similar results compared to none attention model, and attentive model has substantial improvements on both of them. It demonstrates that pair-specific evidence can be easily ignored without guiding information.

Results. The comparisons of previous models and our model are shown in Table 3. Note that previous methods exploited massive hand-crafted *surface features* (word pair features, constituent parse features, dependency parse features, and contextual features), or other external linguistic knowledge such as Brown clusters and entity semantics [9], while our proposed model simply uses word information. However, with the ability of extracting pair-specified features, our model achieves even better results against them.

4 Conclusion and Future Work

This paper presents a pair-aware sentence modeling method for implicit discourse relation classification. Unlike previous works, the proposed model generates sentence representations via pair-specified feature extraction. Experiments on benchmark dataset show that with an attention mechanism, our model achieves improved performance over baseline models and outperforms previous state-of-the-art methods in the way without any feature engineering.

Although the proposed method is originally designed for implicit discourse relation, it can be easily generalized to other text relation classification tasks, such as paraphrase detection.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
3. Braud, C., Denis, P.: Combining natural and artificial examples to improve implicit discourse relation identification. In: Proceedings of the 25th International Conference on Computational Linguistics: Technical papers, Dublin, Ireland, pp. 1694–1705 (2014)
4. Cai, D., Zhao, H.: Neural word segmentation learning for Chinese. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, vol. 1, Long Papers, pp. 409–420 (2016)
5. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
6. Forbes-Riley, K., Webber, B., Joshi, A.: Computing discourse semantics: the predicate-argument semantics of discourse connectives in D-LTAG. J. Semant. **23**(1), 55–106 (2006)
7. Graves, A., Jaitly, N., Mohamed, A.R.: Hybrid speech recognition with deep bidirectional LSTM. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 273–278 (2013)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Ji, Y., Eisenstein, J.: One vector is not enough: entity-augmented distributed semantics for discourse relations. Transactions of the Association for Computational Linguistics (2015)
10. Lan, M., Xu, Y., Niu, Z.: Leveraging synthetic discourse data via multi-task learning for implicit discourse relation recognition. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, vol. 1, Long Papers, pp. 476–485 (2013)
11. Li, Z., Zhao, H., Pang, C., Wang, L., Wang, H.: A constituent syntactic parse tree based discourse parser. In: Proceedings of the CoNLL-16 Shared Task, pp. 60–64 (2016)
12. Lin, Z., Kan, M.Y., Ng, H.T.: Recognizing implicit discourse relations in the Penn discourse treebank. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 1, pp. 343–351 (2009)

13. Liu, P., Qiu, X., Chen, X., Wu, S., Huang, X.: Multi-timescale long short-term memory neural network for modelling sentences and documents. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2326–2335 (2015)
14. Louis, A., Joshi, A., Prasad, R., Nenkova, A.: Using entity features to classify implicit discourse relations. In: Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 59–62 (2010)
15. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 1412–1421 (2015)
16. McKeown, K., Biran, O.: Aggregated word pair features for implicit discourse relation disambiguation. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 69–73 (2013)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
18. Miltsakaki, E., Dinesh, N., Prasad, R., Joshi, A., Webber, B.: Experiments on sense annotations and sense disambiguation of discourse connectives. In: Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories, Barcelona, Spain, December 2005
19. Park, J., Cardie, C.: Improving implicit discourse relation recognition through feature set optimization. In: Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 108–112 (2012)
20. Pitler, E., Louis, A., Nenkova, A.: 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, vol. 2, pp. 683–691 (2009)
21. Pitler, E., Raghupathy, M., Mehta, H., Nenkova, A., Lee, A., Joshi, A.: Easily identifiable discourse relations. In: Proceedings of the 22nd International Conference on Computational Linguistics, Manchester, UK, pp. 87–90 (2008)
22. Qin, L., Zhang, Z., Zhao, H.: Implicit discourse relation recognition with contextaware character-enhanced embeddings. In: the 26th International Conference on Computational Linguistics, Osaka, Japan, December 2016
23. Qin, L., Zhang, Z., Zhao, H.: Shallow discourse parsing using convolutional neural network. In: Proceedings of the CoNLL-16 Shared Task, pp. 70–77 (2016)
24. Qin, L., Zhang, Z., Zhao, H.: A stacking gated neural architecture for implicit discourse relation classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, USA, November 2016
25. Prasad, R., Nikhil Dinesh, A., Webber, B.: The Penn discourse treebank 2.0. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation. Marrakech, Morocco (2008)
26. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 379–389 (2015)
27. Rutherford, A., Xue, N.: Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, pp. 645–654 (2014)
28. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)

29. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
30. Versley, Y.: Subgraph-based classification of explicit and implicit discourse relations. In: Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)-Long Papers, pp. 264–275 (2013)
31. Wang, X., Li, S., Li, J., Li, W.: Implicit discourse relation recognition by selecting typical training examples. In: Proceedings of the 24th International Conference on Computational Linguistics: Technical papers, pp. 2757–2772 (2012)
32. Zhang, B., Su, J., Xiong, D., Lu, Y., Duan, H., Yao, J.: Shallow convolutional neural network for implicit discourse relation recognition. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 2230–2235 (2015)