Comparative Investigation of Deep Learning Components for End-to-end Implicit Discourse Relationship Parser

Dejian Li, Man Lan*, and Yuanbin Wu*

School of Computer Science, East China Normal University, P.R.China
{51194506071}@stu.ecnu.edu.cn ; {mlan,ybwu}@cs.ecnu.edu.cn

Abstract. The neural components in deep learning framework are crucial for the performance of many natural language processing tasks. So far there is no systematic work to investigate the influence of neural components on the performance of implicit discourse relation recognition. To address it, in this work we compare many different components and build two implicit discourse parsers base on the sequence and structure of sentence respectively. Experimental results show due to different linguistic features, the neural components have different effects in English and Chinese. Besides, our models achieve state-of-the-art performance on CoNLL-2016 English and Chinese datasets.

Keywords: Deep learning · Implicit discourse relation classification · Word embedding · Neural network.

1 Introduction

Discourse consists of a series of consecutive text units, such as clauses, sentences or paragraphs. They are coherent both in form and content, conveying the complete information together. Discourse relation (e.g., *Contrast, Conjunction*) is the semantic logic relationship between two text units. Discourse relation recognition benefits many downstream NLP tasks such as Sentiment Analysis [28], Machine Translation [7] and Summarization [24], etc.

Discourse relation can be divided into explicit discourse relation and implicit discourse relation according to whether the arguments contain discourse connectives. The recognition of explicit discourse relationship reaches 93% accuracy by using only discourse connectives [14], but the performance of implicit discourse relationship recognition is always poor due to the lack of discourse connectives, which is the bottleneck of the whole discourse parser. In order to fix this problem, early researchers designed many complex features with expert knowledge. [1] used an aggregated approach to word pair features and [18] employed Brown cluster pairs to represent discourse relation. However, this method performs badly in generalization.

With the development of deep learning in the NLP field, researchers begin to use this method to recognize implicit discourse relation. Their methods can

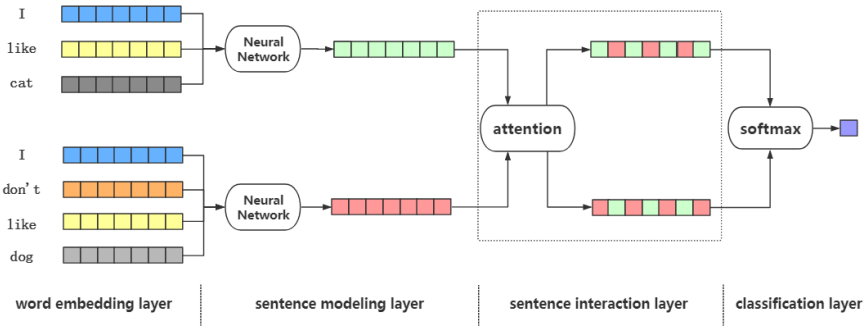

Fig. 1. Architecture of our implicit discourse relation parser system.

be divided into two lines in general. One research line is to learn from explicit discourse relation. [2, 19, 6] tried to expand the implicit training dataset with the help of the discourse connectives. [26] learned discourse-specific word embedding from massive explicit data. [16] presented their implicit network to learn from another neural network which has access to connectives. The other line focuses on the expression of words and the structure of the model. [23, 15] used word2vec word embedding and Convolutional Neural Network (CNN) to determine the senses. [8] used CNN to model argument pairs with GloVe word embedding and multi-task learning system. [3] used BiLSTM to model the sentences and adopted gated relevance network to calculate the relevance score between two arguments. [25] employed new network structure TreeLSTM to model the sentences.

However, with the emergence of the new word embeddings (e.g., ELMo, BERT) and neural network models, there is no systematic research work to deeply analyze the influence of each component on the performance of the parser in the deep learning method. To fill this gap we conduct comparative experiments from three aspects which are word embedding layer, sentence modeling layer and sentence interaction layer. Specifically, we construct two parsers which are different in sentence modeling. One parser focuses on sentence order and the other focuses on the sentence grammar. Besides, we select four word embeddings and two ways of sentence interactions. We conduct our experiment both in English and Chinese corpora to verify the semantic expression of components in different languages.

2 Implicit Discourse Relation Parser

Our comparative study is based on deep learning framework. We aim to compare the different components to find the key influencing factors. The proposed implicit discourse parser contains four independent components as shown in Figure 1. First,the word embedding layer converts each single word into word vector. Then the sentence modeling layer makes semantic modeling and obtains the semantic vector expression of the sentence. Later, two sentences interact with each

other to obtain semantic information in the sentence interaction layer. Finally, the predictions are obtained in classification layer by using *softmax* function.

2.1 Word Embedding Layer

In deep learning framework, the pre-trained models play an important role because the exciting performance of deep learning relies on the training in large corpus. Word embedding is the first and crucial step in deep learning framework, which transforms the natural language into word vector as the input of the neural network. Different pre-trained word vector models are chosen to verify whether there is any loss or misinterpretation between the conversion.

We convert each word $\boldsymbol{w}$ in the argument into word vector $\boldsymbol{x} \in \mathbb{R}^{d_w}$, where d_w is the dimension of the word vector. Let $\boldsymbol{x}_i^1 \left(\boldsymbol{x}_i^2\right)$ be the i-th word vector in Arg-1(Arg-2), then the two discourse arguments are represented as:

$$Arg\text{-}1 : \left[\boldsymbol{x}_1^1, \boldsymbol{x}_2^1, \cdots, \boldsymbol{x}_{L_1}^1\right] \tag{1}$$

$$Arg\text{-}2 : \left[\boldsymbol{x}_1^2, \boldsymbol{x}_2^2, \cdots, \boldsymbol{x}_{L_2}^2\right] \tag{2}$$

where Arg-1(Arg-2) has $L_1(L_2)$words. Generally, the word embeddings are pre-trained on large corpus and supposed to contain latent semantic and syntactic information. In recent years several supreme word embeddings have been presented by researchers. To examine their different effectiveness in word conversion, we choose two types of pre-trained word vector models, i.e., context-free models and contextual models.

Context-free models generate a "word embedding" representation for each word in the vocabulary. This means the vector representation of the word in argument has no relation with the specific context of this argument. Here we choose word2vec [11] and GloVe [12] models which are widely used. The word2vec uses local text which is controlled by window size from large corpus to train the word vector. While GloVe is trained on aggregated global word-word co-occurrence statistics from the corpus.

Contextual models generate a representation of each word which is based on the other words in the sentence. It is usually pre-trained on large corpus by learning the language model rather than the "word embedding". Thus for specific sentence, contextual model generates the word representation base on its context. Here we choose ELMo[13] and BERT[5] models.

2.2 Sentence Modeling Layer

After the word embedding layer, we get the sentence representation in a word vector matrix. This only represents the information of source corpus rather than the target recognition task. Therefore, in order to better fit the task, we use the neural network models to convert the vector matrix into the semantic representation of the arguments.

Considering that both sentence order and grammar are important in understanding the semantics of sentences, so we choose two types of sentence modeling

methods, i.e., sequential relation modeling focusing on sentence order and structural relation modeling focusing on sentence grammar.

- **Sequential relation modeling**: We select three representative sequential models: Long short Term Memory (LSTM), Bi-directional Long Short Term Memory (BiLSTM) and Convolutional Neural Network (CNN).
- **Structural relation modeling**: We use Tree-LSTM [22] to capture the structure relation information, which is the combination of LSTM and tree structured neural networks. Here we choose Child-Sum-Tree-LSTM and Binary-Tree-LSTM. The former has flexible number of child nodes in its tree while the latter has only two child nodes for each father node. The above two tree structures come from the sentence constituency parse tree which is obtained by the Stanford CoreNLP toolkit [9].

Given the two arguments representations as Formula (1) and (2), the LSTM computes the state sequence $[\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_L]$ for each time step i using the following formulate:

$$\boldsymbol{i}_i = \sigma(\boldsymbol{W}_i[\boldsymbol{x}_i, \boldsymbol{h}_{i-1}] + \boldsymbol{b}_i) \tag{3}$$

$$\boldsymbol{f}_i = \sigma(\boldsymbol{W}_f[\boldsymbol{x}_i, \boldsymbol{h}_{i-1}] + \boldsymbol{b}_f) \tag{4}$$

$$\boldsymbol{o}_i = \sigma(\boldsymbol{W}_o[\boldsymbol{x}_i, \boldsymbol{h}_{i-1}] + \boldsymbol{b}_o) \tag{5}$$

$$\tilde{\boldsymbol{c}}_i = tanh(\boldsymbol{W}_c[\boldsymbol{x}_i, \boldsymbol{h}_{i-1}] + \boldsymbol{b}_c) \tag{6}$$

$$\boldsymbol{c}_i = \boldsymbol{i}_i \odot \tilde{\boldsymbol{c}}_i + \boldsymbol{f}_i \odot \boldsymbol{c}_{i-1} \tag{7}$$

$$\boldsymbol{h}_i = \boldsymbol{o}_i \odot tanh(\boldsymbol{c}_i) \tag{8}$$

where σ denotes the *sigmoid* function and $\odot$ denotes element-wise multiplication. On this basis, BiLSTM get the information from both past and future rather than only from the past in LSTM. Therefore, at each position i of the sequence, we obtain two states $\overrightarrow{\boldsymbol{h}}_i$ and $\overleftarrow{\boldsymbol{h}}_i$, where $\overrightarrow{\boldsymbol{h}}_i, \overleftarrow{\boldsymbol{h}}_i \in \mathbb{R}^{d_h}$. Then we concatenate them to get the intermediate state, i.e. $\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i, \overleftarrow{\boldsymbol{h}}_i]$. After that, we sum up the sequence states $[\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_L]$ to get the representations of *Arg*-1 and *Arg*-2 respectively as follows:

$$\boldsymbol{R}_{Arg_1} = \sum_{i=1}^{L_1} \boldsymbol{h}_i^1 \tag{9}$$

$$\boldsymbol{R}_{Arg_2} = \sum_{i=1}^{L_2} \boldsymbol{h}_i^2 \tag{10}$$

As for CNN model, we use $\boldsymbol{Arg}[i:j]$ to represent the sub-matrix of $\boldsymbol{Arg}$ from row i to row j. A convolution involves a filter $\mathbf{w} \in \mathbb{R}^{h \times d}$ (h is the height of filter and d is the dimensionality of the word vector). The output sequence o_i of the convolution operator is obtained by repeatedly applying the filter on sub-matrices of $\boldsymbol{Arg}$:

$$o_i = \mathbf{w} \cdot \mathbf{Arg}[i : i + h - 1] \tag{11}$$

where i=1...$s-h+1$. A bias term $b \in \mathbb{R}$ and an activation function f are added to each o_i to compute the feature map c_i for this filter:

$$c_i = f\left(o_i + b\right) \tag{12}$$

Then we use *max pooling* operation to get the representation of the argument:

$$\boldsymbol{R_{arg}} = max\{c_i\} \tag{13}$$

2.3 Sentence Interaction Layer

After the sentence modeling layer, the representation of two arguments is still isolate. This is not what we expected. Since discourse relationship is annotated by the two arguments rather than one single argument, we suppose that the interaction relationship between two arguments is helpful to the discourse relation recognition. In order to obtain argument interaction representation, in this work we choose the following interaction mechanisms:

- Attention: Two sentence vectors do Attention operation, then concatenated together
- Con-self-Attention: Two vectors concatenated, then do self-Attention operation
- Self-Attention-con: Two sentence vectors do self-Attention operation respectively, then concatenated together
- Attention-mlp: Two sentence vectors do Attention interact operation, then put into Multi-Layer Perceptron (MLP)

Through above interactions, the two separate representations, i.e., $\boldsymbol{R_{Arg1}}$, $\boldsymbol{R_{Arg2}}$, become joint pair representation $\boldsymbol{R_{pair}}$ which contains the overall information of the two arguments. Finally we feed the $\boldsymbol{R}_{pair}$ into a full-connected *softmax* layer to make sense prediction.

3 Experiment

3.1 Dataset

The Penn Discourse Treebank (PDTB) and the Chinese Discourse Treebank (CDTB) are the most widely used discourse datasets in English and Chinese, respectively. To make comparison reasonable, we use the adapted version of the data provided by CoNLL-2016 Shared Task. Table 1 show the distributions of the two datasets.

Except Explicit and EntRel, we extract remaining relations as our experimental implicit dataset in English. Our experiments focus on multi-class classification on the four top-level classes. The statistics of these four labels are shown in Table 2. As for Chinese, we follow previous research in [17, 23] and select the non-Explicit (i.e., Implicit, EntRel and AltLex) samples as our dataset. The statistics of Chinese data is shown in Table 3.

Table 1. The distribution of discourse relation types in the English and Chinese data.

	English		Chinese	
	amount	percentage(%)	amount	percentage(%)
Explicit	18,459	45.5	2,398	21.75
Implicit	16,053	39.5	7,238	65.66
EntRel	5,210	12.8	1,219	11.06
AltLex	624	1.5	223	2.02
NoRel	254	0.6	0	0
Total	40,600	100	11,023	100

Table 2. Statistics of CoNLL-2016 English implicit discourse sense.

Sense Label	Train Set	Development Set	Test Set
Comparison	2,035	96	134
Contingency	3,720	134	221
Expansion	7,378	286	434
Temporal	849	51	20
Total	13,982	567	809

Table 3. Statistics of CoNLL-2016 Chinese non-Explicit discourse sense.

Sense Label	Training	Development	Test
Conjunction	5,196	189	228
Expansion	1,228	49	40
EntRel	1,098	50	71
Causation	260	12	11
Purpose	79	2	6
Contrast	72	3	1
Temporal	36	0	1
Conditional	32	1	1
Progression	14	0	0
Total	8,015	306	359

3.2 Experiment Setup

We choose cross-entropy loss function and Adam with a learning rate of 0.001 and a mini-batch size of 64 to train the model. Follow previous work, we use macro-F_1 to evaluate the performance in English and accuracy to evaluate performance in Chinese since the Chinese corpus is skewed distributed and the macro-F_1 is prone to be affected by the uneven samples. In CNN model, we choose filter window size (1, 3, 5) to represent the *unigram, trigram* and *5-gram* features in sentence. And we set hidden size as 50 in LSTM, BiLSTM and TreeLSTM

models. We applied dropout before the classification layer and set the dropout rate as 0.5.

In context-free word embedding, we select different models for English and Chinese. For English experiment, we train two word2vec models as follows:

- BLLIP-50d: The 50-dimensional word vector trained on BLLIP [10]
- Google-300d: The 300-dimensional word vector trained on 100 billion words from Google News

and select four GloVe models[1] which are different in vocabulary size and training corpus as follows:

- 6B-50d/100d/300d: 6B tokens, 400K vocab, uncased, 50/100/300 dimensions
- 840B-300d: 840B tokens, 2.2M vocab, cased, 300 dimensions

For Chinese, we use the Tagged Chinese Gigaword[2] to train 300-dimension word2vec and GloVe word embeddings.

In the contextual model, we use ELMo tool provided by allennlp[3] and the three layers respectively as word representations both for English and Chinese experiment. As for BERT, we choose four models for English:

- Base-single/pairs: 12-layer, 768-hidden, encode single sentence/sentence pairs
- Large-single/pairs: 24-layer, 1024-hidden, encode single sentence/sentence pairs

and one pre-trained Chinese model provided by Google with 12-layer, 768-hidden, 12-heads and 110M parameters.

3.3 Results

We perform a series of comparison experiments to explore the influence of each component on the performance of the parser.

First, we design experiments to evaluate the impact of word embeddings. Table 4 and 5 show the performance comparison of context-free models and contextual models for English. Here we use the sequence sentence modeling without sentence interaction layer. From the two tables, we see that contextual word embeddings perform better than context-free embeddings in English. The best performance of contextual models (BERT, 51.18%) is 7.64% higher than context-free models' best performance (GloVe, 43.54%).

The Chinese result are shown in Table 6. It is clear that the best performance of contextual models outperform the context-free model. Due to the language difference and the ELMo and BERT generate the "single Chinese character" embeddings rather than the "word" embeddings, the performance gap between contextual models and context-free models in Chinese is not as much as that in English. From Table 4 and 5, we state that context information in embeddings is helpful to the implicit relation recognition both in English and Chinese.

[1] `https://nlp.stanford.edu/projects/GloVe/`
[2] `https://catalog.ldc.upenn.edu/LDC2007T03`
[3] `https://github.com/allenai/allennlp`

Table 4. Comparisons of F_1 scores (%) for English context-free word embedding with sequence sentence modeling.

	word2vec		*GloVe*			
	BLLIP-50d	Google-300d	6B-50d	6B-100d	6B-300d	840B-300d
LSTM	41.75	39.41	31.61	38.19	43.01	43.08
BiLSTM	40.12	40.29	33.22	38.88	41.54	**43.54**
CNN	39.29	36.44	37.09	40.35	38.11	40.06

Table 5. Comparisons of F_1 scores (%) for English contextual word embedding with sequence sentence modeling.

	ELMo			*BERT*			
	1	2	3	Base-single	Base-pairs	Large-single	Large-pairs
LSTM	42.19	45.62	44.97	45.24	**51.18**	44.01	50.24
BiLSTM	41.1	48.09	45.97	45.17	49.5	46.32	47.59
CNN	42.59	46.28	46.28	44.27	46.31	45.22	44.14

Next, we evaluate the effects of sentence modeling in sequence and structure on parser performance. Table 7 show the comparison of best sequential models from previous experiments and structural models in English, where GloVe is 840B-300d, ELMo is the 2nd layers, and BERT is Base-single. We see that the Binary-Tree-LSTM outperforms sequential models. This proves that after adding the grammatical parsing information of the sentence, the tree model is able to capture the semantics of the sentence more effectively. In Chinese, we choose ELMo and BERT as embeddings and list the results in Table 6. It is surprising that CNN outperforms BiLSTM and Tree-LSTM, which is conflict with the finding in English. This may result from the different characteristic in languages.

Further, we select several representative models to examine the sentence interaction layer for experiments. For English experiment, we find that the model is unable to fit the training data due to complexity increasing of the model after adding the interaction layer. So we adjusted the dropout values to fit the

Table 6. Comparisons of Accuracy(%) for Chinese word embeddings with sequence sentence modeling.

	word2vec	*GloVe*	*ELMo*			*BERT*	
			1	2	3	single	pairs
LSTM	67.68	70.47	70.31	71.59	67.41	67.69	66.57
BiLSTM	70.19	71.30	71.03	72.70	70.47	66.30	68.24
CNN	70.75	70.20	69.92	71.59	72.42	**74.09**	70.47
Child-Sum-Tree-LSTM	-	-	68.24	70.75	70.75	72.98	-
Binary-Tree-LSTM	-	-	70.19	70.19	72.14	73.53	-

Table 7. Comparisons of F_1 scores (%) for English sequence and structural sentence modeling.

	GloVe	*ELMo*	*BERT*
LSTM	43.08	45.62	45.24
BiLSTM	43.54	48.09	45.17
Child-Sum-Tree-LSTM	41.14	46.45	47.01
Binary-Tree-LSTM	44.10	48.53	**50.44**

Table 8. Comparisons of F_1 scores (%) for English sentence interaction layer.

			dropout				
			0.5	0.4	0.3	0.2	0.1
BiLSTM	BERT(Large-single)	without Attention	46.32	45.54	45.69	46.76	44.80
		Attention	47.24	51.93	50.75	48.98	48.28
		Con-self-Attention	21.22	41.10	44.05	47.33	49.77
		Self-Attention-con	38.28	42.52	46.93	49.12	50.06
		Attention-mlp	49.32	**53.05**	48.38	50.54	50.97
	BERT(Base-pairs)	without-Attention	49.5	50.37	48.54	47.57	50.64
		Self-Attention	27.82	36.84	49.58	**52.33**	50.95
		Attention-mlp	28.42	34.90	44.49	45.14	49.64
	ELMo(second)	without Attention	48.09	45.45	43.48	45.12	43.46
		Attention	30.97	32.55	35.80	35.87	33.54
		Con-self-Attention	43.35	45.31	46.68	46.57	47.89
		Self-Attention-con	45.65	47.65	48.87	48.28	45.26
		Attention-mlp	**49.72**	45.63	45.58	47.25	46.92

Table 9. Comparisons of Accuracy(%) for Chinese sentence interaction layer.

	word2vec +BiLSTM	$ELMo_2$ + BiLSTM	$BERT_{single}$ + CNN
without Attention	70.19	72.70	**74.09**
Attention	68.80	68.52	70.75
Con-self-Attention	70.75	65.74	71.30
Self-Attention-con	72.98	70.20	70.75
Attention-mlp	70.47	64.90	69.63

training data as shown in Table 8. Note that although the BERT-pairs model the two arguments simultaneously and get some interaction information at the level of word representation, we still add the interaction layer to this model to make comparison. The results show that the sentence interaction level is helpful to identify discourse relations.

Table 9 lists the results of interaction in Chinese. We find that the interaction layer does not help even after the dropout adjusting. This may be caused by the language characteristics. The interaction layer aims to amplify the corresponding

Table 10. Comparisons of our model with recent systems for English implicit dataset.

	$P(\%)$	$R(\%)$	$F_1(\%)$
Wang and Lan (2016) [23]	46.51	46.33	46.42
Xu et al. (2018)[27]	60.63	-	44.48
Dai et al. (2018)[4]	59.75	-	51.84
Ours(BERT+BiLSTM+Attention-mlp)	58.22	51.14	**53.05**

Table 11. Comparisons of our model with recent systems for Chinese non-Explicit dataset, accuracy(%).

	Development Set	Test Set
Wang and Lan (2016) [23]	**73.53**	72.42
Rutherford and Xue (2016) [20]	71.57	67.41
Schenk et al. (2016)[21]	70.59	71.87
Rönnqvist et al. (2017)[17]	-	73.01
Ours(BERT-single+CNN)	72.54	**74.09**

parts of the two arguments and pay less attention on the noise of the sentences to promote the classification performance. But most Chinese sentences are short due to the omission of sentence elements. Thus the attention operation may not effectively capture the interaction information between two arguments, leading to bad performance in Chinese.

Finally, Table 10 and Table 11 show the comparison of our best model with recent systems for multi-class classification for English and Chinese result. Our models achieve state-of-the-art performance on English and Chinese datasets.

4 Conclusion

In this paper, we study the influence of each component on the recognition of English and Chinese implicit discourse relation with deep learning method. The contextual word embeddings outperform context-free embeddings for both English and Chinese. But structural sentence modeling and attention interaction have positive impact on English data rather than on Chinese data. Due to different linguistic features, the neural components have different effects on implicit discourse relationship recognition. Besides, our models achieve state-of-the-art performance on English and Chinese discourse benchmark corpora.

References

1. Biran, O., McKeown, K.: Aggregated word pair features for implicit discourse relation disambiguation (2013)

2. Braud, C., Denis, P.: Combining natural and artificial examples to improve implicit discourse relation identification. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 1694–1705 (2014)
3. Chen, J., Zhang, Q., Liu, P., Qiu, X., Huang, X.: 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). vol. 1, pp. 1726–1735 (2016)
4. Dai, Z., Huang, R.: 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). pp. 141–151 (2018)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186 (2019)
6. Ji, Y., Zhang, G., Eisenstein, J.: Closing the gap: Domain adaptation from explicit to implicit discourse relations. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2219–2224 (2015)
7. Li, J.J., Carpuat, M., Nenkova, A.: Assessing the discourse factors that influence the quality of machine translation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). vol. 2, pp. 283–288 (2014)
8. Liu, Y., Li, S., Zhang, X., Sui, Z.: Implicit discourse relation classification via multi-task neural networks. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
9. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. pp. 55–60 (2014)
10. McClosky, D., Charniak, E., Johnson., M.: Bllip north american news text, complete. In: Linguistic Data Consortium. p. 4: 3 (2008)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, S., G.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems (2013)
12. Pennington, Jeffrey, Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (2014)
13. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of NAACL (2018)
14. Pitler, E., Nenkova, A.: Using syntax to disambiguate explicit discourse connectives in text. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. pp. 13–16. Association for Computational Linguistics (2009)
15. Qin, L., Zhang, Z., Zhao, H.: Shallow discourse parsing using convolutional neural network. Proceedings of the CoNLL-16 shared task pp. 70–77 (2016)
16. Qin, L., Zhang, Z., Zhao, H., Hu, Z., Xing, E.: 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). pp. 1006–1017 (2017)

17. Rönnqvist, S., Schenk, N., Chiarcos, C.: 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. pp. 256–262. Association for Computational Linguistics, Vancouver, Canada (2017)
18. 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. pp. 645–654 (2014)
19. Rutherford, A., Xue, N.: Improving the inference of implicit discourse relations via classifying explicit discourse connectives. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 799–808 (2015)
20. Rutherford, A., Xue, N.: Robust non-explicit neural discourse parser in english and chinese. Proceedings of the CoNLL-16 shared task pp. 55–59 (2016)
21. Schenk, N., Chiarcos, C., Donandt, K., Rönnqvist, S., Stepanov, E., Riccardi, G.: Do we really need all those rich linguistic features? a neural network-based approach to implicit sense labeling. Proceedings of the CoNLL-16 shared task pp. 41–49 (2016)
22. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 1556–1566 (2015)
23. Wang, J., Lan, M.: Two end-to-end shallow discourse parsers for english and chinese in conll-2016 shared task. In: Proceedings of the CoNLL-16 shared task. pp. 33–40 (2016)
24. Wang, X., Yoshida, Y., Hirao, T., Sudoh, K., Nagata, M.: Summarization based on task-oriented discourse parsing. IEEE Transactions on Audio, Speech, and Language Processing **23**(8), 1358–1367 (2015)
25. Wang, Y., Li, S., Yang, J., Sun, X., Wang, H.: Tag-enhanced tree-structured neural networks for implicit discourse relation classification. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 496–505 (2017)
26. Wu, C., Shi, X., Chen, Y., Su, J., Wang, B.: Improving implicit discourse relation recognition with discourse-specific word embeddings. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 269–274 (2017)
27. Xu, Y., Hong, Y., Ruan, H., Yao, J., Zhang, M., Zhou, G.: Using active learning to expand training data for implicit discourse relation recognition. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 725–731 (2018)
28. Yang, B., Cardie, C.: Context-aware learning for sentence-level sentiment analysis with posterior regularization. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 325–335 (2014)