Neurocomputing 369 (2019) 39–49

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Boosting implicit discourse relation recognition with connective-based word embeddings

Changxing Wu[a,*], Jinsong Su[b], Yidong Chen[c,d], Xiaodong Shi[c,d]

[a] Virtual Reality and Interactive Techniques Institute, East China Jiaotong University, Nanchang 330013, China
[b] Xiamen University, Xiamen 361005, China
[c] Fujian Key Lab of the Brain-like Intelligent Systems, Xiamen University, Xiamen 361005, China
[d] Department of Cognitive Science, School of Information Science and Technology, Xiamen University, Xiamen 361005, China

ARTICLE INFO

Article history:
Received 4 September 2018
Revised 12 June 2019
Accepted 27 August 2019
Available online 30 August 2019

Communicated by Dr. Y. Chang

Keywords:
Connective-based word embeddings
Implicit discourse relation recognition
Connective classification
Neural network

ABSTRACT

Implicit discourse relation recognition is the performance bottleneck of discourse structure analysis. To alleviate the shortage of training data, previous methods usually use explicit discourse data, which are naturally labeled by connectives, as additional training data. However, it is often difficult for them to integrate large amounts of explicit discourse data because of the noise problem. In this paper, we propose a simple and effective method to leverage massive explicit discourse data. Specifically, we learn connective-based word embeddings (*CBWE*) by performing connective classification on explicit discourse data. The learned *CBWE* is capable of capturing discourse relationships between words, and can be used as pre-trained word embeddings for implicit discourse relation recognition. On both the English PDTB and Chinese CDTB data sets, using *CBWE* achieves significant improvements over baselines with general word embeddings, and better performance than baselines integrating explicit discourse data. By combining *CBWE* with a strong baseline, we achieve the state-of-the-art performance.

1. Introduction

Recognizing discourse relations (e.g., *Comparison*) between two text spans is a crucial subtask of discourse structure analysis. These relations can benefit many downstream natural language processing tasks, including question answering, machine translation and so on. A discourse relation instance is usually defined as a discourse connective (e.g., *but, and*) taking two arguments (e.g., *clause, sentence*). Example (a) is an explicit discourse instance signaled by the connective *but*, while Example (b) is an implicit discourse instance with the *Comparison* relation, and the connective is absent. For explicit discourse relation recognition, using only connectives as features achieves more than 93% in accuracy [28]. However, due to the absence of connectives, implicit discourse relation recognition needs to inference discourse relations based on two arguments, and is still challenging. Earlier researchers usually develop surface features and use supervised learning method to perform the task [6,16,20,27,31]. Among these features, word pairs occurring in argument pairs are considered as important features, since they can partially catch discourse relationships between two arguments. For example, word pairs with antonymic relation, like *(crude, advance)* in Example (b), may mean a *Comparison* relation, while synonym word pairs like *(good, great)* may indicate a *Conjunction* relation. However, previous classifiers based on these features do not work well because of the data sparsity problem.

(a) *[The computers were crude by today's standards,]*$_{Arg1}$
***but**, [Apple II was a major advance from Apple I.]*$_{Arg2}$
(b) *[We have seen a big advance of the project.]*$_{Arg1}$
[The others are still very crude.]$_{Arg2}$

To address this problem, some researchers attempt to take advantage of unlabeled data, especially explicit discourse data, to enrich the training data. For example, explicit instances signaled by the connective *but* can be potentially used as additional training data for the *Comparison* relation in implicit discourse relation recognition. They remove connectives from explicit discourse instances, and automatically labeled them by mapping connectives into corresponding discourse relations (e.g., *but – Comparison*). However, according to Sporleder and Lascarides [33], directly using these data as additional training data would degrade the performance due to the following two drawbacks: (1) The meaning shift problem. Considering the explicit instance:*'I am eager to go home for the vacation. Nonetheless, I will book a flight to Beijing.'*, one would infer the *Contingency* relation rather than the *Comparison* relation if *nonetheless* is dropped. (2) The domain

* Corresponding author.
E-mail address: wuchangxing@ecjtu.edu.cn (C. Wu).

https://doi.org/10.1016/j.neucom.2019.08.081
0925-2312/

problem. There are different word distributions and different relation distributions between explicit and implicit discourse data. For example, in the PDTB data set, the four top-level discourse relations include: explicit instances (18.9% *Temporal*, 28.8% *Comparison*, 18.7% *Contingency* and 33.6% *Expansion*) and implicit instances (5.7% *Temporal*, 16.9% *Comparison*, 24.9% *Contingency* and 52.5% *Expansion*). In other words, Implicit and explicit discourse data can be considered as data from different domains. Accordingly, for implicit discourse relation recognition, explicit discourse instances can be potentially used as additional labeled data, but with some noise.

Recent researchers seek to leverage explicit discourse data via domain adaptation [5], data selection [32] or multi-task learning [12,13,19,38]. While showing better results, they all directly use explicit data to train classifiers. As a result, a small amount of explicit data is just used because of the noise problem. Intuitively, incorporating massive explicit discourse data would further improve the performance. Recently, some researchers use word embeddings instead of words as input features, and design various neural networks to capture discourse relationships between arguments [8,9,11,14,18,30,42]. While achieving promising results, they are all based on general word embeddings which ignore discourse information (e.g., *good, great,* and *bad* are often mapped into close vectors). In general, using task-specific word embeddings would further boost the performance.

Based on the above analysis, we propose to learn connective-based word embeddings (*CBWE*) from massive explicit data for implicit discourse relation recognition. Explicit data can be considered to be automatically labeled by connectives. While they cannot be directly used as training data for implicit discourse relation recognition and contain some noise, they are effective enough to provide weakly supervised signals to train the connective-based word embeddings. Our method is inspired by the observation that synonym (antonym) word pairs tend to appear around the discourse connective *and (but)*. Other connectives can also provide some discourse clues. We expect to mine these discourse clues from explicit data, and encode them into distributed representations of words. These representations can used as features for implicit discourse relation recognition and other discourse-related tasks, and boost their performance potentially. Compared with previous work, our method provides two benefits: (1) Discourse relevant word pair information is encoded into connective-based word embeddings, such information is helpful for implicit discourse relation recognition. As shown in the explicit Example (a), word pair (*crude, advance*) is related with the connective *but*. Our method can encode the semantic relation signaled by *but* into the embeddings of words *crude* and *advance*, which are obviously helpful for distinguish the *Comparison* relation of the implicit Example (b). (2) Our method is a two-stage method which first learns *CBWE* and then uses it as features. In this way, our method leverages massive explicit data indirectly and thus can reduce the influence of noise. On the other hand, both data selection and multi-task methods use explicit data to train their recognition models directly, which makes them more susceptible to noise and harder to incorporate massive explicit data.

Specifically, we use two simple and effective neural networks to learn *CBWE* by performing connective classification on massive explicit data: an average model capturing discourse relationships between words implicitly, and an interaction model capturing these relationships explicitly. We apply *CBWE* as pre-trained word embeddings for a neural implicit discourse relations recognition model. On both the English PDTB [29] and Chinese CDTB [15] data sets, using *CBWE* yields significantly better performance than using general word embeddings, and recent methods incorporating explicit discourse data. The interaction model shows better results than the average model. More importantly, the learned *CBWE* can be easily transferred to strong implicit discourse relation models like [3,9], to boost their performance further.

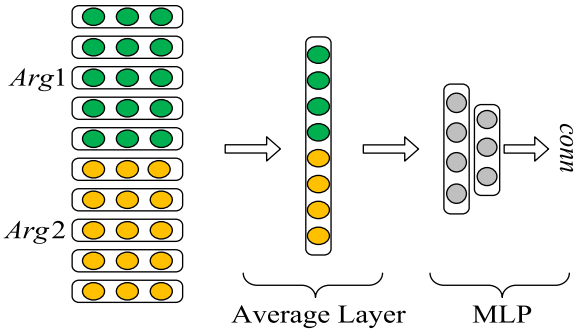

Fig. 1. The average model for learning *CBWE*.

The major contributions of this paper include: (1) We introduce a simple and effective method to leverage explicit discourse data. (2) The learned *CBWE*[1] can be easily combined with other techniques, to potentially boost the performance of implicit discourse relation recognition further. (3) We achieve the state-of-the-art performance, to the best of our knowledge. The contents of this paper are organized as follows. We detail models for learning *CBWE* in Section 2 and the used implicit discourse relation recognition model in Section 3. We conduct experiments to validate the effectiveness of *CBWE* in Section 4. Finally, we review the related work in Section 5 and draw conclusions in Section 6.

2. Connective-based word embeddings

We induce *CBWE* based on explicit discourse data by performing connective classification. The connective classification task predicts which connective is suitable for combining two given arguments. In this section, we first introduce two simple and effective neural network models for learning *CBWE*, and then the way of collecting explicit discourse data for training.

2.1. The average model

We adapt the model in [38] to learn *CBWE* by performing connective classification, and call it the average model. As illustrated in Fig. 1, it uses an average layer to represent two arguments, and then a multi-layer perception (MLP) for classification. The average model is simple enough to enable us to train on massive data efficiently.

Formally, let (Arg_1, Arg_2, *conn*) denotes an explicit discourse instance, where Arg_1 and Arg_2 are arguments, *conn* is the connective. Let $x_i \in R^d$ and $y_j \in R^d$ denote the embeddings of ith word of Arg_1 and jth word of Arg_2, m and n denote the lengths of Arg_1 and Arg_2, respectively. All word's embeddings can be denoted as a matrix $L \in R^{v \times d}$, where v is the size of vocabulary and d the dimension of word embeddings. The average model first represents an argument as the average of words, x for Arg_1 and y for Arg_2. And then it concatenates x and y as h_0, a single representation of both Arg_1 and Arg_2:

$$x = \frac{1}{m}\sum_{i=1}^{m} x_i, \quad y = \frac{1}{n}\sum_{j=1}^{n} y_j, \quad h_0 = [x, y] \tag{1}$$

Finally, $h_0 \in R^{2d}$ is fed into a MLP for classification. Specially, l nonlinear hidden layers are stacked to get a more abstractive representation h_l, and then a *softmax* layer to get probabilities of different

[1] Our learned *CBWE* is publicly available at here, and we will make the source code available after review.

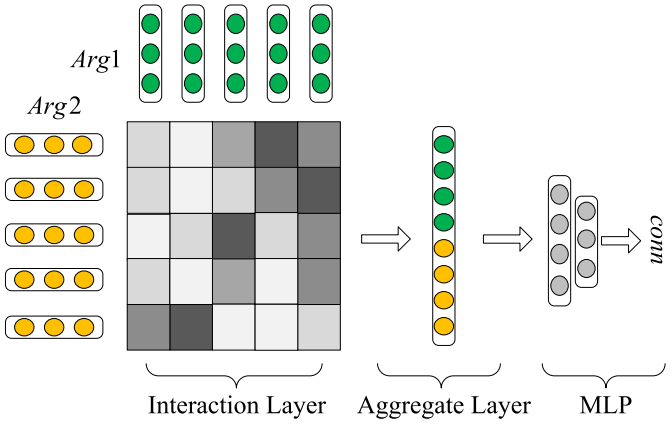

Fig. 2. The interaction model for learning *CBWE*.

classes o:

$$h_i = f(W_i h_{i-1} + b_i), i \in [1, l]$$
$$o = softmax(W_c h_l + b_c) \quad (2)$$

where f is the nonlinear activation function for all hidden layers, W_i, b_i, W_c, b_c are parameters. We combine the cross-entropy error and regularization error as the objective function:

$$J(\theta) = -\sum_{k=1}^{q} g_k \times \log(o_k) + \frac{\lambda}{2} \parallel \theta' \parallel^2, \quad (3)$$

where g is the ground-truth label vector for an training instance, q the number of classes, λ the regularization coefficient and $\theta = (L, W_i, b_i, W_c, b_c)$ the set of parameters. Note that b_i, b_c and L are not included in θ'. During training, L is first randomly initialized, and then tuned to minimize the objective function. The finally obtained L is our *CBWE*.

2.2. The interaction model

Recently, neural network models which incorporate word pair information directly achieve superior performance on nature language inference [24] and implicit discourse relation recognition [14]. Therefore, we propose to use a simplified version of these models to learn *CBWE*, and call it the interaction model. As illustrated in Fig. 2, the interaction model first uses an interaction layer to capture the cross-argument word pair information, then an aggregate layer to represent arguments, and finally a MLP layer for classification. The interaction model captures discourse relationships between words explicitly, while the average model does this implicitly.

Formally, we first calculate the word interaction score matrix $E \in R^{m \times n}$, computing e_{ij} for each possible ith word (x_i) in Arg_1 and jth word (y_j) in Arg_2, and normalize them as follows:

$$e_{ij} = x_i y_j^T, \quad \alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})}, \quad \beta_{ji} = \frac{exp(e_{ij})}{\sum_{k=1}^{m} exp(e_{kj})}. \quad (4)$$

A high interaction score e_{ij} means the corresponding word pair is well correlated with a particular discourse relation. $\alpha_{i1}, \ldots, \alpha_{in}$ indicate the normalized interaction scores between the ith word in Arg_1 and each word in Arg_2. Similarly, $\beta_{j1}, \ldots, \beta_{jm}$ indicate the normalized interaction scores between the jth word in Arg_2 and each word in Arg_1. As described in Eq. (5), based on these normalized scores, we augment the representation of ith word in Arg_1 as $[x_i, x_i']$, where x_i' can be considered as its related parts in Arg_2. Similarly, the jth word in Arg_2 is represented as $[y_j, y_j']$. Finally, an argument is represented as the average of augmented representations of words in it, x for Arg_1 and y for Arg_2. And the concatenation $[x, y]$ is fed into a multi-layer perception for classification.

$$x_i' = \sum_{j=1}^{n} \alpha_{ij} y_j, \quad x = \frac{1}{m} \sum_{i=1}^{m} [x_i, x_i'],$$
$$y_j' = \sum_{i=1}^{m} \beta_{ji} x_i, \quad y = \frac{1}{n} \sum_{j=1}^{n} [y_j, y_j']. \quad (5)$$

Essentially, the task of connective classification is similar to implicit discourse relation recognition, just with different output labels. Therefore, any effective neural network model for implicit relation recognition can be easily adapted for connective classification. The reasons why we choose simple models instead of complicated models for learning *CBWE* are two-folds: (1) training simple models on massive explicit data is time-efficient, and (2) simple models usually contain fewer other parameters, which makes as much as possible information be encoded into the word embeddings. For example, both the average model and the interaction model only have two sets of parameters: word embeddings and parameters of the MLP. In our experiments, if we compute $e_{ij} = F(x_i)F(y_j)^T$ as that in [24], where F is a feed-forward neural network, or $e_{ij} = x_i A y_j^T + B[x_i, y_j] + c_{ij}$ as that in [14], where A, B, c_{ij} are parameters to model and encode word pair semantics, the resulted *CBWE* is not as good as that learned by the average model or the interaction model. The reason behind is that some useful information is encoded into the parameters of F or A, B, c_{ij}.

Word order information is not used in both the average model and the interaction model, which makes our models are very simple and can be trained on large amounts of explicit discourse data efficiently. Intuitively, considering the word order information can boost the performance of connective classification. For example, we can enhance the representation of a word by concatenating its word embedding and position embedding as [10], or use recurrent neural networks (e.g., LSTM) instead of the average operator in Eq. (1) or Eq. (5). We conduct experiments (results not listed) to verify the above two methods and find that: (1) Both methods do not result in better *CBWE*. Especially when LSTM is used, the resulting *CBWE* is not as good as that learned by the interaction method. The reason is that some useful information is encoded into the parameters of LSTM. (2) Transferring the learned position embeddings is not helpful. The reasoning behind is that word order information can be learned from any sentences, not just limited to the explicit discourse data. (3) Using LSTM is very time-consuming because it cannot be parallelized. Our primary purpose in this paper is to learn better *CBWE*. The learned *CBWE* can be easily transferred to not only the *IDRR* model (Section 3), but also future models for this task or discourse-related tasks, to potentially boost their performance. Based on the above analysis, we do not consider the order information in our models for *CBWE*.

Compared with the attention mechanisms in sequence-to-sequence models [2] and multi-head self-attention models [35], our interaction model is bidirectional and usually used in scenarios with two sentences. Compared with the bi-attention models used in [14,24], our interaction model is a simplified version. Specifically, we adapt commonly used bi-attention models to catch word-pair information explicitly and encode these information into word embeddings. We simplify them by retaining only two sets of parameters that are necessary, word embeddings and parameters of the MLP layer. Therefore, as much as possible information is encoded into the word embeddings as expected. Experimental results in Table 6 also show that our interaction model is more suitable for learning *CBWE* than the commonly used bi-attention models. Accordingly, to some extent, our interaction model is proposed for a new application, learning connective-based word embeddings (*CBWE*) from massive explicit discourse data.

2.3. Collecting explicit discourse data

Collecting explicit discourse data includes two steps: (1) distinguish whether a connective occurring reflects a discourse relation. For example, the connective *and* can either function as a discourse connective to join two *Conjunction* arguments, or be just used to link two nouns in a phrase. (2) identify the positions of two arguments. According to Prasad et al. [29], Arg_2 is defined as the argument following a connective, however, Arg_1 can be located within the same sentence as the connective, in some previous or following sentence. Lin et al. [17] show that the accuracy of distinguishing English connectives is more than 97%, while identifying arguments is below than 80%. Therefore, we use the existing toolkit[2] to find English discourse connectives, and just collect explicit instances using patterns like [Arg_1 *conn* Arg_2], where two arguments are in the same sentence, to decrease noise.

The restriction of the same sentence seems to have two potentially detrimental effects on the collected corpus. First, it would skew the distribution of discourse relations towards those that are typically expressed within the same sentence. Second, it would actually exclude explicit discourse instances consisting of two separate sentences, which are more similar to implicit discourse instances. To explore this problem, we compare two collected English explicit discourse corpora: (1) one collected by our method, and (2) the other collected from the results of the pdtb-style parser [17], where all identified explicit instances are included. We find that: (1) there is no obvious difference in the distribution of connectives between the two corpora, and (2) connective based word embeddings trained on two corpora achieve similar performance on implicit discourse relation recognition. Therefore, our way of collecting explicit data is feasible when using a very large corpus. It can also be easily generalized to other languages, one just need to train a classifier to find discourse connectives following [17].

3. Model for implicit discourse relation recognition

To evaluate the effectiveness of our learned *CBWE*, we use it as the pre-trained word embeddings for a popular implicit discourse relation recognition model (*IDRR* model, hereafter), to see if it improves the performance. In fact, our *CBWE* can be easily used for any neural implicit discourse relation recognition model. In the following, we briefly introduce the used *IDRR* model[3] [24] for this paper to be self contained. Let us recall that the interaction model for *CBWE* (Section 2.2) is essentially a simplified version of the *IDRR* model. They both use an interaction layer to capture the cross-argument word pair information, then an aggregate layer to represent arguments, and finally a MLP layer for classification. The only differences between them are the *IDRR* model uses Eq. (6) instead of Eq. (4) for word interaction scores, and Eq. (7) instead of Eq. (5) for argument representations. Specifically, F in Eq. (6) and G in Eq. (7) are multi-layer feed-forward neural networks and added for learning more abstract feature representations.

$$e_{ij} = F(x_i)F(y_j)^T, \quad \alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})}, \quad \beta_{ji} = \frac{exp(e_{ij})}{\sum_{k=1}^{m} exp(e_{kj})} \tag{6}$$

$$\begin{aligned} x'_i &= \sum_{j=1}^{n} \alpha_{ij} y_j, \quad x = \frac{1}{m} \sum_{i=1}^{m} G([x_i, x'_i]) \\ y'_j &= \sum_{i=1}^{m} \beta_{ji} x_i, \quad y = \frac{1}{n} \sum_{j=1}^{n} G([y_j, y'_j]) \end{aligned} \tag{7}$$

[2] https://github.com/linziheng/pdtb-parser.

[3] Though the model described in [24] is used for nature language inference, we find it is also effective for implicit discourse relation recognition, and achieves slightly better results than the model in [14] in our experiments.

Table 1
Statistics of data sets on the PDTB.

Relation	Training	Validation	Test
Temp	582	48	55
Comp	1855	189	145
Cont	3235	281	273
Expa	6673	638	538

Table 2
Top 10 most frequent connectives in the collected explicit discourse data.

Connective	Frequency	Connective	Frequency
and	1,040,207	*when*	224,116
but	770,705	*after*	209,224
also	665,039	*if*	202,497
while	238,364	*however*	155,811
as	227,702	*because*	150,589

4. Experiments

In this section, we evaluate the effectiveness of our proposed method on both the English PDTB and Chinese CDTB data sets. We focus on testing whether our method is more helpful than previous methods which use explicit discourse data as additional training data, and whether our learned *CBWE* can be combined with other techniques to boost the performance further.

4.1. Data and settings

Implicit discourse relation recognition is usually considered as a multi-way classification task. Following [19], we perform a 4-way classification on the four top-level relations in the PDTB, including *Temporal* (*Temp*), *Comparison* (*Comp*), *Contingency* (*Cont*) and *Expansion* (*Expa*). We adopt the standard settings and split the PDTB corpus into the training set (Sections 2–20), validation set (Sections 0–1) and test set (Sections 21–22). Table 1 lists the statistics of these data sets.

We collect explicit data from the *Xin* and *Ltw* parts of the English Gigaword Corpus (3rd edition), and get about 4.92M explicit instances. There are 100 discourse connectives in the PDTB, we ignore four parallel connectives (e.g., *if... then*) for simplicity. Due to the space limitation, we only list the top 10 most frequent English connectives in the collected corpus in Table 2. We randomly sample 20,000 instances as the validation set, 20,000 instances as the test set and the others as the training set for *CBWE*. After discarding words occurring less than 5 times, the size of the vocabulary is 185,048. For connective classification with the interaction model, we obtain an accuracy of about 58.3% on the test set when all 96 connectives are considered, and about 58.9%, 60.8%, 62.8%, 69.9% with the top 60, 30, 20, 10 most frequent connectives, respectively. Accuracies of the average model are about 4–5% lower than those of the interaction model. These results indicate that: (1) our simple models for connective classification are effective, and (2) the interaction model is more powerful than the average model by capturing discourse relationships between words explicitly.

Hyper-parameters for *CBWE* and *IDRR* are selected based on their corresponding validation sets, and listed in Table 3. The same hyper-parameters are used for the average model and the interaction model for *CBWE*. d means the dimension of word embeddings, *bsize* the batch size of training data, *lr* the learning rate, *dropout* the dropout rate [34] in hidden layers, λ the regularization coefficient in Eq. (3), *update* the parameter update strategy. The nonlinear function f is used in Eq. (2), hidden layers of F in Eq. (6) and G in Eq. (7). The learning rate for *CBWE* is decayed by a factor of 0.8 per epoch. In addition, *hsizesMLP*, *hsizesF* and *hsizesG* are the sizes of hidden layers in the MLP, F in Eq. (6) and G in

Table 3
Hyper-parameters for training *CBWE* and *IDRR*.

Hyper-parameter	*CBWE*	*IDRR*
d	300	300
bsize	64	32
lr	1.0	0.1
dropout	–	0.2
λ	0.0001	0.0001
update	SGD	AdaDelta
f	ReLU	ReLU
hsizesMLP	[200]	[200, 50]
hsizesF	–	[100, 100]
hsizesG	–	[100, 100]

Eq. (7), respectively. Note that [200, 50] means that two hidden layers with the sizes of 200 and 50 are used, and parameters in all hidden layers are initialized with the default xavier_initializer function in Tensorflow [1]. We also find that, AdaDelta is more stable than other parameter update strategies for *IDRR*, and a relative large learning rate can effectively speed up the training process of *CBWE*. The learned *CBWE* is used as the pre-trained word embeddings for *IDRR*, and fixed during training. Validation sets are used to early stop the training process. Different from the conference version [39] of this paper, no any surface features are used here for a fair comparison with other work.

Due to the small and uneven test data set, we use both the *Accuracy* and *Macro-averaged* F_1 (*Macro-*F_1) to evaluate the whole system. We run our method 10 times with different random seeds (therefore different initial parameters), and report the results (of a run) which are closest to the average results.

4.2. Comparison with general word embeddings

We first compare the learned connective-based word embeddings (*CBWE*) with two publicly available word embeddings[4]:

- *GloVe*[5]: trained on 840B words from internet (common crawl) using the count based model in [25], with a vocabulary of 2.2M and a dimensionality of 300.
- *word2vec*[6]: trained on 100B words from *Google News* using the CBOW model in [22], with a vocabulary of 3M and a dimensionality of 300.
- $CBWE_{avg}$: our connective-based word embeddings learned with the average model.
- $CBWE_{int}$: our connective-based word embeddings learned with the interaction model.

Results in Table 4 show that *IDRR* using *CBWE* gains significant improvements (one-tailed *t*-test with $p<0.05$) over using *GloVe* or *word2vec*, on both *Accuracy* and *Macro-*F_1. More importantly, using *CBWE* achieves substantial improvements across all relations on the F_1 score, which indicates that our proposed method can not only help minority relations (*Temp, Comp*), but also major relations (*Cont, Expa*). In addition, *IDRR* using $CBWE_{int}$ achieves better performance over using $CBWE_{avg}$, which suggests that modeling interaction between words explicitly is really helpful. Overall, our *CBWE* can effectively incorporate discourse information in explicit discourse data, and thus benefits implicit discourse relation recognition.

[4] The reasons for using these word embeddings are: (1) They are both trained on massive data. (2) It will be convenient for other people to reproduce our experiments. (3) Using *GloVe* or *word2vec* word embeddings trained on the same corpus as *CBWE* achieves worse performance than the public embeddings.

[5] http://nlp.stanford.edu/data/glove.840B.300d.zip.

[6] https://code.google.com/archive/p/word2vec/GoogleNews-vectors-negative300.bin.gz.

Table 4
Results of using different word embeddings. We also list the Precision (*P*), Recall (*R*) and F_1 score for each relation.

IDRR		*+GloVe*	*+word2vec*	*+*$CBWE_{avg}$	*+*$CBWE_{int}$
Temp	*P*	42.11	51.72	50.00	44.90
	R	29.09	27.27	34.55	40.00
	F_1	34.41	35.71	40.86	42.31
Comp	*P*	39.29	40.54	40.00	47.44
	R	22.76	20.69	24.83	25.52
	F_1	28.82	27.40	30.64	33.18
Cont	*P*	49.00	46.69	49.64	49.26
	R	45.05	49.08	49.82	48.72
	F_1	46.95	47.86	49.73	48.99
Expa	*P*	62.23	62.48	64.70	64.82
	R	73.79	72.12	73.23	73.98
	F_1	67.52	66.95	68.70	69.10
Accuracy		56.28	56.08	57.86	**58.36**
Macro-F_1		44.42	44.48	47.48	**48.39**

In the first line of Table 4, there is a drop from 50.00% to 44.90% on Precision of *Temp* (*+*$CBWE_{avg}$ vs. *+*$CBWE_{int}$). The possible reason is that the number of test instances (*Temp*) is very small, only 55 in the test set (as listed in Table 1). In this case, the Precision and Recall scores on *Temp* (class-wise) are relatively sensitive. Note that, the same phenomenon can also be found in [19], where the Precision on *Temp* drops from 60.00% to 42.42% when more auxiliary tasks are used.[7] It is worthy to note that, for the other three relations, the performance is more stable. In addition, the test set is very uneven, for example, the *Expa* instances account for 53.2% of total instances. Therefore, like most previous work, both the *Accuracy* and *Macro-*F_1 on the whole test set are used to evaluate our method.

4.3. Comparison with recent methods

In this section, we compare our method with recent methods which also use explicit discourse data to boost the performance:

- [32]: a data selection method which directly enlarges the training data with the chosen explicit discourse data.
- [7]: a count-based method to learn connective-based word representations from explicit discourse data, which are then used as features in a logistic regression model.
- [19]: a multi-task neural network model to incorporate several discourse-related data, including explicit discourse data and the RST-DT corpus [37].
- [38]: a bilingually-constrained method to synthesize additional training data and a multi-task neural network to incorporate these synthetic data.
- [12]: an attention-based mechanism to learn representations through interaction between arguments, and a multi-task neural network to leverage knowledge from explicit discourse data.
- [9]: a paragraph-level neural network to model interdependencies between discourse units, and a CRF layer to predict a sequence of explicit and implicit relations in a paragraph. Both the labeled implicit and explicit instances in the PDTB are used.

Results in Table 5 show the superiority of our proposed method, with the highest *Accuracy* and a comparable *Macro-*F_1 among these methods. The main reason for these improvements is that our method can effectively utilize massive explicit discourse data, up to about 4.88M instances. Both the data selection method [32] and multi-task methods [12,19,38] directly use explicit data to estimate parameters of implicit discourse relation classifiers. As a result, it

[7] Please refer to Table 7 in [19] for more details.

Table 5
Comparison with recent methods.

Method	*Accuracy*	*Macro-F_1*
[32]	57.10	40.50
[7]	52.81	42.27
[19]	57.27	44.98
[38]	58.06	45.19
[12]	57.39	47.80
[9]	57.44	**48.82**
IDRR+CBWE$_{avg}$	57.86	47.48
IDRR+CBWE$_{int}$	**58.36**	48.39

Table 6
The transfer of *CBWE*. *CBWE$_{IDRR}$* means learning the *CBWE* with the *IDRR* model. * means that we run their code and report results. *+CBWE$_*$* means using the learned *CBWE$_*$* instead of general word embeddings.

Method	*Accuracy*	*Macro-F_1*
IDRR+CBWE$_{int}$	58.36	48.39
IDRR+CBWE$_{IDRR}$	57.17	46.63
IDRR+CBWE$_{IDRR}$+Feature layer transfer	58.46	48.00
[9]	57.44	48.82
[3]*	60.14	50.69
[9]*+CBWE$_{int}$*	58.85	49.21
[3]*+CBWE$_{int}$*	**60.93**	**51.32**

is hard for them to incorporate massive explicit data because of the noise problem. For example, only 20,000 and 40,000 explicit discourse instances are used in [32] and [19], respectively. While Braud and Denis [7] uses massive explicit discourse data, it is limited by the fact that the maximum dimension of word representations is restricted by the number of connectives, for example 96 in their work. By comparison, we learn *CBWE* by predicting connectives conditioning on arguments, which has no such dimension limitation and yields better performance. Overall, our method can conveniently and effectively leverage massive explicit discourse data, and thus is more powerful than recent baselines.

Dai and Huang[9] performs slightly better on *Macro-F_1*. In addition to using explicit discourse data, it also boosts the performance by using both argument-level and paragraph-level context to encode argument, casting the relation recognition task as a sequence labeling task, and augmenting input word representations with Part-of-Speech tags and named entity tags. In comparison, our *IDRR* model is relatively weak. More importantly, as the next section shows, our learned *CBWE* can be easily used to boost the performance of Dai and Huang [9].

4.4. Transfer of CBWE

From the perspective of transfer learning [23], our method only transfers word embeddings between two related tasks. Let us recall that the task of connective classification (for learning *CBWE*) is similar to implicit discourse relation recognition, just with different output labels. If two similar models (just with different MLP layers) are separately used for the two tasks, we can transfer not only word embeddings but also parameters of feature layers. We conduct some experiments to explore this problem. Specifically, we construct two models by using different MLP layers in the *IDRR* model (Section 3). One model for learning *CBWE*, the other for relation recognition. The learned *CBWE* is referred as *CBWE$_{IDRR}$*. In the upper part of Table 6, *IDRR+CBWE$_{IDRR}$* means transferring only the *CBWE$_{IDRR}$* for relation recognition, *IDRR+CBWE$_{IDRR}$+Feature layer transfer* means transferring both the *CBWE$_{IDRR}$* and parameters in feature layers (the interaction and aggregate layers). From these results, we can find that: (1) *IDRR+CBWE$_{IDRR}$* gets a significantly lower performance than our *IDRR+CBWE$_{int}$* (Line 2 vs. Line 1), and (2) *IDRR+CBWE$_{IDRR}$+Feature layer transfer* achieves comparable performance with ours (Line 3 vs. Line 1). These results indicate that the learned *CBWE$_{IDRR}$* is not as good as our *CBWE$_{int}$* and some useful information is encoded into the parameters of feature layers. In addition, it is hard to transfer parameters of feature layers to other neural network models. Therefore, the simple interaction model is more suitable for learning *CBWE* than relatively complicated models, in terms of efficiency and effectiveness. In this case, as much as possible information is encoded into *CBWE*, which can also be easily transferred to other implicit discourse relation recognition models.

In the bottom part of Table 6, using our learned *CBWE$_{int}$* instead of the pre-trained *word2vec* embeddings, both Dai and Huang [9]*+CBWE$_{int}$* and Bai and Zhao [3]*+CBWE$_{int}$* achieve better performance (Line 6 vs. Line 4 and Line 7 vs. Line 5). In addition, Bai and Zhao [3]*+CBWE$_{int}$* obtains the state-of-the-art performance, to the best of our knowledge. Note that [9] is a strong baseline by modeling both the argument-level and paragraph-level context,[8] and Bai and Zhao [3] achieves the SOTA performance via a deeper neural model augmented by different grained text representations.[9] We use the source codes provided by the authors. These results indicate that our *CBWE* can be easily combined with other advance techniques to boost the performance further, for example, the powerful contextualized word embedding ELMo [26] used in [3]. Overall, we recommend to use *CBWE* instead of general *word2vec* or *GloVe* word embeddings for implicit discourse relation recognition.

4.5. Effect of noise

The main advantage of our method is that it can leverage massive explicit discourse data, while previous methods are usually troubled by noise. In this section, we conduct experiments to show to what extent the noise in explicit data affects these methods. Specifically, we compare our method with the following two methods:

- *IDRR+word2vec+Direct*: directly extending the training data with explicit discourse data. We first map connectives to corresponding discourse relations. In order to alleviate the noise problem, we discard explicit instances with ambiguous connectives (eg. *while*), and randomly sample a subset of explicit data, with the same distribution of implicit data.
- *IDRR+word2vec+MT*: leveraging explicit discourse data in a multi-task framework. Following Liu et al. [19], a connective classification task is defined on explicit discourse data, and used as the auxiliary task to boost the relation recognition task. The two tasks share the same input and feature layers, and use separate MLP layers for classification. A relatively small learning rate is used for connective classification when training the two task simultaneously, to conflict with noise.

As illustrated in Fig. 3, we conduct experiments with 10, 100, 500, 1000 and 4880 thousands explicit instances, respectively. Note that 10 thousands instances are about the same amount of labeled implicit data, 100 thousands instances are usually used in previous multi-task methods, and the others can be considered as massive data. We can find that: (1) Directly using explicit data as additional training data is harmful, with significant drops in both *Accuracy* and *Macro-F_1*. The more explicit instances are used, the more the performance is affected by noise. The observation is consistent with the finding in [33]. (2) The multi-task method achieves improvements when 10 or 100 thousands explicit instances are used, but degrades the performance when more explicit instances

[8] https://github.com/ZeyuDai/paragraph-level_implicit_discourse_relation_classification.

[9] https://github.com/hxbai/Deep_Enhanced_Repr_for_IDRR.

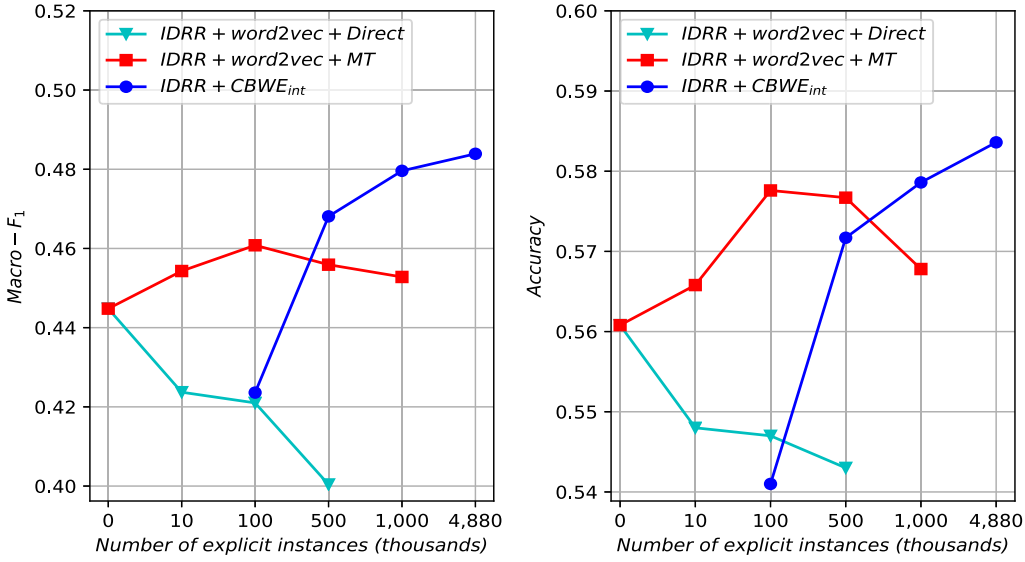

Fig. 3. The effect of noise. *Direct* means using explicit discourse data directly. *MT* means incorporating explicit discourse data in multi-task learning. *CBWE$_{int}$* means using our learned word embeddings instead of *word2vec*.

Table 7
Top 15 closest words of *not* and *good* in both *word2vec* and *CBWE*.

not			*good*		
word2vec	*CBWE$_{avg}$*	*CBWE$_{int}$*	*word2vec*	*CBWE$_{avg}$*	*CBWE$_{int}$*
do	no	n't	great	great	happy
did	n't	no	bad	lot	interesting
anymore	never	neither	terrific	very	positive
necessarily	nothing	nothing	decent	better	pleased
anything	neither	never	nice	success	great
anyway	none	none	excellent	well	helpful
does	difficult	nowhere	fantastic	happy	definitely
never	nor	unaware	better	certainly	glad
want	refused	unable	solid	respect	deserve
neither	impossible	nobody	lousy	fine	deserves
if	limited	unknown	wonderful	import	better
know	declined	refused	terrible	positive	fine
anybody	nobody	seldom	Good	help	lot
yet	little	hardly	tough	useful	reasonable
either	denied	impossible	best	welcome	ok

are used. In some extent, the multi-task method also uses explicit instances directly, because it updates parameters of the *IDRR* model according to the loss on explicit data. (3) Our proposed method gets better performance when massive explicit data are used, and is almost not affected by noise. The reason behind is that our method uses these additional data indirectly, learning *CBWE* first and then using it as the input for implicit discourse relation recognition. These results show that, when large amounts of discourse data are used, our methods can effectively control the noise problems.

4.6. Quality of CBWE

To give an intuition of what information is encoded into the learned *CBWE*, we list in Table 7 the top 15 closest words of *not* and *good*, according to the cosine similarity. We can find that, in *CBWE*, words similar to *not* to some extent have negative meanings. And since *refused, declined* are similar to *not*, a classifier may easily identify implicit instance *[A network spokesman would* ***not****comment. ABC Sports officials* ***declined****to be interviewed.]* as the *Expansion.Conjunction* relation. For *good* in *CBWE*, the similar words no longer include words like *bad* and *terrific*. Furthermore, the similar score between *good* and *great* in *CBWE$_{int}$* is 0.48 while the score between *good* and *bad* is just 0.30, which may make a classifier easier to distinguish word pairs *(good, great)* from *(good, bad)*, and thus is helpful for predicting the *Expansion.Conjunction* relation. This qualitative analysis demonstrates the ability of our *CBWE* to capture discourse relationships between words.

4.7. Case study

Two examples shown in Figs. 4 and 5 give us some evidence that the learned *CBWE* is superior than the *word2vec* word embeddings when used for implicit discourse relation recognition. These figures show the attention scores (e_{ij} in Eq. (6)) calculated by the *IDRR* model. Word pairs assigned with high attention scores are highlighted, the higher the score and the darker the color. From these figures, we can take a deep look into which word pairs are important when making prediction. Specifically, we show the interaction matrices of the *IDRR+word2vec* model and the *IDRR+CBWE$_{int}$* model on two test instances, to demonstrate how they behave differently.

We can find that: (1) For the *Expansion* instance in Fig. 4, the *IDRR+CBWE$_{int}$* model succeeds in detecting cross-argument word pairs that indicate the corresponding relations, e.g., *injuries-collapsed*. While the *IDRR+word2vec* model focuses on word pairs like *injuries-lines*. (2) For the *Comparison* instance in Fig. 5, the *IDRR+word2vec* model gives the wrong prediction *Expansion*. The reason is that it focuses more attention on word pairs like *options-stock*. On the other hand, the *IDRR+CBWE$_{int}$* model makes the correct prediction by giving more attention on word pairs *stopped-remained, stopped-open*. (3) After examining all test instances, we notice that *IDRR+CBWE$_{int}$* usually focuses on less words than *IDRR+word2vec*, and general words like *and, were, in* are given little attention. All these suggest that, the *CBWE$_{int}$* can catch different information from those in the *word2vec*. It catches word pair information (from explicit discourse data) that is relevant to the discourse relation recognition task. With the help of our leaned *CBWE$_{int}$*, the *IDRR* model can really focus on relation-relevant word pairs, and thus boost the recognition performance.

4.8. Number of connectives

We conduct experiments to investigate the impact of connectives used in training *CBWE* on the performance of *IDRR*.

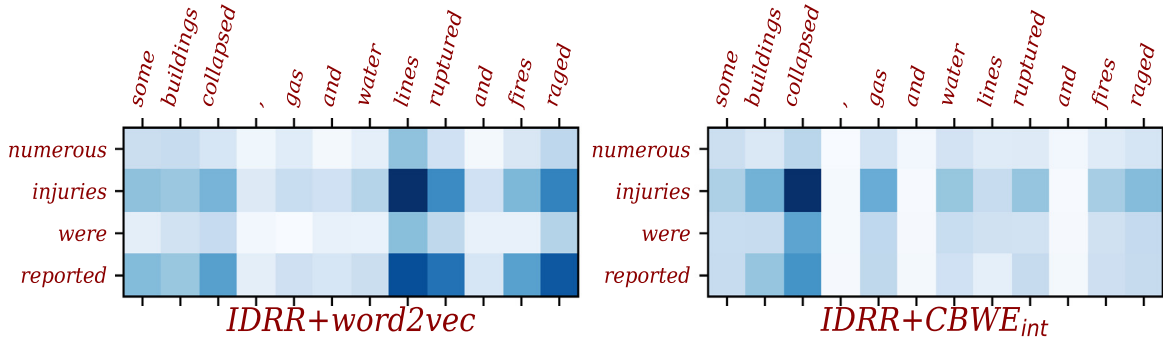

Fig. 4. An *Expansion* instance in the test set.

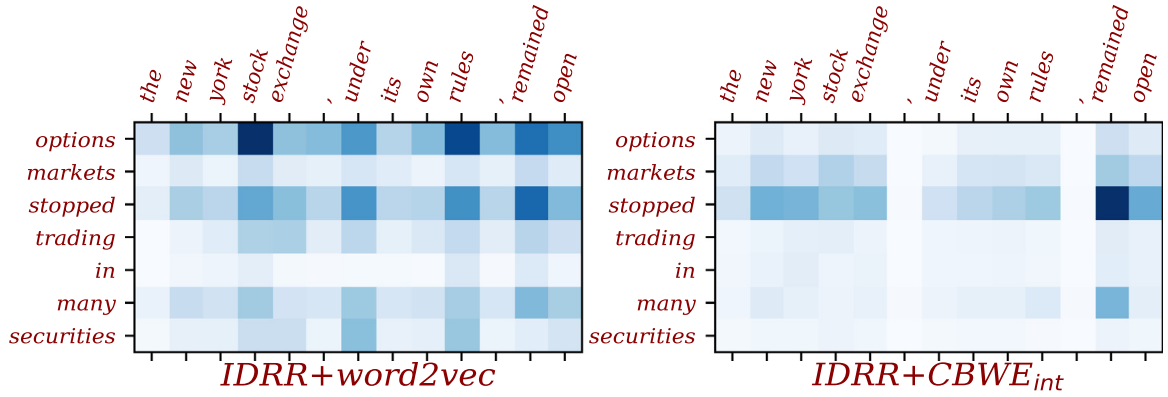

Fig. 5. A *Comparison* instance in the test set.

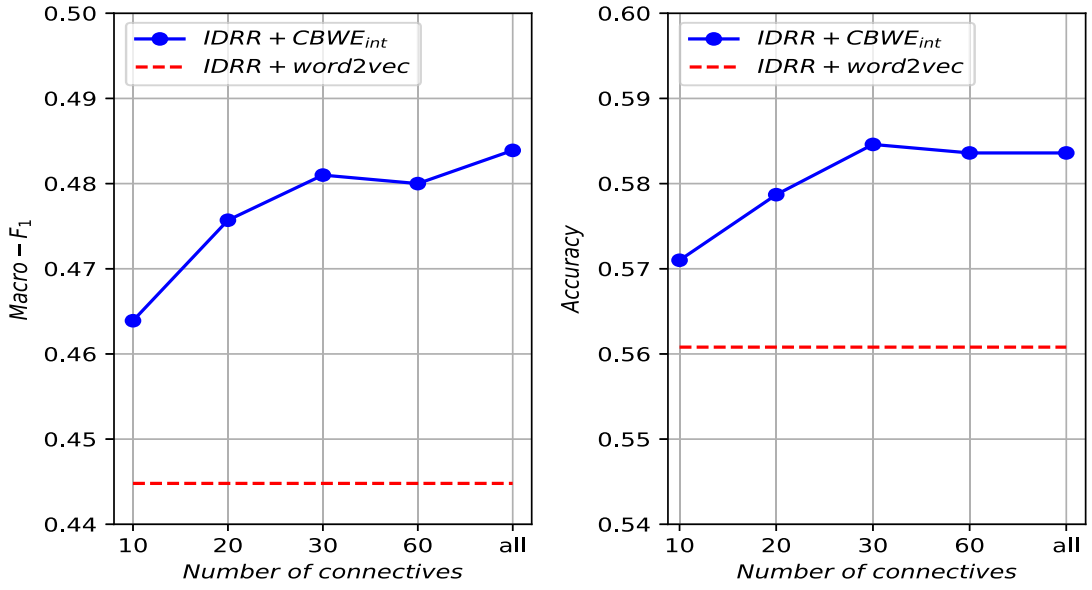

Fig. 6. Impact of connectives used in training $CBWE_{int}$.

Specifically, we use explicit discourse instances with top 10, 20, 30, 60 most frequent or all connectives to learn $CBWE_{int}$, accounting for 78.9%, 91.9%, 95.8%, 99.4% or 100% of total instances, respectively. The top 10 most frequent connectives are: *and, but, also, while, as, when, after, if, however* and *because*. According to connectives and their related relations in the PDTB, in most cases, *and* and *also* indicate the *Expansion* relation, *if* and *because* the *Contingency* relation, *after* the *Temporal* relation, *but* and *however* the *Comparison* relation. Connectives *as, when* and *while* are ambiguous. Overall, these connectives have covered all four top-level relations defined in the PDTB. As illustrated in Fig. 6, with only the top 10 connectives, the learned $CBWE_{int}$ achieves better performance than the *word2vec* word embeddings (red dotted lines). We observe significant improvements when using top 20 connectives, almost the best performance with top 30 connectives, and no further substantial improvements with more connectives. These results indicate

Table 8
Statistics of data sets on the CDTB.

Relation	Training	Validation	Test
Tran	33	1	5
Caus	682	88	95
Expl	1143	147	126
Coor	2300	529	347

Table 9
Results on the CDTB. * means that we run their codes on the CDTB. +*CBWE** means using the learned *CBWE** instead of general word embeddings.

Method	*Accuracy*	*Macro-F_1*
IDRR+GloVe	70.30	58.04
IDRR+word2vec	69.44	57.12
IDRR+CBWE$_{avg}$	73.42	63.16
IDRR+CBWE$_{int}$	73.59	64.56
[38]	74.30	62.57
[9]*	73.77	64.24
[3]*	74.82	65.95
[9]+*CBWE$_{int}$*	74.12	64.75
[3]+*CBWE$_{int}$*	**75.70**	**66.27**

that we can use only top n most frequent connectives to collect explicit discourse data for *CBWE*, which is very convenient for most languages.

4.9. Results on the CDTB

Four top-level relations are defined in the CDTB (an Chinese discourse corpus), including *Transition* (*Tran*), *Causality* (*Caus*), *Explanation* (*Expl*) and *Coordination* (*Coor*). We use instances in the first 50 documents as the test set, second 50 documents as the validation set and remaining 400 documents as the training set. Table 8 lists the statistics of these data sets. We conduct a 3-way classification because of only 39 instances for *Tran*. About 6.5M explicit discourse instances collected from the Chinese Gigaword Corpus (3rd edition), with 88 connectives, are used for *CBWE*. Note that connectives occurring less than 10,000 times are discarded. We find that hyper-parameters selected on the PDTB (see Table 3) also work well on the CDTB, except that the learning rate is set to 0.08 and batch size to 16 for training *IDRR*. For the mode in [3], we use the pre-trained Chinese ELMo embeddings[10] and ignore the sub-word information in the input layer.

Results in Table 9 show that the performance of our method on the CDTB has the similar trend as that on the PDTB. Specifically, *IDRR+CBWE* achieves significant improvements over *IDRR+GloVe* or *IDRR+word2vec* (the upper part of Table 9), and using *CBWE$_{int}$* for strong baselines achieves substantial improvements (the bottom part of Table 9). Bai and Zhao [3]+*CBWE$_{int}$* obtains the state-of-the-art performance on the CDTB, to the best of our knowledge. These results indicate that our proposed method is also effective on the Chinese implicit discourse relation recognition, and the learned *CBWE* can be easily combined with other techniques to boost the performance further.

5. Related work

Implicit discourse relation recognition attracts more attention since the release of PDTB [29], the first large discourse corpus distinguishing implicit instances from explicit ones. Most previous research focuses on designing surface features manually, including lexical and polarity features [27], word pairs and parse information [16], entity features [20], word cluster pairs [31], and so on. Recently, researchers resort to neural networks to learn distributed features automatically, for example, a shallow convolutional network [42], entity-augmented recursive networks [11], a convolutional neural network with dynamic pooling [19], gated relevance networks [8], repeated reading neural networks with multi-level attention [18], a simple word interaction model [14], and a deeper enhanced model with different grained text representations [3]. However, due to the limited training data, methods based on surface features (high dimensions) or distributed features (complicated models with many parameters) usually face the data sparsity problem.

Therefore, the second line of research tries to take advantage of unlabeled data, especially explicit discourse data (weakly labeled by connectives), to enrich the training data. For the first time, Marcu and Echihabi [21] propose to use explicit discourse instances as additional training data by removing connectives and mapping them to corresponding relations. However, Sporleder and Lascarides [33] suggest that using these artificial implicit data indiscriminately degrades the performance, because of the domain problem and meaning shift problem. Subsequently, to effectively use explicit data, some researchers use multi-task learning methods [12,13,19,38]. Specifically, they leverage auxiliary tasks (e.g., connective classification on explicit data) to promote the performance of main task (implicit discourse relation recognition), by sharing common information between them. Some researchers use data selection methods [32,36,40,41]. They select explicit instances (similar to implicit ones) according to some criteria, and use them to enlarge the training corpus directly. Both the multi-task learning and date selection methods show promising results. However, they use explicit data to train classifiers directly, which makes them hard to incorporate massive explicit data because of the noise problem. Different from the above work, we learn connective-based word embeddings from explicit data, and use them as inputting features. Our method leverages massive explicit data indirectly, and thus can reduce the influence of noise.

Some aspects of this work are similar to [4,7]. Based on massive explicit instances, they first build a word-connective (or word pair-connective) co-occurrence frequency matrix, and then weight these raw frequencies as word (word pair) representations. In this way, they represent words (word pairs) in the space of connectives to directly encode their discourse function. The major limitation of their approach is that the dimension of word representations must be less than or equal to the number of connectives. By comparison, we learn word embeddings by predicting connectives conditioning on arguments, which has no such dimension limitation. Essentially, they use count-based methods to learn word representations, while we adopt a prediction-based method and achieve better performance.

6. Conclusion

In this paper, we propose to learn connective-based word embeddings from massive explicit data for implicit discourse relation recognition. Experiments on both the PDTB and CDTB data sets show that using our learned word embeddings as features can significantly boost the performance. We also show that our method can use massive explicit data more effectively than previous work. Since most of neural network models for implicit discourse relation recognition and discourse-related tasks use pre-trained word embeddings as inputs, we hope our learned word embeddings would benefit them.

In the future, we would like to explore how to learn task-specific sentence representations based on abundance of weakly-labeled explicit discourse data. We are also interested in verifying

[10] https://github.com/HIT-SCIR/ELMoForManyLangs.

the effectiveness of our resulting word embeddings on tasks like sentiment classification, since they seem useful even beyond discourse related tasks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank all the reviewers for their constructive and helpful suggestions on this paper. This work is supported by the Natural Science Foundation of China (Grant No. 61866012), the Natural Science Foundation of Jiangxi Province (Grant No. 20181BAB202012), and Science and Technology Research Project of Education Department of Jiangxi Province (Grant No. GJJ180329).

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, TensorFlow: a system for large-scale machine learning, in: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI'16, Berkeley, CA, USA, 2016, pp. 265–283.

[2] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Proceeding of International Conference on Learning Representations, ICLR 2015, 2015, pp. 1–11.

[3] H. Bai, H. Zhao, Deep enhanced representation for implicit discourse relation recognition, in: Proceedings of International Conference on Computational Linguistics, COLING, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 571–583.

[4] O. Biran, K. McKeown, Aggregated word pair features for implicit discourse relation disambiguation, in: Proceedings of Association for Computational Linguistics, ACL, Sofia, Bulgaria, 2013, pp. 69–73.

[5] C. Braud, P. Denis, Combining natural and artificial examples to improve implicit discourse relation identification, in: Proceedings of International Conference on Computational Linguistics, COLING, Dublin, Ireland, 2014, pp. 1694–1705.

[6] C. Braud, P. Denis, Comparing word representations for implicit discourse relation classification, in: Proceedings of Empirical Methods in Natural Language Processing, EMNLP, Lisbon, Portugal, 2015, pp. 2201–2211.

[7] C. Braud, P. Denis, Learning connective-based word representations for implicit discourse relation identification, in: Proceedings of Empirical Methods in Natural Language Processing, EMNLP, Austin, Texas, 2016, pp. 203–213.

[8] J. Chen, Q. Zhang, P. Liu, X. Qiu, X. Huang, Implicit discourse relation detection via a deep architecture with gated relevance network, in: Proceedings of Association for Computational Linguistics, ACL, Berlin, Germany, 2016, pp. 1726–1735.

[9] Z. Dai, R. Huang, Improving implicit discourse relation classification by modeling inter-dependencies of discourse units in a paragraph, in: Proceedings of North American Chapter of the Association for Computational Linguistics, NAACL, Melbourne, Australia, 2018, pp. 141–151.

[10] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N. Dauphin, Convolutional sequence to sequence learning, in: Proceedings of International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 2017, pp. 1243–1252. JMLR.org.

[11] Y. Ji, J. Eisenstein, One vector is not enough: entity-augmented distributed semantics for discourse relations, Trans. Assoc. Comput. Linguist. 3 (2015) 329–344.

[12] M. Lan, J. Wang, Y. Wu, Z.-Y. Niu, H. Wang, Multi-task attention-based neural networks for implicit discourse relationship representation and identification, in: Proceedings of Empirical Methods in Natural Language Processing, EMNLP, Copenhagen, Denmark, 2017, pp. 1310–1319.

[13] M. Lan, Y. Xu, Z. Niu, Leveraging synthetic discourse data via multi-task learning for implicit discourse relation recognition, in: Proceedings of Association for Computational Linguistics, ACL, Sofia, Bulgaria, 2013, pp. 476–485.

[14] W. Lei, X. Wang, M. Liu, I. Ilievski, X. He, M.Y. Kan, SWIM: a simple word interaction model for implicit discourse relation recognition, in: Proceedings of International Joint Conferences on Artifical Intelligence, IJCAI, Melbourne, Australia, 2017, pp. 4026–4032.

[15] Y. Li, W. Feng, J. Sun, F. Kong, G. Zhou, Building Chinese discourse corpus with connective-driven dependency tree structure, in: Proceedings of Empirical Methods in Natural Language Processing, EMNLP, Doha, Qatar, 2014, pp. 2105–2114.

[16] Z. Lin, M.-Y. Kan, H.T. Ng, Recognizing implicit discourse relations in the penn discourse treebank, in: Proceedings of Empirical Methods in Natural Language Processing, EMNLP, PA, USA, 2009, pp. 343–351.

[17] Z. Lin, H.T. Ng, M.-Y. Kan, A PDTB-styled end-to-end discourse parser, Nat. Lang. Eng. 20 (02) (2014) 151–184.

[18] Y. Liu, S. Li, Recognizing implicit discourse relations via repeated reading: neural networks with multi-level attention, in: Proceedings of Empirical Methods in Natural Language Processing, EMNLP, Austin, Texas, 2016, pp. 1224–1233.

[19] Y. Liu, S. Li, X. Zhang, Z. Sui, Implicit discourse relation classification via multi-task neural networks, in: Proceedings of Conference on Artificial Intelligenc, AAAI, Arizona, USA, 2016, pp. 2750–2756.

[20] A. Louis, A. Joshi, R. Prasad, A. Nenkova, Using entity features to classify implicit discourse relations, in: Proceedings of Special Interest Group on Discourse and Dialogue, SIGDIAL, PA, USA, 2010, pp. 59–62.

[21] D. Marcu, A. Echihabi, An unsupervised approach to recognizing discourse relations, in: Proceedings of Association for Computational Linguistics, ACL, PA, USA, 2002, pp. 368–375.

[22] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Proceedings of Workshop at ICLR, 2013, pp. 1–12.

[23] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.

[24] A. Parikh, O. Täckström, D. Das, J. Uszkoreit, A decomposable attention model for natural language inference, in: Proceedings of Empirical Methods in Natural Language Processing, EMNLP 2016, Association for Computational Linguistics, Austin, Texas, 2016, pp. 2249–2255.

[25] J. Pennington, R. Socher, C. Manning, Glove: global vectors for word representation, in: Proceedings of Empirical Methods in Natural Language Processing, EMNLP, Doha, Qatar, 2014, pp. 1532–1543.

[26] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of North American Chapter of the Association for Computational Linguistics, NAACL 2018, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237.

[27] E. Pitler, A. Louis, A. Nenkova, Automatic sense prediction for implicit discourse relations in text, in: Proceedings of ACL-IJCNLP, PA, USA, 2009, pp. 683–691.

[28] E. Pitler, A. Nenkova, Using syntax to disambiguate explicit discourse connectives in text, in: Proceedings of ACL-IJCNLP, PA, USA, 2009, pp. 13–16.

[29] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, B. Webber, The penn discourse TreeBank 2.0, in: Proceedings of International Conference on Language Resources and Evaluation, LREC, 24, 2008, pp. 2961–2968.

[30] L. Qin, Z. Zhang, H. Zhao, A stacking gated neural architecture for implicit discourse relation classification, in: Proceedings of Empirical Methods in Natural Language Processing, EMNLP, Austin, Texas, 2016, pp. 2263–2270.

[31] A. Rutherford, N. Xue, Discovering implicit discourse relations through brown cluster pair representation and coreference patterns, in: Proceedings of European Chapter of the Association for Computational Linguistics, EACL, Gothenburg, Sweden, 2014, pp. 645–654.

[32] A. Rutherford, N. Xue, Improving the inference of implicit discourse relations via classifying explicit discourse connectives, in: Proceedings of North American Chapter of the Association for Computational Linguistics, NAACL, Denver, Colorado, 2015, pp. 799–808.

[33] C. Sporleder, A. Lascarides, Using automatically labelled examples to classify rhetorical relations: an assessment, Nat. Lang. Eng. 14 (3) (2008) 369–416.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceeding of Advances in Neural Information Processing Systems, NIPS, 2017, pp. 6000–6010.

[36] X. Wang, S. Li, J. Li, W. Li, Implicit discourse relation recognition by selecting typical training examples., in: Proceedings of International Conference on Computational Linguistics, COLING, Mumbai, India, 2012, pp. 2757–2772.

[37] M. William, S. Thompson, Rhetorical structure theory: towards a functional theory of text organization, Text 8 (3) (1988) 243–281.

[38] C. Wu, X. Shi, Y. Chen, Y. Huang, J. Su, Leveraging bilingually-constrained synthetic data via multi-task neural networks for implicit discourse relation recognition, Neurocomputing 243 (2017) 69–79.

[39] C. Wu, X. Shi, Y. Chen, J. Su, B. Wang, Improving implicit discourse relation recognition with discourse-specific word embeddings, in: Proceedings of Association for Computational Linguistics, ACL, 2, Vancouver, Canada, 2017, pp. 269–274.

[40] C. Wu, X. Shi, J. Su, Y. Chen, Y. Huang, Co-training for implicit discourse relation recognition based on manual and distributed features, Neural Process. Lett. 46 (1) (2017) 233–250.

[41] Y. Xu, Y. Hong, H. Ruan, J. Yao, M. Zhang, G. Zhou, Using active learning to expand training data for implicit discourse relation recognition, in: Proceedings of Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 725–731.

[42] B. Zhang, J. Su, D. Xiong, Y. Lu, H. Duan, J. Yao, Shallow convolutional neural network for implicit discourse relation recognition, in: Proceedings of Empirical Methods in Natural Language Processing, EMNLP, Lisbon, Portugal, 2015, pp. 2230–2235.

Changxing Wu received his Ph.D. degree in School of Information Science and Technology, Xiamen University, Xiamen, China, in 2017. He is now working in East China Jiaotong University. His research interests include natural language processing and deep learning.

Jinsong Su received his Ph.D. degree in computer science from Chinese Academy of Sciences, Beijing, China, in 2011. He is currently an associate professor at Xiamen University. His research interests include natural language processing and deep learning.

Yidong Chen received his Ph.D. degree in mathematics from Xiamen University, Xiamen, China, in 2008. He is now an associate professor in the Cognitive Science Department of Xiamen University. His research interests include machine translation and semantic analysis.

Xiaodong Shi received his Ph.D. degree in computer software from National University of Defense Technology, Changsha, China, in 1994. He is now a professor in the Cognitive Science Department of Xiamen University. His research interests include natural language processing and artificial intelligence.