Hindawi
Journal of Control Science and Engineering
Volume 2017, Article ID 3924641, 10 pages
https://doi.org/10.1155/2017/3924641

Research Article

Providing Definitive Learning Direction for Relation Classification System

Pengda Qin, Weiran Xu, and Jun Guo

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Weiran Xu; xuweiran@bupt.edu.cn

Received 2 July 2017; Accepted 10 August 2017; Published 12 October 2017

Academic Editor: Guang Wang

Deep neural network has adequately revealed its superiority of solving various tasks in Natural Language Processing, especially for relation classification. However, unlike traditional feature-engineering methods that targetedly extract well-designed features for specific task, the diversity of input format for deep learning is limited; word sequence as input is the frequently used setting. Therefore, the input of neural network, to some extent, lacks pertinence. For relation classification task, it is not uncommon that, without specific entity pair, a sentence contains various relation types; therefore, entity pair indicates the distribution of the crucial information in input sentence for recognizing specific relation. Aiming at this characteristic, in this paper, several strategies are proposed to integrate entity pair information into the application of deep learning in relation classification task, in a way to provide definitive learning direction for neural network. Experimental results on the SemEval-2010 Task 8 dataset show that our method outperforms most of the state-of-the-art models, without external linguistic features.

1. Introduction

With the increasing amount of information being available, it becomes harder to obtain the content we want. Under this circumstance, people tend to get the answer to the question directly rather than find the answer from text by themselves. As a result, some well-designed data-driven approaches have become mainstream [1–3]. To meet this need, vast amount of unstructured text data should be transformed into structured knowledge that is more accessible for further processing. Relation classification [4] is one of the key technologies in this procedure. It has wide applications in the field of information retrieval [5], question answering [6], and knowledge base completion [7]. Given a sentence annotated with two entities, the relation classification system is to recognize the correct relation type from a predefined relation set. For example, for the annotated sentence.

"The Pulitzer Committee issues an official $[citation]_{e1}$ *explaining the* $[reasons]_{e2}$ *for the award,"* the relation type between entity *citation* and *reasons* in this context is *Message-Topic(e1, e2)*.

In order to cope with the variety of natural language, many complicated statistical and analysis methods should be considered [8, 9]. Currently, deep learning method [10] has occupied the most influential position in dealing with relation classification task. The powerful learning capability of deep neural network enables it to deeply mine the implicit and crucial information of input sentence, without the assistance of external linguistic features. Convolutional neural network (CNN) performs well in capturing vital local information, and the advantage of time efficiency is obvious [11–13]. By comparison, Recurrent Neural Network (RNN) is better at modeling sequence information and has the capability to remember long-distance context information [14, 15]. Considering that the inputs of task are text sequences and the occurrence order of entity pair is one of the key factors, we adopt RNN as the main network. Long Short-Term Memory (LSTM) [16] and Gated Recurrent Units (GRU) [17] are two effective models among current RNN variants; moreover, due to low computational cost of GRU under the same performance, we select GRU to model our sequence information.

Entity pair is the prerequisite of relation classification, which guides the computer to discover the discriminative information. Moreover, the occurrence of entity pair indicates the main difference against other NLP tasks. Based on

this property, traditional feature-based methods specifically extract the features located in the text segment between or near entity pair; kernel methods [18] are devoted to the structural similarity of the shortest dependency parts within entity pair. Obviously, the learning direction is manually set so that classifier can avoid the interfere of noise context as far as possible. However, in most cases, text sequence is directly regarded as input for deep learning method, which naturally introduces noise for relation classification. Consequently, for deep learning method, how to take full advantage of entity pair information is the key point. The prevailing solution is to integrate position feature into text sequence to highlight the words close to entity pair and achieves good effect. But, semantic knowledge involved in entity pair has not been fully utilized. It is noteworthy that, sometimes, the relation type can be determined directly from the meaning of entity pair. Observing the following instances, the expression patterns of these two entity pairs are both "*A of B*", but the relation types are different. In other words, without the semantic information of entity pair, it is impossible to recognize the correct relation class.

I was attacked by a $[flock]_{e1}$ *of* $[pigeons]_{e2}$ *today. (Member-Collection(e2, e1))*

He decided to pad the $[heel]_{e1}$ *of* $[shoes]_{e2}$ *with a shock absorbing insole and heel pad. (Component-Whole(e1, e2))*

Therefore, how to better leverage entity information to guide the learning direction of deep neural network is of practical value for relation classification task.

In this paper, we propose several strategies to incorporate entity pair information into deep neural network, in a way to provide definitive learning direction for relation classification task. In terms of sequence modeling, we employ the bidirectional GRU (Bi-GRU) as our main body. This structure can effectively alleviate the biased problem of unidirectional version (later inputs are more dominant than earlier inputs). With respect to entity pair, due to variable length, we firstly transform them into the high-level fixed-length embedding. Considering the occurrence order cannot be overlooked, we adopt a unidirectional GRU to model entity pair information into a fixed-length embedding. Several strategies have been attempted to integrate entity pair information into sequence modeling. First, we concatenate entity pair embedding with the intermediate feature embeddings, including word embeddings and the generated sentence representation by Bi-GRU. Despite a simple strategy, the practicability has been validated by many previous works. Zeng et al. [11] concatenate the automatically learned lexical vector and sentence level vector generated from CNN into the final sentence embedding to classify relation type; Wang et al. [19] connect the random-initialized aspect embedding with hidden vectors and word input vectors to rationally finish aspect-level sentiment classification. Second, we employ an entity pair-based attention mechanism to rationally allocate attention over words of input sentence. Recently, attention mechanism is a mature technique to enhance deep neural networks and has a wide range of applications [20, 21]. It is initially proposed for sequence-to-sequence learning. In addition, attention weights are computed under the adaptive prior knowledge; in other words, the prior knowledge for different input sentences is different. Inspired by this, according to the characteristic of relation classification task, we treat entity pair information as the adaptive prior knowledge to calculate attention weights, in a way to instruct Bi-GRU to better complete the mission. Through a series of experiments on the SemEval-2010 Task 8 dataset, it is demonstrated that the proposed methods effectively improve the previous performances of RNN and its variants; moreover, we compare the visualization of our entity pair-based attention mechanism and the original attention technique for relation classification. The comparable results intuitively reveal that the proposed entity pair-based attention mechanism yields more rational distribution.

The contributions of this paper can be summarized as follows:

(i) Without external linguistic features, this paper adequately exploits the implicit value of the information provided by entity pair and further improves the performance of relation classification system.

(ii) In order to adopt entity pair information to provide definitive direction for neural network, the utilization of entity pair information is from two different angles: concatenation operation and attention mechanism.

(iii) Aiming at the characteristic of relation classification task, the paper specially designs an entity pair-based attention mechanism which employs entity pair information to adaptively generate attention weights.

2. Related Works

Deep neural network has received increasing attention in the field of NLP, including relation classification task. The current successful deep learning structures all have been attempted to be tackled with relation classification task. Socher et al. [22] adopt a recursive neural network to extract features via the constituent parse tree of sentence. Zeng et al. [11] expand the concept of local receptive fields to natural language and employ a convolutional neural network to model sequence; moreover, the position feature is proposed to indicate the relative position of tokens and entity pair. Similarly, the external linguistic features are incorporated into both models to further enhance the performance, such as POS, grammatical relations, and WordNet hypernyms. By comparison, our model merely depends on the input sentences annotated with entity pair. Zhang et al. [23] adopt Bidirectional LSTM as main network to alleviate the bias problem of unidirectional RNN; equally, we also leverage the bidirectional structure, but LSTM is substituted by GRU in consideration of computational cost.

Concatenation operation is the common strategy for deep learning method to combine external features. Zeng et al. [11] concatenate the sentence embedding from CNN and the lexical feature embeddings into a compositional embedding and feed it into a full-connected layer to recognize relation

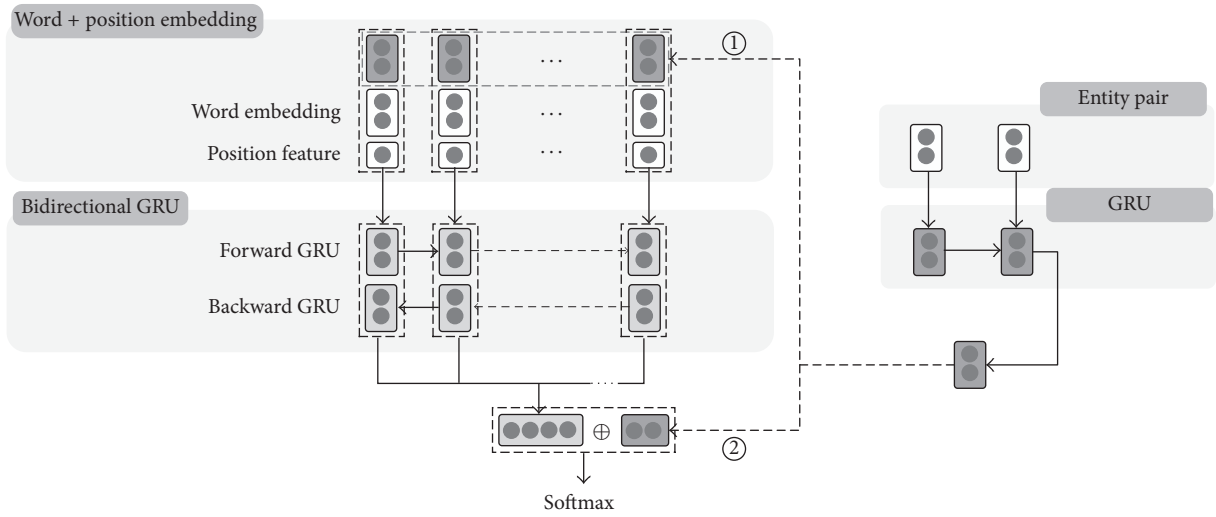

Figure 1: The structure of concatenation operation. As showed above, two substrategies are applied. With the obtained high-level entity pair embedding, substrategy ① concatenates it with word embedding and position embedding; substrategy ② adopts concatenation operation between sentence level embedding and entity pair embedding, where the compositional embedding is directly used to classify relation type.

type; Xu et al. [24], respectively, train four LSTMs to model word sequence, part-of-speech tags, grammatical relations, and WordNet hypernyms and then concatenate these four high-level feature representations into a global representation for the input instance. In addition to relation classification task, the same concept has been utilized to other NLP tasks. Wang et al. [19] propose an attention-based LSTM for aspect-level sentiment classification, where aspect information is converted into the aspect embeddings and connected with word embeddings and hidden vectors to provide crucial information for sentiment classification. Based on the same strategy, we parameterize the sentiment knowledge of entity pair into entity pair embedding and, respectively, adopt the concatenation operation into input layer and hidden layer to provide definitive learning direction for relation classification.

When we are reading, we tend to pay more attention to the words or phrases that are crucial for understanding; furthermore, for different purpose, the distribution of our attention is not quite the same. On this basis, attention mechanism is proposed to teach deep neural network to automatically calculate the attention distribution of input. Naturally, for different NLP tasks, different prior knowledge should be provided. Initially, attention mechanism is implemented in the sequence-to-sequence problem, like machine translation, parsing [25], and question answering [26]. The prior knowledge is provided by the tokens before the predicted token. However, for sequence-to-label problem, including document classification and relation classification, this idea is not workable. Aiming at this issue, for document classification task, Yang et al. [27] randomly initialize a unique vector, and the attention weights are calculated by dot product operation between this vector and corresponding word-level feature vectors. Following this idea, Att-BLSTM [28] is adopted for relation classification task. However, it is not persuasive that a random-initialized vector is capable of providing the adequate and reasonable prior knowledge; particularly, it is prone to overfitting when training set is in small scale. To overcome this dilemma, we employ the entity pair information to generate rational prior knowledge for detecting relation types. It is because, in most cases, different entity pairs have different relation types. More importantly, the semantic knowledge of entity pair can, to some extent, limit the search scope of relation types.

3. Methodology

Given a sentence annotated with entity mentions e_1 and e_2, relation classification system is to select a relation type of the highest confidence from a predefined relation set. In general, we adopt the bidirectional GRU to modeling input text sequence. To be specific, in terms of the input layer, we first convert words into word embedding; then, with respect to the hidden layer, two standard GRU, respectively, mine latent features from the opposite directions and generate high-level vector representations; finally, sentence representation is synthesized in the output layer and utilized to determine the ultimate relation type. The semantic knowledge carried by entity pair plays a vital role in providing definitive learning direction for deep neural network. Moreover, considering the occurrence order, the unidirectional GRU is adopted to parameterize entity pair into high-level embedding. In order to use this information to assist relation classification, two strategies are presented: concatenation and entity pair-based attention mechanism. Figures 1 and 2, respectively, depict the framework of the proposed methods.

3.1. Sequence Modeling. For relation classification, the available information of input consists of word sequence $S = \{w_1, w_2, \ldots, w_n\}$ and entity pair $[w_{e1}, w_{e2}]$; correspondingly, word embedding and position feature are employed to, respectively, reflect the information. Word embedding is

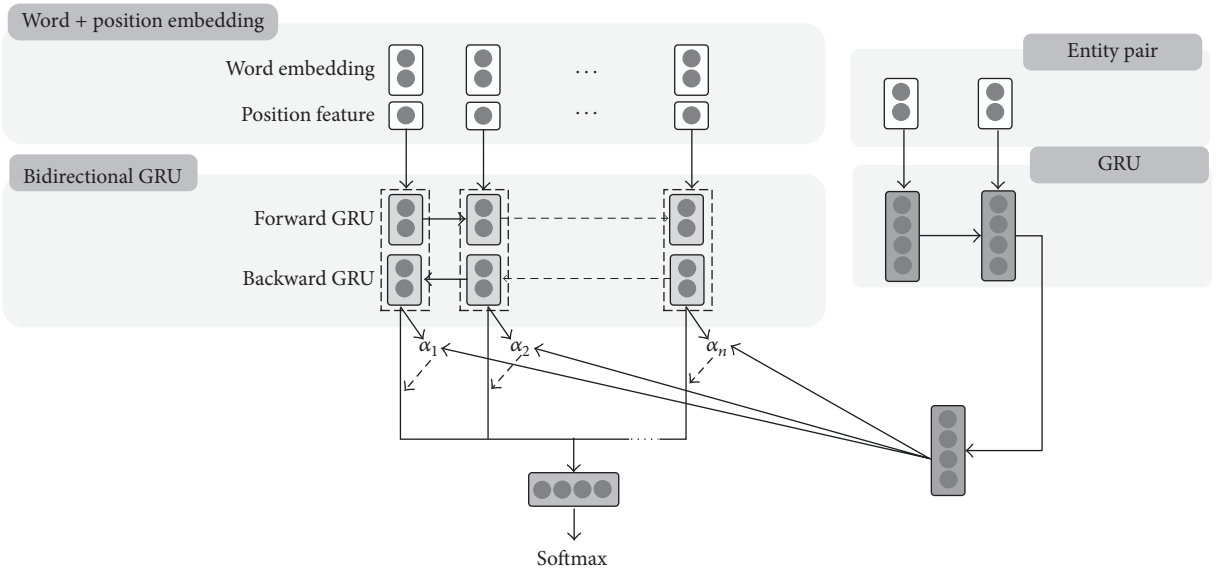

Figure 2: The overview of entity pair-based attention mechanism. With the generated entity pair embedding, the information distribution of word-level representations is measured by the dot product of entity pair embeddings and Bi-GRU word-level embeddings.

the distributed representation of word, which involves the semantic and syntactic information; particularly, the characteristic of low-dimension is beneficial to deep neural network. For word w_i presented in sentence, we look up the word embedding matrix $W^e \in \mathscr{R}^{d_e \times |V|}$ and extract the specific column to represent w_i as x_i; for words that cannot find the corresponding word embedding, we randomly initialize vectors for them. Position feature indicates the relative distance $[p_i^1, p_i^2]$ from w_i to entity pair $[w_{e1}, w_{e2}]$. $p_i \in \mathscr{R}^{d_p}$ is the vector representation of directional distance variable, which means the distance of word (distance on the left is negative value; otherwise it is positive value). Therefore, the overall input representation of w_i is $x_i^* = [(x_i)^T, (p_i^1)^T, (p_i^2)^T]$.

Gated Recurrent Units (GRU) is a mature neural network for modeling sequence information, which adopts the adaptive gating mechanism to alleviate the dilemma of vanishing and exploding of traditional RNN. Two gates are defined: the reset gate r_t and the update gate z_t. Under the control of these two gates, related information is prone to be remembered and noise information tends to be filtered by well-designed gates. For input text sequence, the hidden state h_i of i step is calculated by a linear interpolation between the previous hidden state h_{i-1} and the intermediate hidden state $\widetilde{h}_i$:

$$h_i = (1 - z_i) h_{i-1} + z_i \widetilde{h}_i. \tag{1}$$

The function of the update gate z_i is to define how much previous memory should be retained and how much new information should be added. z_i is determined by the input of i step and the previous state h_{i-1}:

$$z_i = \sigma\left(W_z x_i^* + U_z h_{i-1} + b_z\right), \tag{2}$$

where $\sigma(\cdot)$ denotes the logistic sigmoid function. The calculation of intermediate hidden state $\widetilde{h}_i$ is controlled by the reset gate r_i. It decides how to combine the new input with the previous memory.

$$\widetilde{h}_i = \tanh\left(W_h x_i^* + U_h\left(h_{i-1} \odot r_i\right) + b_h\right). \tag{3}$$

Analogously to z_i, r_i is jointly determined by the input of i step and the previous state h_{i-1}:

$$r_i = \sigma\left(W_r x_i^* + U_r h_{i-1} + b_r\right). \tag{4}$$

Standard GRU, also called unidirectional GRU, analyzes the input according to the word order and regards the last hidden output as the final text representation. However, when the length of input sequence is relatively long, it is inescapable to forget some crucial information. Thus, inspired by CNN, the final text representation can be computed by combining hidden states from all the time steps. From this perspective, in order to alleviate the unbalanced distribution of information, bidirectional structure is adopted. For every word in input sentence, Bi-GRU calculates two hidden states $\overrightarrow{h_i} \in \mathscr{R}^{d_h}$ and $\overleftarrow{h_i} \in \mathscr{R}^{d_h}$, respectively, from forward and backward direction. Thus, the final hidden state of w_i is represented as

$$h_i^* = \left[\overrightarrow{h_i}, \overleftarrow{h_i}\right]. \tag{5}$$

3.2. Entity Pair Embedding. Each instance is annotated with two predefined entity mentions, and in most cases entity mentions can be found in the vocabulary of word embedding. Therefore, firstly, we transform entity pair into a series of word embeddings. In the SemEval-2010 Task 8 dataset, every actual relation type has two subtypes, *like Component-Whole(e1,e2) and Component-Whole(e2,e1)*. Therefore, considering that the occurrence order of entity pair also reflects vital information and entity mentions are not of fixed length, we regard entity pair as a sequence

$$S_e = \{w_{e1}, \mathrm{sp}, w_{e2}\}, \tag{6}$$

where sp represents the split symbol whose embedding is randomly initialized and in the same dimension with word embedding. Then, we adopt a unidirectional GRU to model entity pair.

$$u_a = \mathrm{GRU}\left(S_e\right). \tag{7}$$

Particularly, because entity pair sequence is in relatively short length, we just use the output of the last hidden state as the entity pair embedding u_a.

3.3. Concatenation. The high-performance of deep neural network comes from its powerful ability to mine the latent features of input. However, because of this, the generated feature representations are uninterpretable. Consequently, there exist difficulties to incorporate external information into the neural network. A prevailing and simple solution is to generate the embeddings of external features and concatenate them with the intermediate vector representations. It is noteworthy that external embeddings should be simultaneously learned with deep neural network, in a way to unify the semantic space. Inspired by this, we design two subscenarios to integrate entity pair embedding into sequence modeling process.

(i) Concatenation in Input Layer. In order to indicate the influence of entity pair on the text sequence, position feature is proposed to identify the relative position of entity pair so that the words close to them are highlighted. However, this identification is merely in the structure level, which means it cannot demonstrate the semantic level influence of entity pair to other words. For the sake of using the specific information to targetedly adjust the semantic level of input layer, Wang et al. [19] concatenate the aspect embedding and each word embedding into a new input word-level embedding for aspect-level sentiment classification task. Following this idea, for every input representation x_i^*, the entity pair-level compositional input representation w_i^{ep} is represented as

$$x_i^{\mathrm{ep}} = \left[x_i^*, u_a\right]. \tag{8}$$

(ii) Concatenation in Output Layer. Besides combining entity pair information indirectly from word level, we also attempt to directly integrate entity pair embedding into the high-level sentence representation. In the previous work, Zeng et al. [11] employ CNN to generate sentence level representation and then connect the lexical level embedding with it to jointly predict relation type. Similarly, first, from the set of hidden output sequences $H = \{h_1^*, h_2^*, \ldots, h_i^*, \ldots, h_n^*\}$, we adopt the max-pooling operation to obtain the sentence level representation s, and the jth dimension of s is computed as follows:

$$s_j = \max_i h_{ij}^*, \quad \forall j = 1, \ldots, 2d_h. \tag{9}$$

Subsequently, the generated entity pair embedding u_a is directly concatenated with s_j to form the compositional sentence representation s^*:

$$s^* = s_j \oplus u_a, \tag{10}$$

where $\oplus$ denotes the concatenation operation.

3.4. Entity Pair-Based Attention Mechanism. The concatenation strategy mentioned above is a straightforward idea. In this section, we attempt to utilize the entity pair embedding to adjust the information distribution of the original intermediate vectors rather than directly add information to them. Attention mechanism is a specific technique for deep neural network, which is aimed at teaching computer to automatically pay attention to the vital part of input. From the bionics aspect, the starting point of this mechanism is rational, because when people are reading an article or watching a picture, it is natural to purposefully concentrate on the valuable part and alleviate the interference of the rest as far as possible. It is noteworthy that the premise is that the purpose should be provided by enough prior knowledge. Attention mechanism is initially proposed and applied in the field of question answering, machine translations, speech recognition, and image captioning. It can be easily found that these application tasks belong to the sequence-to-sequence problem. For different input sequence, the calculation of attention weight vector is injected with different prior knowledge; thus, there is explicit purpose. As for relation classification task, it is defined as a sequence-to-label problem, which cannot provide the discriminable a priori knowledge under the same setting. Faced with this demand, entity pair-based attention mechanism leverages the entity pair information to adaptively offer prior knowledge for the calculation of attention weight vector.

For each output h_i^* of Bi-GRU from the matrix $H = \{h_1^*, h_2^*, \ldots, h_n^*\}$, we first apply a nonlinear transformation operation and obtain u_i:

$$u_i = \tanh\left(h_i^*\right). \tag{11}$$

Then, we combine the entity pair embedding u_a with u_i to determine the importance of w_i for recognizing the relation type. The contribution of w_i is computed by dot product between these two vectors; in order to obtain the normalized attention weight distribution, we apply a softmax operation:

$$\alpha_i = \frac{\exp\left(u_i^{\mathrm{T}} u_a\right)}{\sum_i \exp\left(u_i^{\mathrm{T}} u_a\right)}. \tag{12}$$

Based on a set of attention weight α_i, we adopt two different schemes to calculate the final sentence representation s^*:

(i) Vector sum: in this strategy, the columns of hidden output matrix H are aggregated by the vector sum operation weighted on attention weight α_i:

$$s^* = \sum_i \alpha_i h_i^*. \tag{13}$$

(ii) Max-pooling: The max-pooling operation is to select the most striking feature in the specific feature dimension. Despite commonly being used in CNN, it is also applicable to compute the global sentence representation of RNN [23]. With the attention weight, the final sentence representation s^* is formulated as follows:

$$s_j^* = \max_i \left[a_i h_{ij}^*\right], \quad \forall j = 1, \ldots, 2d_h, \tag{14}$$

where s_j^* is the jth dimension of s^*.

3.5. Classification and Regularization. Due to the transformation of above layers, the obtained sentence representation s^* is capable of representing the input sequence and directly used to determine the relation type. First, we need to calculate the probability distribution of each candidate relation type y in the predefined relation set; thus, a softmax operation is applied as follows:

$$p(y \mid s^*) = \text{soft}\max\left(W_c s^* + b_c\right). \tag{15}$$

Then candidate relation type y with the highest probability value is designated as the predicted relation type $\hat{y}$:

$$\hat{y} = \arg\max_y p(y \mid s^*). \tag{16}$$

The whole model is trained in end-to-end way via gradient backpropagation, where the objective function is the cross-entropy error, and the goal of training is to minimize the cross-entropy error between y and $\hat{y}$:

$$\text{loss} = -\sum_i \sum_j y_i^j \log \hat{y}_i^j + \lambda \|\theta\|^2, \tag{17}$$

where i represents the index of sentence and j denotes the index of candidate relation type. Besides, L_2-regularization is also applied, where λ is the L_2-regularization term and θ stands for the parameter set.

Because of limited scale of training data, deep neural network with a great quantity of parameters is prone to overfitting. Therefore, dropout [29] is also leveraged to obtain more robust parameters. We exert the dropout rate ρ_w on the input embedding layer, ρ_a on the entity pair embedding, and ρ_c on the output layer. In addition, we also adopt Max-norm to avoid the blow-up of weights caused by huge learning rate. After a gradient descent step, the neural network is optimized under the constraint $\|W\|_2 \ll \varepsilon$, where ε is a tunable hyperparameter decided by validation set.

4. Datasets and Experimental Setup

We have validated and analyzed the practicability of our proposed method on the SemEval-2010 Task 8 benchmark (https://docs.google.com/document/d/1QO_CnmvNRnYwN-Wu1QCAeR5ToQYkXUqFeAJbdEhsq7w/preview) [30]. This benchmark is the commonly used relation classification dataset, which consists of 10717 sentences annotated with two entity mentions and a unique relation type. The whole dataset is previously divided into two parts: the training dataset of 8000 instances and the test dataset of 2717 instances. In terms of relation type, there are 10 predefined relation classes which include 9 actual classes and the *other* class. It is noteworthy that, due to the occurrence order of entity mentions, each actual class has two subclass, for example, *Message-Topic(e1,e2)* and *Message-Topic(e2,e1)*. In other words, not only should the relation classification system focus on the difference between relation classes, but also the distinction induced by direction should be concentrated on. Therefore, there exist 19 relation classes. Naturally, the official evaluation metric judges the performance with macroaveraged $F1$-score that takes directionality into account.

Table 1: Hyperparameter settings.

Hyperparameter	Value
d_e, d_p	300, 10
d_h	100
ρ_w, ρ_a, ρ_c	0.6, 0.2, 0.5
ε	3
Learning rate	10
Batch size	20

The words of input sequence are directly initialized by the pretrained word embedding set *GoogleNewsvectors-negative300.bin* (https://code.google.com/p/word2vec/). It is learned by Mikolov's word2vec tool and contains 300-dimensional vectors for 3 million words and phrases; thus, the vast majority of words in this benchmark can yield corresponding word embedding. With respect to out-of-vocabulary words, word embeddings are initialized from a Gaussian distribution $\mathcal{U}(-\epsilon, \epsilon)$, where $\epsilon = 0.01$. For the parameters matrices of the proposed model, we employ a Gaussian distribution to randomly initialize them. As for some hyperparameters, they are determined by a cross-validation procedure on a validation set which consists of 800 randomly selected examples. In addition, our end-to-end model is trained via AdaDelta [31] with a mini-batch size. The detailed settings are presented in Table 1.

5. Results and Discussions

This section presents a series of comparative experimental results and the detailed analyses to demonstrate the effectiveness of the proposed models. Firstly, compared with the previous works, we present the overall performance with the same relation classification benchmark. Subsequently, some concrete analyses from different angles have been elaborated to reflect the influence derived from the injection of entity pair information. Particularly, some visualization results are showed to reveal the superiority of the proposed attention mechanism against previous strategies.

5.1. Overall Performance. Table 2 has listed some representative relation classification systems. It is obvious that, in recent years, a variety of deep learning methods are in the dominant position and reveal their superiority of analyzing text sequence. In order to further promote the performance, some methods have combined the complicated human-designed features to assist the deep neural network; however, without external linguistic features, the proposed method still yields better performance.

(i) SVM [32] is the only one feature-engineering method mentioned in this paper, because, to some extent, it represents the best performance of traditional method, where 16 types of linguistic features are combined to generate the representation of input text sequence. Still, deep learning method is capable of

Table 2: Comparison with previous relation classification systems on SemEval-2010 Task 8 benchmark. Symbol "∘" means the experimental result is implemented by us. In the last line of the form, the performance of two strategies described in Section 3.4 is presented; they are represented as *Max-pooling* and *Sum*.

Model	Additional information	$F1$-score
SVM (Rink and Harabagiu 2010)	POS, Prefixes, Morphological, WordNet, Dependency Parse, Levin Classed, ProBank, FrameNet, NomLex-Plus, Google N-Gram, Paraphrases, TextRunner	82.2
MVRNN (Socher et al. 2012)	Word embedding, syntactic parsing tree	79.1
	+POS, NER, WordNet	82.4
CNN (Zeng et al. 2014)	Word embedding, position feature	78.9
	+WordNet, words around nominal	82.7
BRNN (Zhang and Wang 2015)	Word embedding	82.5
CR-CNN (Santos et al. 2015)	Word embedding	82.8
	+position feature	84.1
SDP-LSTM (Xu et al. 2015)	Word embedding	82.4
	+POS, GR, WordNet embedding	83.7
BLSTM (Zhang et al. 2015)	Word embedding	82.7
	+position feature, POS, NER, WNSYN, DEP	84.3
Att-BLSTM (Zhou et al. 2016)	Word embedding, Position Indicator	84.0
Att-BLSTM∘	Word embedding, position feature	83.5
Bi-GRU + InConcat	Word embedding, position feature	83.9
Bi-GRU + OutConcat	Word embedding, position feature	84.6
Bi-GRU + EAtt + Max-pooling	Word embedding, position feature	83.5
Bi-GRU + EAtt + Sum	Word embedding, position feature	**84.7**

obtaining the same performance without any external features.

(ii) MVRNN [22] is the pioneer, to the best of our knowledge, to utilize deep neural network to solve relation classification. The recursive neural network extracts latent features on the syntactic parsing tree of input text sequence and, simultaneously, additional parameter matrices are trained to modify the meanings of neighboring words. With several linguistic features, it achieves an $F1$-score of 82.4%. Our model directly uses text sequence as input, which effectively avoids the noise from wrong syntactic parsing results.

(iii) CNN [11] and CR-CNN [12] both adopt convolutional method to model sequence information. The convolutional parameters are shared by different local windows, in a way to reduce the number of parameters. CNN first proposes position feature, and it is similarly adopted in CR-CNN and the proposed method. Despite the efficiency of CNN, considering the importance of sequence information, particularly the occurrence order of entity pair, we employ a recurrent model, bidirectional GRU.

(iv) BRNN [33] and BLSTM [23] adopt the same bidirectional structure as our model. The difference is that we utilize the standard GRU for both directions, which not only alleviates the vanishing and exploding problem of RNN but also has lower computational complexity than LSTM.

(v) In SDP-LSTM [24], input text sequences are beforehand transferred into the shortest dependency paths with entity pair as the endpoints. A part of noise information is indeed removed; however, the error from the analysis of Dependency Parsing is naturally propagated into the identification of relation types. In addition, several manually annotated linguistic knowledge is incorporated to improve the performance. Yet, without external features, the proposed method still obtains the best $F1$-score of 84.7.

(vi) Att-BLSTM [28] equally introduces attention mechanism into the relation classification system. The prior knowledge merely comes from a unique randomly initialized vector, while our model adequately integrates entity pair information to adaptively calculate attention weights. It is noteworthy that Att-BLSTM utilizes Position Indicator [34] to annotate entity pair rather than position feature. In order to compare under the same condition, we reproduce its idea with position feature [11]. Under this circumstance, we obtain the improvement of 1.2% in $F1$-sore. The comparison results demonstrate the superiority of our method, which reflects that entity pair information effectively improves the performance of relation classification. Meanwhile, the practicability of the proposed three integration strategies is also proved.

Although all three proposed methods achieve good performances, there still exist some property differences among

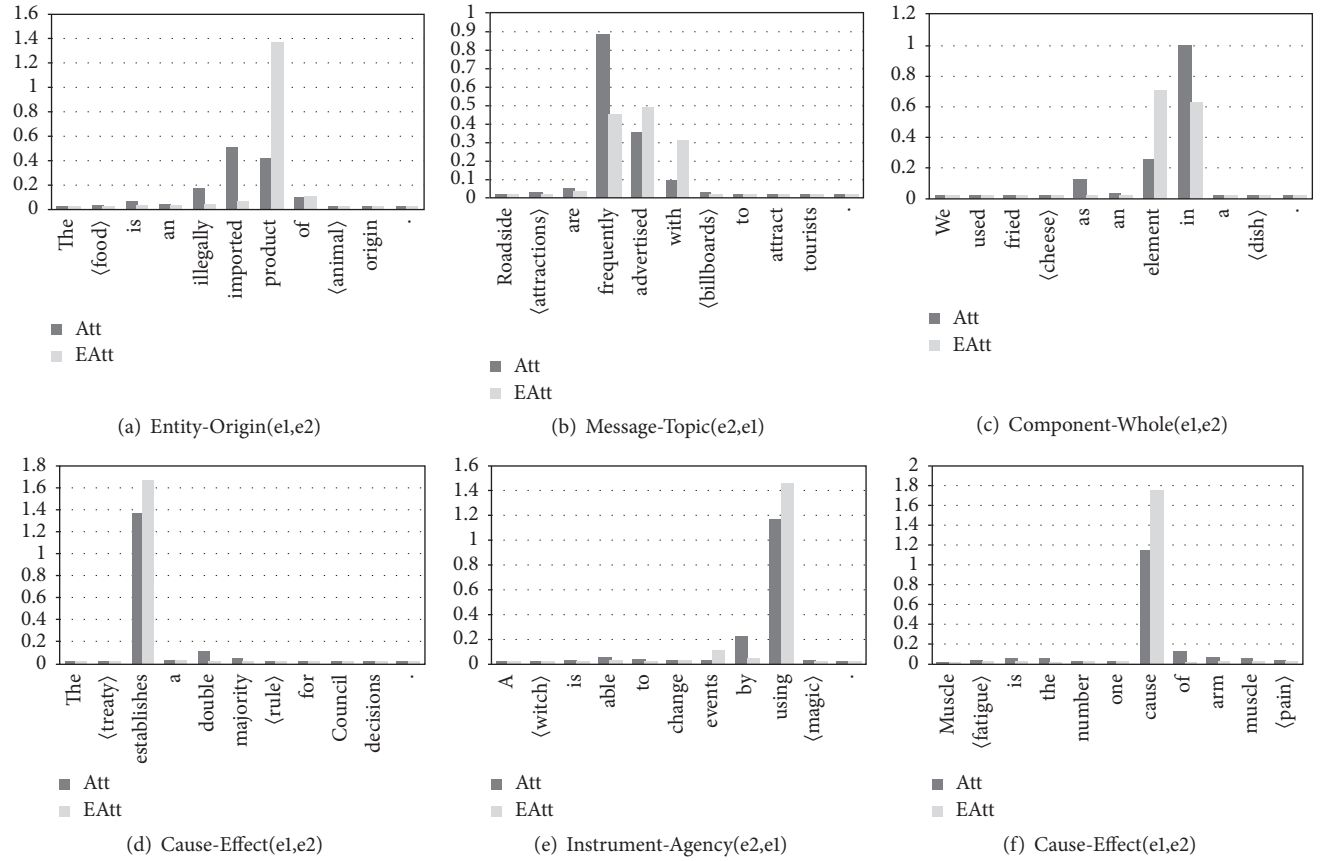

Figure 3: Attention visualization comparison between original attention mechanism and entity pair-based attention mechanism. "Att" denotes the attention mechanism which employs a random-initialized vector to generate attention weights; "EAtt" stands for the proposed entity pair-based attention mechanism.

them. First, only in the view of the overall $F1$-score, entity pair-based attention mechanism has the most prominent performance. Compared with concatenation operation, entity pair-based attention mechanism employs the entity pair information to adjust the original information distribution, in a way to instruct system to model relation classification task in a definitive direction. The reason of its superiority may be originated from the inexistence of the space inconsistence which is inevitable during concatenation operation. However, the gap between *Bi-GRU + OutConcat* and *Bi-GRU + EAtt* is not obvious, which indicates the above-mentioned inconsistence does not have obvious negative influence. Second, similarly in the concatenation operation, *Bi-GRU + OutConcat* has distinct advantage against *Bi-GRU + InConcat*. It is because the addition of entity pair embedding in input layer can, to some extent, lead to the bias of word embedding information, and these impacts are propagated into the subsequent calculation and thus influence the performance.

For entity pair-based attention mechanism, two strategies are applied to generate final sentence representations, vector sum, and max-pooling operation. As proved in previous works, for Bidirectional RNN structure without attention mechanism, two strategies are capable of obtaining similar performance. However, as presented in Table 2, vector sum operation reveals distinct superiority. The interpretation of this phenomenon is that attention mechanism has similar function to max-pooling. More concretely, attention mechanism is to assign soft weights for word-level information; however, max-pooling operation can be treated to allocate hard weights, which means "1" for the most important information and "0" for the rest. With the joint application, the information loss is aggravated and therefore causes negative influence.

5.2. Visualization of Entity Pair-Based Attention Mechanism. As previous works with attention mechanism, the practicability of the proposed attention mechanism can be intuitively reflected by the attention weight distribution of some specific examples. Similarly, we present some comparison results in Figure 3. For better visual effect, the actual attention weight α_i is adjusted according to the formula $(e^{\alpha_i} - 0.98)$. In addition, the entity pair is annotated with angle brackets in horizontal coordinate. These 6 histograms can be divided into two cases: histograms (a)~(c) demonstrate that two attention mechanisms give the highest attention weight for different words, but histograms (d)~(f) present the case that the same word is assigned the highest weight but the numeric values are different. In the first case, combining the annotated sentences with the relation types, it is easy to find that entity

pair-based attention mechanism assigns the feasible weight distribution for words, especially for the trigger word [35] (the words that convey the most obvious information for recognizing relation type), such as "advertised" for *Message-Topic(e2, e1)* and "element" for *Component-Whole(e1,e2)*. For the second case, despite the same highest-weight words, entity pair-based attention mechanism is capable of giving more distinctive weight for trigger words; in other words, entity pair-based attention mechanism has more excellent capability to enlarge the distinction between correct relation type and noise options.

6. Conclusion

The powerful learning capability of deep neural network enables it to well finish various NLP tasks merely from word sequence as input. Even so, for relation classification, the noise information of word sequence still, to some extent, brings negative impact. Considering the annotated entity pair reflect crucial information for the importance contribution of words, we propose three strategies, including two concatenation operations and entity pair-based attention mechanism, to employ the implicit semantic information involved in entity pair to provide definitive learning direction for neural network. The experimental results of SemEval-2010 Task 8 benchmark demonstrate the effectiveness of the proposed strategies. Entity pair-based attention mechanism achieves the best $F1$-score because the attention weights are adaptively calculated from entity pair information. Despite the suboptimal performances of concatenation operation, the gaps are not obvious and the superiority is still distinct against most of the previous works. In conclusion, without external linguistic features, the proposed strategies effectively apply entity pair information to instruct deep neural network to pay more attention to significant semantic information, in a way to further improve the performance of relation classification system.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] S. Yin, H. Yang, and O. Kaynak, "Sliding mode observer-based FTC for markovian jump systems with actuator and sensor faults," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3551–3558, 2017.

[2] S. Yin, H. Gao, J. Qiu, and O. Kaynak, "Descriptor reduced-order sliding mode observers design for switched systems with sensor and actuator faults," *Automatica*, vol. 76, pp. 282–292, 2017.

[3] S. Yin, H. Gao, J. Qiu, and O. Kaynak, "Fault detection for nonlinear process with deterministic disturbances: a just-in-time learning based data driven method," *IEEE Transactions on Cybernetics*, 2016.

[4] R. Grishman, "Information extraction: Techniques and challenges," in *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*, vol. 1299 of *Lecture Notes in Computer Science*, pp. 10–27, Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[5] Z. Luo, "Improving Twitter Retrieval by Exploiting Structural Information," *AAAI*, 2012.

[6] Y. Zhang, S. He, K. Liu, and J. Zhao, "A joint model for question answering over multiple knowledge bases," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 3094–3100, usa, February 2016.

[7] M. Bienvenu, C. Bourgaux, and F. Goasdoúe, "Explaining inconsistency-tolerant query answering over description logic knowledge bases," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 900–906, usa, February 2016.

[8] X. Zhao, H. Yang, H. R. Karimi, and Y. Zhu, "Adaptive neural control of MIMO nonstrict-feedback nonlinear systems with time delay," *IEEE Transactions on Cybernetics*, vol. 46, no. 6, pp. 1337–1349, 2015.

[9] X. Zhao, P. Shi, and X. Zheng, "Fuzzy Adaptive Control Design and Discretization for a Class of Nonlinear Uncertain Systems," *IEEE Transactions on Cybernetics*, vol. 46, no. 6, pp. 1476–1483, 2016.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proceedings of the COLING*, pp. 2335–2344, 2014.

[12] C. N. D. Santos, B. Xiang, and B. Zhou, "Classifying relations by ranking with convolutional neural networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 2015, https://arxiv.org/abs/1504.06580.

[13] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *Neural and Evolutionary Computing*, 2015, https://arxiv.org/abs/1510.03820.

[14] T. Mikolov, "Recurrent neural network based language model," in *Interspeech*, 2010.

[15] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proceedings of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011*, pp. 5528–5531, cze, May 2011.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] K. Cho, B. van Merrienboer, C. Gulcehre et al., "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014.

[18] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *Journal of Machine Learning Research (JMLR)*, vol. 3, no. Spec. Issue Machine Learn. Methods Text Images, pp. 1083–1106, 2003.

[19] Y. Wang, M. Huang, x. zhu, and L. Zhao, "Attention-based LSTM for Aspect-level Sentiment Classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 606–615, Austin, Texas, November 2016.

[20] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, 2014.

[21] T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural*

Language Processing, pp. 1412–1421, Lisbon, Portugal, September 2015.

[22] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012*, pp. 1201–1211, kor, July 2012.

[23] S. Zhang et al., "Bidirectional long short-term memory networks for relation classification," *PACLIC*, 2015.

[24] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "Classifying relations via long short term memory networks along shortest dependency paths," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pp. 1785–1794, prt, September 2015.

[25] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," *Advances in Neural Information Processing Systems*, pp. 2773–2781, 2015.

[26] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proceedings of the 29th Annual Conference on Neural Information Processing Systems, NIPS 2015*, pp. 2440–2448, can, December 2015.

[27] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical Attention Networks for Document Classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, San Diego, California, June 2016.

[28] P. Zhou, W. Shi, J. Tian et al., "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, pp. 207–212, deu, August 2016.

[29] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors , Computer Vision and Pattern Recognition,.

[30] I. Hendrickx, S. N. Kim, Z. Kozareva et al., "SemEval-2010 task 8," in *Proceedings of the the Workshop*, p. 94, Boulder, Colorado, June 2009.

[31] M. D. Zeiler, *ADADELTA: an adaptive learning rate method*, 2012.

[32] B. Rink and S. Harabagiu, "Utd: Classifying semantic relations by combining lexical and semantic resources," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, Association for Computational Linguistics, 2010.

[33] D. Zhang and D. Wang, "Relation classification via recurrent neural network," *Neural and Evolutionary Computing*, 2015.

[34] P. Qin, W. Xu, and J. Guo, "An empirical convolutional neural network approach for semantic relation classification," *Neurocomputing*, vol. 190, pp. 1–9, 2016.

[35] W. Xu and C. Zhang, "Trigger word mining for relation extraction based on activation force," *International Journal of Communication Systems*, vol. 29, no. 14, pp. 2134–2146, 2016.