IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 5, 2018, accepted November 22, 2018, date of publication January 14, 2019, date of current version March 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2892724

Shared Representation Generator for Relation Extraction With Piecewise-LSTM Convolutional Neural Networks

DANFENG YAN AND BO HU
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding authors: Danfeng Yan (yandf@bupt.edu.cn) and Bo Hu (jadfi@bupt.edu.cn)

This work was supported in part by the National Key R&D Program of China under Grant 2018YFC0831502, and in part by the National Technology Project under Grant 520626170011.

ABSTRACT Traditional distant supervision for relation extraction is faced with the problem of introducing noises. In this paper, we present a shared representation generator to de-emphasize the noisy expressions by extracting common features in relation. Different from computing weighted sum in widespread attention mechanism, we directly generate bag representation in multi-instance learning by feature transformation, which only remains the semantics related to predict relation. We introduce the generator loss into objective function to improve the performance of shared representation. Also, the structure of our proposed generator is flexible and scalable. To capture more structural information, piecewise convolutional neural network (PCNN) is widely used to divide the output of convolutional layer into three segments, but this approach breaks the consistence and inner relationship of the sentence. We encode the sentence with piecewise-LSTM convolutional neural network (PLSTM-CNN) to alleviate this issue, which adopts BiLSTM after the pooling layer of PCNN. The experimental results show that we achieve significant improvement on relation extraction as compared with the baselines.

INDEX TERMS Distant supervision,relation extraction, shared representation,generator loss,BiLSTM.

I. INTRODUCTION

Relation extraction is the basic task in text mining and information extraction, and has been applied in many practical scenarios [1], [2]. With the development of large-scale knowledge bases such as Freebase [3] and DBpedia [4], more and more NLP tasks rely on these KBs to acquire the high-quality training examples. Considering that the existing KBs will be far from complete if only depending on the manually labeling, relation extraction has been attracted much attention from researchers. It is an important way to supplementary KBs.

The study of relation extraction mainly employs the supervised relation classification for a known entities (e_1, e_2). One of the difficult problems of classification is the need for a large number of training samples. However, a lot of manual annotation is time-consuming and inefficient. Distant supervision [5] is introduced to automatically annotate corpus by using the existing open KBs. There is a strong hypothesis in distant supervision that if two entities have some kinds of relationship, this relationship exists in any sentence containing both two entities at the same time. Take Freebase for example, (*Founders*, *Apple*, *SteveJobs*) is a relational triple tuple, so ‘‘[Steve Jobs] was the co-founder of [Apple]’’ and ‘‘[Steve Jobs] passed away the day before [Apple] unveiled iPhone 4S’’ are both the training samples generated by distant supervision.

However, there are two obvious problems in distant-supervised samples generation. 1) False positive. In the second example, the mention does not express the relation of *Founders* but it's labeled by distant supervision falsely. So the noises are introduced. 2) False negative. The relation not in KBs will be labeled as *NA*, although there is a relationship in two entities. It is caused by the incompleteness of KBs.

Multi-instance learning is adopted by Riedel *et al.* [6] to alleviate the wrong labeling problem. The classification is not based on sentence level but on bag level. The bag with relation r is formed by all sentences containing both e_1 and e_2 for triple (r,e_1,e_2) in KB. Each sentence is an instance and has no relation category. Only the bag has relation labels. Suppose that the training set has N bags B_1, B_2,..B_N and the relation labels of bags are R_1,R_2,..R_N, the i-th bag contains M_i

instances, B_i={s_{i1},s_{i2},..s_{iM_i}}. The objective is to utilize these bags to train the classifier and predict the relation for unseen entities.

In multi-instance learning, the bag representation dominates the performance of classification. Lin *et al.* [7] take the attention mechanism into representation of bags and outperforms the past work, which is widely used by other researchers. However, attention mechanism is too simple and indirect to generate the bag embeddings. It is hard for the network to learn appropriate attention weights because there is no enough supervisory information in loss function. With the development of generative adversarial networks(GAN) [8], some researchers utilize the generator to extract the common features. Liu *et al.* [9] use generator to learn the shared information among multiple tasks. Inspired by the work, we construct a shared representation generator for each relation to extract shared features among instances. We abandon the attention mechanism and directly generate the bag embeddings. In order to decrease the complexity of network, we introduce the generator loss instead of discriminator to play a supervisory role and improve the performance. Note that our network is not adversarial because there is no discriminator and adversarial process, which can be considered in the feature work.

Owing to the limitation of hand-designed features, researchers are tend to apply deep learning to encode instance in relation extraction. Zeng *et al.* [10] propose piecewise convolutional neural network(PCNN) according to position of entities, which has been proved to be more effective than traditional CNN. But the inner relation across three pieces is ignored. We consider the three segments after max-pooling as a sentence consisting of three words. Therefore, the approach to represent the sentence is extensible. In this paper we exploit BiLSTM [11] after pooling layer of PCNN to further enhance the representation of instances, which can extract more structure features. To the best of our knowledge, this is the first effort to introduce LSTM layer into PCNN.

The main contributions of this paper are: 1) Propose a new method to directly generate the bag representation. We construct a generator for each relation to capture the common features; 2) Introduce extra loss to supervise the performance of generator; 3) Introduce BiLSTM after the pooling layer of PCNN to obtain better representation of sentence, learning more interactive information across pieces. As far as we know, we first focus on the inner-relationship among three pieces in PCNN; 4) Our method is flexible and has good scalability. BiLSTM can be replaced by other sequential modeling methods and the shared representation generator can be constructed as the generator of GAN does.

The rest of this paper is organized as follows: Section II will describe the related work of relation extraction. Section III will elaborate on the principles. In section IV, we conduct experiments to illustrate the validity of our method. The results and related analysis will be shown detailedly. The last section concludes our work and the prospects for the further work are presented.

II. RELATED WORK

As one of the most important tasks in NLP, many efforts have been invested for relation extraction, especially in supervised classification. In view of the fact that manually labeling is inefficient, Mintz *et al.* [5] propose distant supervision to utilize existing knowledge bases. If there is a relationship between two entities in KBs, all sentences mention the both entities will be labeled as the relationship. This hypothesis is not supported in real world, leading to the introducing of noise. Riedel *et al.* [6] employ multi-instance learning to weaken the influence of noise. With the development of deep learning, neural network is adopted in relation extraction. Liu *et al.* [12] and Zeng *et al.* [13] exploit simple CNN to extract the features of text automatically, outperforming the rigorously engineered feature-based or kernel-based models [14], [15]. Huang and Wang [16] introduce ResNet [17] to improve the performance of CNN. According to the position of two entities, Zeng *et al.* [10] propose PCNN to pool the sentence in three pieces. This way can learn more context information that is relevant to entities. In addition, Zeng *et al.* [13] employ position embeddings to express the sentences better. Also, at-least-one multi-instance learning is adopted to address the noise in distant supervision. Although it alleviates the problem, most of valid information is lost. Lin *et al.* [7] employ attention in multi-instance learning to overcome the problem. The representation of bag is obtained by the weighted sum of instances vector, making full use of all informative sentences. So attention mechanism is used widely in relation extraction.

Wu *et al.* [18] introduce adversarial training based on the work of Lin *et al.* [7], improving the robustness of relation extraction. Adversarial noises are added in word embeddings and a new loss function is proposed. Ji *et al.* [19] exploit (e_1-e_2) to represent the relation between entity e_1 and entity e_2 and calculate the attention weights between instances and (e_1-e_2). To improve the representation of entity embeddings, the description information of entity is extracted from Freebase and Wikipedia. Sorokin and Gurevych [20] apply attention to the representation of sentence rather than bag level. The method considers relationship between other entities and the given entities in one sentence. Besides, Jat *et al.* [21] also employ attention of words and relation. In addition to attention mechanism, Jiang and Li [22] exploit cross-sentence pooling to produce the bag representation, only keeping the maximum value of every dimension of sentence vectors. Liu *et al.* [23] propose a soft label to correct the wrong labeling bags, which is automatically adjusted by the network in training.

In this paper, we propose a shared representation generator to learn the common features, which is inspired by the application of GAN. Liu *et al.* [9] adopt GAN to classify texts by adversarial multi-task learning. Two samples from two different tasks only remain the shared features after passing the generator, and the discriminator tries to determine whether two samples are from the same task. The generator can also be served as mapping function, mapping original space to the

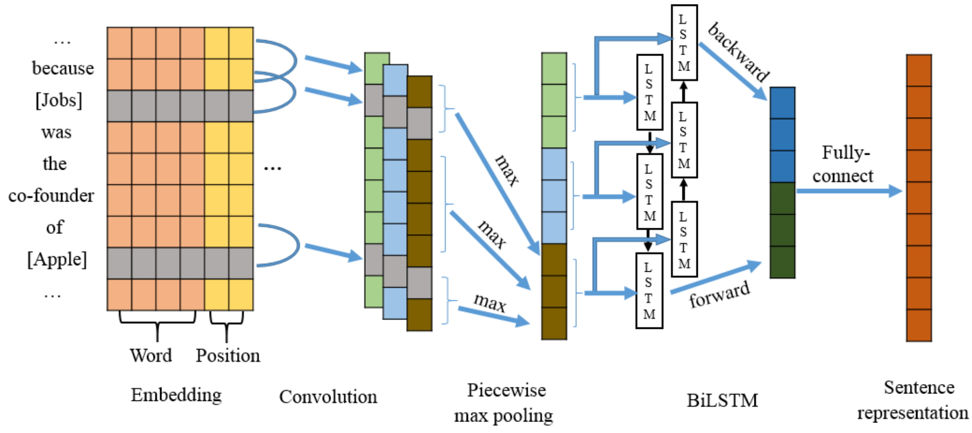

FIGURE 1. The architecture of PLSTM-CNN used for sentence encoder.

target space. Zhang *et al.* [24] use GAN in bilingual lexicon translation. The generator is implemented to map the source word embedding to target embedding. In our work, the original instance representation space is mapped to shared features space by generator.

III. METHODOLOGY

In this section, we will introduce our model for relation extraction. The procedure includes two main parts:

1) Sentence Encoder: Zeng *et al.* [10] have proved that PCNN can extract more relational information between entities. But it ignores the inner-relationship among three pieces. So we exploit BiLSTM to make full use of the interactive information. Each sentence in one bag will be encoded by PLSTM-CNN.

2) Shared representation generator: We construct a shared representation generator to directly generate the bag embeddings. Then we train the relation classifier based on bag level and predict the relation for unseen entity pairs. In order to improve the bag representation, we introduce generator loss in the objective function.

A. SENTENCE ENCODER

In order to encode the sentence, we first transform the word tokens into vectors by looking up the pre-trained word embeddings matrix. Position information makes a difference in sentence encoding, so we employ position embeddings to specify the relative positions between words and entities. After concatenating the word and position embeddings to represent the input of the sentence, PCNN is used to do convolutions and piecewise max pooling. To extract features across three-pieces, we exploit BiLSTM into the output of pooling layer. Then a fully-connected layer is utilized to adjust the dimension of output vector and obtain the final sentence encoding.

Fig.1 illustrates the details of our PLSTM-CNN. As shown in Fig.1, PCNN will divide the sentence into three segments to pool based on the position of entity pairs.As the result, the output dimension of PCNN is three times that of CNN. BiLSTM is adopted for the output of PCNN to learn the inner-relationship and the continuity among three pieces. In our method, BiLSTM only has three timesteps.

1) WORD EMBEDDINGS

Word embeddings is the fundamental work in NLP. The objective of word embeddings is to map each raw word in real word to the distributed vector. The similarity between words can be represented as the distance in vector space. As one of the most widely used methods, we use word2vec [25] to train the words and obtain the words embeddings matrix $V \in \mathbb{R}^{d_w \times |V|}$, where V is a fixed-sized vocabulary and d_w is the size of word embedding. Then the input words are transformed by looking-up tables.

2) POSITION EMBEDDINGS

In this paper, We employ the position embedding following Zeng *et al.* [13]. A PF is defined as the combination of the relative distances from each word to corresponding entities e_1 and e_2. For example, "[Steve] is the CEO of [Apple]", the relative distance form "CEO" to "Steve" and "Apple" is 3 and -2.

Similar to word embeddings, we transform the relative distance into position embeddings by looking up the position matrix. The matrix is initialized randomly and the embeddings are fine-tuned by neural network in the process of deep learning.

Because there are two entities in one sentence, PF_1 and PF_2 are initialized to specify the position information. Then we concatenate the word embeddings and position embeddings to express the full information of words. We denote the size of position embedding as d_p, the input of sentence can be represent as the matrix $S \in \mathbb{R}^{L \times d}$, where L is the length of sentence and $d = d_w + d_p * 2$.

Note that the positional embedding approach is scalable enough to handle sentences with more than 2 entities. In multi-instance learning, sentences containing the same

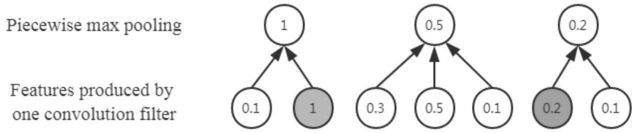

FIGURE 2. An example of piecewise max pooling.

entity pair form a bag together. If there are more than 2 entities, the sentence will appear in more than one bag and each bag represents an entity pair.

3) CONVOLUTION LAYER AND PIECEWISE MAX POOLING

A convolution operation is utilized to extract the local features, which will be merged to produce the global features. We apply a window of q words to a filter $h \in \mathbb{R}^{q\times d}$. If the sentence contains L words is denoted as $S = \{x_1, x_2, ..x_L\}$. $x_i \in \mathbb{R}^d$ represents the concatenating of word embeddings and position embeddings. So a feature c_i is generated by a filter as:

$$c_i = f(W \odot x_{i:i+q-1} + b) \tag{1}$$

Here f is a non-linear activation function. We use tanh in this paper. And $\odot$ represents the dot product. $W \in \mathbb{R}^{q\times d}$ is a parameters matrix and $b \in \mathbb{R}$ is a bias term.

A convolution filter will slide from left to right of the sentence and merge all features to produce global feature $c = [c_1, c_2, ..c_{L-q+1}]$, where $c \in \mathbb{R}^{L-q+1}$. It is noteworthy that we set a maximum sentence length L for the input to the convolution layer. And we apply padding in case of shorter sentences. The fixed placeholder will be used to fill in parts of sentences that are not long enough, which adopts random initialization.

Afterwards, a max-pooling layer is applied to remain the most important feature in c. Traditional max-pooling only produce maximal feature, but it's insufficient to capture fine-grained features for relation extraction. Zeng *et al.* [10] propose piecewise max pooling to return the maximum value in three segments, which is divided according to the position of head and tail entity. So c can be divided as $\{c_1, c_2, c_3\}$ and the output of piecewise max pooling generated by i-th filter is defined as:

$$p_i = [max(c_1), max(c_2), max(c_3)] \tag{2}$$

From (2) we can see that in piece-wise pooling each filter will produces three maximum value with max-pooling in three pieces respectively. But every filter in CNN only produces one maximum value with max-pooling.After piecewise max pooling, the output is denoted as $p \in \mathbb{R}^{3m}$, where m is the number of filters. Fig.2 shows an example of piecewise max pooling. The deep color in the figure corresponds to the position of head and tail entity in sentence.

4) BILSTM AND FULLY-CONNECTED LAYER

Furthermore, we employ LSTM to extract the inner-relationship among $\{c_1, c_2, c_3\}$. LSTM is one of widespread adoption of recurrent architectures and it is able to learn the sequential information. As LSTM tends to be biased toward the most recent inputs, we exploit BiLSTM to concatenate the representation of forward LSTM and backward LSTM. Take the forward LSTM for example, we denote the initial state as h_0, the recurrent state transition step for calculating $h_1, h_2, ..., h_{n+1}$ is defined as follows [26]:

$$\begin{aligned}
\hat{i}^t &= \sigma(W_i x_t + U_i \overrightarrow{h^{t-1}} + b_i)\\
\hat{f}^t &= \sigma(W_f x_t + U_f \overrightarrow{h^{t-1}} + b_f)\\
o^t &= \sigma(W_o x_t + U_o \overrightarrow{h^{t-1}} + b_o)\\
u^t &= tanh(W_u x_t + U_u \overrightarrow{h^{t-1}} + b_u)\\
i^t, f^t &= softmax(\hat{i}^t, \hat{f}^t)\\
c^t &= c^{t-1} \odot f^t + u^t \odot i^t\\
\overrightarrow{h^t} &= o^t \odot tanh(c^t)
\end{aligned} \tag{3}$$

We denote the word representation as x_t. i_t,o_t,f_t and u_t represent the values of an input gate, an output gate, a forget gate and an actual input at time step t, which controls the information flow for a recurrent cell c_t and the state vector h_t. W_x and U_x are weight matrices and b_x are bias vectors, where $x \in \{I, f, c, o\}$. The symbols $\sigma(\cdot)$ represents the element-wise sigmoid function and $\odot$ refers to the element-wise multiplication.

We apply BiLSTM model to the result of pooling layer. we denote the output of BiLSTM layer as s_b, which is the concatenation of the forward and backward hidden state. s_b is defined as:

$$s_b = [\overrightarrow{h^3}; \overleftarrow{h^0}] \tag{4}$$

In our method, we adopt BiLSTM for the output of piecewise max pooling. So the input length of BiLSTM is $3 * m$, where m is the number of filters in convolution layer. Our BiLSTM only has three timesteps because there are three pieces. s_b is concatenated by the forward and backward hidden states, so the output length is $2 * L_b$,where L_b represents the dimension size of hidden states.

Traditional PCNN considers the output of piecewise max pooling as the sentence vector, so the size of dimension is $3{*}m$. In order to avoid the interference of sentence dimension, we introduce a fully-connected layer to make adjustment after BiLSTM. Then the final sentence encoding s can be obtained as follows:

$$s = W_b s_b + b_b \tag{5}$$

where $W_b \in \mathbb{R}^{3m\times 2L_b}$represents the weight matrix of fully-connected layer and b_b represents the bias value.

B. SHARED REPRESENTATION GENERATOR

1) BAG REPRESENTATION

In multi-instance learning, the classification is based on bag level rather than traditional sentence level. Thus how to obtain better bag representation is a key point to improve the performance of classification. Attention mechanism is

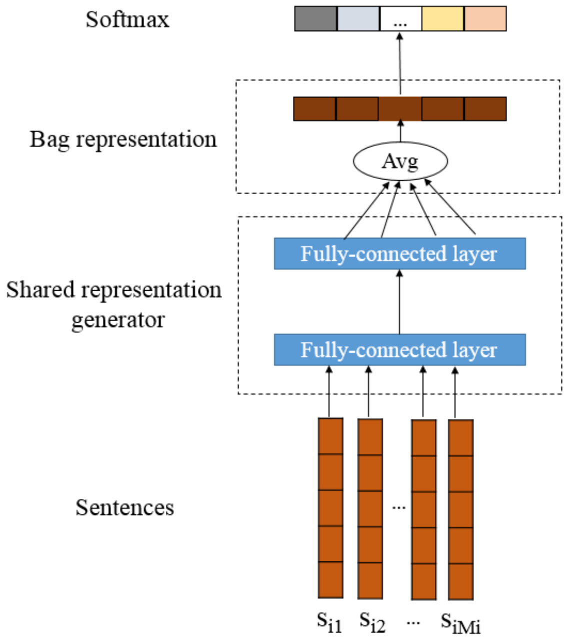

FIGURE 3. The overview of relation extracting using the shared representation generator.

highly applied in [7] and [19]–[21]. The bag embeddings are computed by the weighted sum of instance embeddings. The weights are calculated to score how well the sentence and the predict relation matches. However, the relation representation is randomly initialized and there is no extra loss to supervise the attention mechanism, so it's hard for the network to learn the appropriate weights.

In this paper, we propose a generator to directly learn the shared representation for a bag. Our target is to transform the sentence vector form original semantic space to corresponding relation representation space. After passing the generator, the transformed vector can be regarded as the expression of sentence on the corresponding relation. Irrelevant noise semantics will be filtered. So the output of generator is a shared representation, extracting the common features for the relation. Different form attention mechanism, our algorithm is a feature transformation method rather than feature weighting process. Feature transformation has better non-linear expression ability.

As Fig.3 shows, we adopt two fully-connected layers network to serve as generator. We will construct a generator for each relation. Given a set of sentence in i-th bag $B_i = \{s_{i1}, s_{i2}.., s_{iM_i}\}$, where s_{ij} represents the sentence vector computed by PLSTM-CNN and M_i represents the number of sentences in i-th bag.We denote the predict label as r_i The transformed vector g_{ij} is calculated as follows:

$$\begin{aligned} h_{ij} &= tanh({W_f^{r_i}}_1^{\mathsf{T}} s_{ij} + {b_f^{r_i}}_1) \\ g_{ij} &= tanh({W_f^{r_i}}_2^{\mathsf{T}} h_{ij} + {b_f^{r_i}}_2) \end{aligned} \tag{6}$$

where ${W_f^{r_i}}_k$ and ${b_f^{r_i}}_k$ ($k \in \{1, 2\}$) are the weight matrix and the bias value respectively, which represent the parameters of generator for r_i. Because g_{ij} only remains the semantic information of relation r_i, we can compute the bag representation G_i using the average vector of all transformed sentence vectors:

$$G_i = \frac{1}{M_i} \sum_{j=1}^{M_i} g_{ij} \tag{7}$$

where $G_i \in \mathbb{R}^{L_d}$, L_d represents the size of bag vector.

It is worth noting that for each relation we will construct a generator to produce the shared representation for every bag which is tagged with this relation. In other words, each relation has its own generator and it acts on each bag which belongs to this relation. At training stage, all sentences in each bag will pass the generator of the corresponding relation which this bag belongs to and then the output of generator can be considered as the common features or shared representation in this relation. At the testing stage, each bag will pass every generator because we do not know the true relation label. Then maximum probability in softmax layer will determine the predicted relation label.

2) GENERATOR LOSS

As for multi-class classfication, we use cross-entropy as the base loss function. We denote the corresponding relation label of i-th bag as r_i. Then the conditional probability of relation is:

$$p(r_i|G_i; \theta) = \frac{exp(o_i)}{\sum_{j=1}^{N_r} exp(o_j)} \tag{8}$$

where θ indicates all parameters in our models,N_r indicates the number of relation labels and $\mathbf{o}$ indicates the output of softmax layer. To compute the confidence of relation, we feed the bag vector G_i into a softmax classifier:

$$\mathbf{o} = W_s G_i + b_s \tag{9}$$

where $\mathbf{o} \in \mathbb{R}^{N_r}$ represents the probability matrix for each relation. $W_s \in \mathbb{R}^{N_r \times L_d}$, $b_s \in \mathbb{R}^{N_r}$ represent the weight matrix and offset value of softmax layer respectively. Then we compute the cross-entropy as follows:

$$J_1(\theta) = \sum_{i=1}^{T} logp(r_i|G_i; \theta) \tag{10}$$

where T is the number of training bags. In addition, we introduce extra loss to make generator perform better. Our target is to preserve the semantic components of a bag that are related to relation label. So the transformed sentence embeddings after passing the generator should be as close as possible because the common features should express the same semantics. We define the generator loss as the average distance from each transformed sentence to the center:

$$\begin{aligned} J_2(\theta) &= \frac{1}{T} \sum_{i=1}^{T} (\frac{1}{M_i} \sum_{j=1}^{M_i} (g_{ij} - c_i)^2) \\ c_i &= \frac{1}{M_i} \sum_{j=1}^{M_i} g_{ij} \end{aligned} \tag{11}$$

where M_i is the number of sentences in the bag and c_i is the center representation of a bag. In this paper, it is equivalent to the bag representation G_i. Then we compute the final objective function as the weighted sum of cross-entropy and generator loss:

$$J(\theta) = J_1(\theta) + \alpha J_2(\theta) \tag{12}$$

where α represents the weight of generator loss. It is the hyper parameter and $\alpha > 0$.

Note that in this paper we are devoted to proposing a new method to represent bag vector in multi-instance learning instead of attention mechanism which is widely used in the past work. We present the idea of shared representation generator to generate the bag vector. Out of simplicity, we only adopt fully-connected network to implement the generator and it is easy to understand. However, an important advantage is that the implementation of generator is very flexible and scalable. Actually other more complicated structures can also be introduced like what GAN does, such as CNN [27], LSTM [28], autoencoder [29] and so on. It is not our focus in this paper because the simple structure is enough. But we will explore and discuss more complicated networks in the feature work.

IV. EXPERIMENTS AND RESULTS ANALYSIS

In this section, we conduct experiments and demonstrate that our models can improve the performance of distant supervised relation extraction. We first introduce the experimental dataset. Then we describe the parameter settings determined by cross-validation in our experiments. Finally, we compare our methods with state-of-the-art neural network baselines. PCNN+ATT is proposed by Lin *et al.* [7]. Lin *et al.* [7] introduced attention mechanism into relation extraction and achieved the state-of-the-art performance. Then attention mechanism is widely used by other researchers. The work of Lin *et al.* [7], Liu *et al.* [23], Luo *et al.* [30] has shown that PCNN+ATT, which we use as baseline, is better than the traditional models proposed by Mintz *et al.* [5], Hoffmann *et al.* [31] and Surdeanu *et al.* [32]. So we only compare with one model, i.e. PCNN+ATT, just like what Wu *et al.* [18] does.

PLSTM-CNN+ATT method introduces our proposed PLSTM-CNN, PCNN+GEN adopts generator to produce the bag vector rather than attention mechanism and PLSTM-CNN+GEN represents our model and PLSTM-CNN+GEN+NOLOSS represents our model without generator loss.The results show that our model outperforms baselines and we will discuss it detailedly.

A. EXPERIMENTAL DATA AND EVALUATION METRICS

We evaluate our methods with two different datasets. NYT-dataset is developed by Riedel *et al.* [6] and Hoffmann *et al.* [31] which aligns Freebase relations with the New York Times(NYT) corpus. The training data uses the sentences from year 2005-2006 and the testing instances are generated by the sentences of year 2007. There are 53 relations including NA, which is labeled if two entities have no relationship. UW-dataset is developed by Liu *et al.* [33] and processed by Wu *et al.* [18]. It only includes 5 relations and the number of relations is much smaller than NYT-dataset. Considering that in this paper we utilize the convolutional neural network which needs input of fixed length, both dataset are filtered according to sentence length. Sentences that exceed the specified maximum length will be discarded. The preprocessed statistics are showed in Table 1.

TABLE 1. Dataset statistics.

Dataset	Entity Pair	Mentions	Relational Facts
NYT-Train	288807	525137	16609
NYT-Test	96358	170284	1591
UW-Train	128965	498548	23873
UW-Test	2141	3724	644

To reduce the time-consuming human evaluation, we exploit held-out evaluation as previous work [7], [18], [34] does. Held-out evaluation assumes that the relation facts have the similar performance inside and outside Freebase. So it can provide an approximate estimate. Similarly,the entity pairs are extracted from Freebase and the discovered relational facts from test articles are automatically compared with those in Freebase.

In the experiments we will report the aggregate precision/recall curves, Precision@N(P@N) and AUC. In the PR curve, the samples will be sorted from large to small according to predicted confidence. In this order, the samples are predicted as positive examples one by one and the current recall and accuracy can be calculated at a time. Then the PR curve of the model can be drawn with the precision on Y axis and recall on X axis.The closer the PR curve is to the upper right corner, the better the performance will be. As for P@N, it refers to the precision of the classifier when N positive samples are exactly obtained. And as one of the most widely used evaluation metrics in classification tasks, AUC indicates the probability that the predicted confidence of positive samples is larger than that of negative samples when randomly selecting a positive sample and a negative sample. AUC is expected to be as large as possible.

B. PARAMETER SETTINGS

We employ three-fold validation in the experiments. Table 2 shows the key parameter settings. We select the best parameters using the grid search. Adam [35] is exploited as the optimizer in our experiments. We select the learning rate λ among $\{0.1, 0.01, 0.001, 0.0001\}$, the dimension of position embeddings among $\{3, 5, 10, 15\}$, the sliding window size among $\{1, 2, 3, ..., 6\}$,the weight of generator loss α among $\{1, 10, 20, ..., 200\}$, the batch size among $\{30, 50, 100, 150\}$ and the number of convolution filters among $\{100, 150, 200, 250, 300\}$. Note that in this paper the size of sentence vector is three times the number of convolution filters. Also, we apply dropout [36] to the word

TABLE 2. Parammeter settings.

Learning rate λ	0.001
Weight of generator loss α in NYT-dataset	100
Weight of generator loss α in UW-dataset	10
Position embedding dimension	5
Word embedding dimension	50
Number of convolution filters	250
Batch size	50
Slide window size	3
Dropout probability	0.5
maximum sentence length in NYT dataset	145
maximum sentence length in UW dataset	120

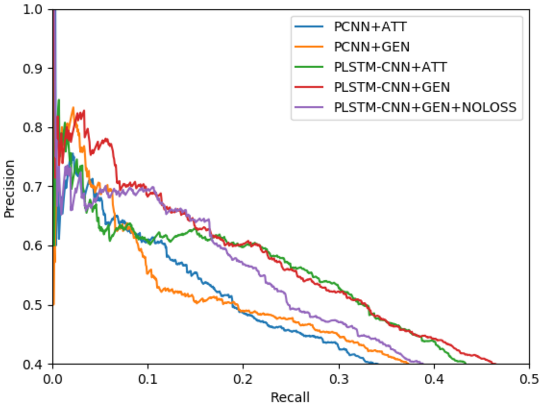

FIGURE 4. Precion/recall curves of a variety of methods on the NYT-dataset.

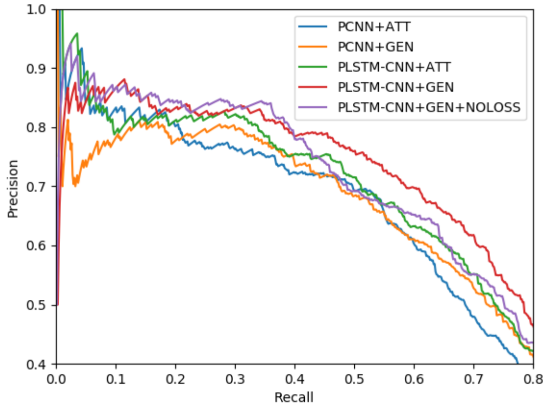

FIGURE 5. Precion/recall curves of a variety of methods on the UW-dataset.

TABLE 3. AUC of various models.

Model	NYT-dataset	UW-dataset
PCNN+ATT	0.301	0.606
PCNN+GEN	0.306	0.610
PLSTM-CNN+ATT	0.352	0.639
PLSTM-CNN+GEN+NOLOSS	0.358	0.646
PLSTM-CNN+GEN	**0.368**	**0.658**

embeddings and the word embeddings are pretrained by Wu *et al.* [18] using word2vec [25].

C. EFFECT OF PLSTM-CNN

We can see from Fig. 4 and Fig. 5 that the introducing of BiLSTM brings better performance because the area enclosed by PR curves of models with PLSTM-CNN is larger. PCNN cuts the sentence semantics into three segments, without considering the continuity. But BiLSTM can learn the forward and backward sequential information between three segments. From Table 3 we can observe that: (1) Comparing PCNN+GEN and PLSTM-CNN+GEN, AUC has been improved by 20.3% and 7.9% in NYT-dataset and UW-dataset respectively. Notably, these numbers are relative rather than absolute improvement. (2) Utilizing BiLSTM significantly improves the performance comparing PCNN+ATT and PLSTM-CNN+ATT. The results indicate that our method is effective and the sentence representation makes a big difference on relation extraction.

Table 4 shows the results of P@100,P@200,P@300 and the mean value in held-out evaluation. We have the following observation: (1) Compared with PCNN+GEN, the mean precision of PLSTM-CNN+GEN is much higher. which even increases by 17.6% in NYT-dataset. We can draw an conclusion that PLSTM-CNN outperforms traditional PCNN. (2) BiLSTM makes the performance drop slower when the confidence predicted declines.Especially, the precision on top-300 has a significant increase when employing BiLSTM, no matter what dataset are used. It implies that PLSTM-CNN is more stable and robust, which has higher tolerance to noises.

D. EFFECT OF GENERATOR

Fig. 4 and Fig. 5 illustrate that our generator performs better than widespread attention mechanism. From Table 3 we can find that: (1) Comparing PCNN+ATT and PCNN+GEN, AUC improves slightly when there is no BiLSTM layer. It indicates that our generator is still effective even though the sentences are encoded by PCNN. (2) When introducing BiLSTM,PLSTM-CNN+GEN significantly outperforms PLSTM-CNN+ATT. It is clear that the generator can work better with the improvement of sentence representation. The reason is that if sentence modeling learns more semantics information, generator will capture more common features from instances for each relation. (3) Compared with traditional PCNN+ATT, our model improves AUC by 22.3% and 8.6% in NYT-dataset and UW-dataset. The results of two different datasets indicate that our proposed method is able to behave well whether the number of relations is large or small.

We can observe from Table 4 that: (1) The introducing of generator also obtains better prediction precision on top-100,top-200 and top-300 instances, in both NYT-dataset and UW-dataset. The result shows that our method can filter out the noises in more efficient way in distant supervision, because the generator is tend to remain the semantic information related to corresponding relation category. Precision of our model on top-100 is obviously higher than other baselines in NYT-dataset,which implies that the positive samples will

TABLE 4. P@N for relation extraction on top 100,200 and 300 in NYT-dataset and UW-dataset.

P@N	NYT-dataset				UW-dataset			
	100	200	300	mean	100	200	300	mean
PCNN+ATT	0.645	0.575	0.509	0.576	0.813	0.763	0.715	0.764
PCNN+GEN	0.69	0.526	0.503	0.573	0.8	0.796	0.719	0.772
PLSTM-CNN+ATT	0.617	0.61	**0.605**	0.611	0.806	0.816	0.742	0.788
PLSTM-CNN+GEN+NOLOSS	0.689	**0.66**	0.575	0.642	**0.854**	0.833	0.723	0.803
PLSTM-CNN+GEN	**0.763**	0.656	0.602	**0.674**	0.847	**0.837**	**0.775**	**0.819**

be given higher confidence by the classifier. (2) The mean precision value of PCNN+GEN is a little smaller than that of PCNN+ATT in the NYT-dataset, but PLSTM-CNN+GEN can obtain much more accurate prediction than PLSTM-CNN+ATT. The result re-emphasizes that the performance of our shared representation generator depends on sentence encoding.

Additionally, from Fig. 4 and Fig. 5, we can find different behavior of the models in two datasets. Fig. 4 shows that PLSTM-CNN+GEN outperforms PLSTM-CNN+ATT. When the recall is lower than 0.16, the precision of our model is obviously higher than that of PLSTM-CNN+ATT. But with the increase of recall, there is almost no difference in precision between two models. The results indicate that the advantage of model using shared representation generator is mainly reflected in the low recall interval. Also, similar findings are found when comparing PCNN+GEN and PCNN+ATT. In PR curves, the lower recall rate means the higher classification confidence threshold. Therefore, in NYT-dataset, our proposed generator can enhance the discrimination ability of samples with high confidence. Positive samples will gain higher confidence in prediction. As for low confidence interval, the difference of performance between the two methods is small. One of the possible reasons is that the test NYT-dataset contains more noises so that both methods are difficult to extract effective semantic information. Text features that express more relation semantics may concentrate upon ''head'' samples, leading to the better discrimination.

However, Fig. 5 shows that the shared representation generator performs exactly the opposite on UW-dataset. Comparing PLSTM-CNN+GEN and PLSTM-CNN+ATT, it is obvious that the model utilizing generator has better performance when the recall is higher. The precision of proposed model is even worse than that of the model using attention mechanism when the recall is lower than 0.05. Because the range of this interval is very small, the model presented in this paper still performs best overall. But the result demonstrates that in UW-dataset the generator focus more on samples with low confidence, which are easily misclassified. Even if the predict confidence is decreasing, the generator can still extract more common features to assist classification, making the confidence of positive samples be higher than that of negative samples.

The different behavior of the model indicates the differences in data distribution and quality between NYT-dataset and UW-dataset. It also indicates that our proposed generator has powerful adaptive and generalization capabilities. In NYT-dataset the generator pays more attention to ''head effect'' but in UW-dataset it emphasizes the ''tail effect''. In summary, constructing bag vector based on shared representation generator is a better choice than using traditional attention mechanism.

E. EFFECT OF GENERATOR LOSS

In this section we will discuss the effect of generator loss. The results of experiments show that: (1) Compared with PLSM-CNN+ATT, we can see from Table 3 that PLSTM-CNN+GEN+NOLOSS can still achieve better performance even though there is no extra loss. Fig. 5 shows that employing the generator without generator loss almost has the best performance when the recall rate is low. So adopting shared representation generator to extract common features can work well. As for traditional attention mechanism, it is essentially a linear weighting process, each instance will preserve part of the original representation and then compute the sum value to obtain bag vector. However, our generator is substantially a process of feature transformation. The original representation will be mapped into the shared representation by the generator of a relation, which express the common features of all sentences in this relation. There is no doubt that feature transformation has stronger non-linear fitting ability than feature weighting, so our model performs better than attention mechanism. (2) The shared representation generator can achieve great improvement on performance when introducing the extra generator loss. PLSTM-CNN+GEN is the best model according to the experimental results. Generator loss is used to measure the generalization of common features. The smaller loss indicates that the transformed sentence vector is closer and meaning of transformed expression is more similar. As a result, this similar semantics can be considered as the common features expressed by all instances. Therefore, the introducing of generator loss can strengthen the ability to extract common features.

Note that we clearly enhance the performance of distantly-supervised relation extraction, only adopting the simple two fully-connected layers. The shared representation generator can exploit other complicated networks. The BiLSTM layer can also be replaced by other sequence modeling methods. Our network is flexible and scalable.

V. CONCLUSION

In this paper, we present a shared representation generator to directly produce the bag vector, which can extract the common features in relation. Our generator implements with two fully connected-layer and we introduce extra generator loss into objective function to enhance the performance of generator. Considering that traditional PCNN will divide the sentence into three segments based on the position of entity pairs, we also introduce BiLSTM to learn more sequential information among three pieces. Our network has good scalability. The experimental results in two datasets show that our proposed method can achieve significant improvement whether the number of relations is large or small compared with baseline systems.

We believe that future research could focus on the introducing of other network to improve our generator such as CNN. We also will explore to utilize generative adversarial network to generate the bag vector.

REFERENCES

[1] A. Madaan, A. Mittal, G. Ramakrishnan, and S. Sarawagi, "Numerical relation extraction with minimal supervision," in *Proc. AAAI*, Feb. 2016, pp. 2764–2771.

[2] G. Li, C. Wu, and K. Vijay-Shanker, "Noise reduction methods for distantly supervised biomedical relation extraction," in *Proc. BioNLP*, 2017, pp. 184–193.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2008, pp. 1247–1250.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a Web of open data," in *Porc. Int. Semantic Web Conf.*, Busan, South Korea, 2007, pp. 722–735.

[5] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. AFNLP*, Aug. 2009, pp. 1003–1011.

[6] S. Riedel, L. Yao, and A. Mccallum, "Modeling relations and their mentions without labeled text," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2010, pp. 148–163.

[7] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 2124–2133.

[8] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[9] P. Liu, X. Qiu, and X. Huang, "Adversarial multi-task learning for text classification," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1–10.

[10] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1753–1762.

[11] A. Graves, *Long Short-Term Memory*. Berlin, Germany: Springer, 2012.

[12] C. Y. Liu, W. B. Sun, W. H. Chao, and W. X. Che, "Convolution neural network for relation extraction," in *Proc. ADMA*, 2013, pp. 231–242.

[13] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. 25th Int. Conf. Comput. Linguistics Tech. Papers*, 2014, pp. 2335–2344.

[14] F. Reichartz, H. Korte, and G. Paass, "Semantic relation extraction with kernels over typed dependency trees," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2010, pp. 773–782.

[15] G. Zhou, M. Zhang, D. H. Ji, and Q. Zhu, "Tree kernel-based relation extraction with context-sensitive structured parse tree information," in *Proc. EMNLP-CoNLL*, 2007, pp. 728–736.

[16] Y. Huang and W. Y. Wang, "Deep residual learning for weakly-supervised relation extraction," in *Proc. EMNLP*, 2017, pp. 1803–1807.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.

[18] Y. Wu, D. Bamman, and S. Russell, "Search adversarial training for relation extraction," in *Proc. EMNLP*, Sep. 2017, pp. 1778–1783.

[19] G. Ji *et al.*, "Distant supervision for relation extraction with sentence-level attention and entity descriptions," in *Proc. AAAI*, Feb. 2017, pp. 3060–3066.

[20] D. Sorokin and I. Gurevych, "Search context-aware representations for knowledge base relation extraction," in *Proc. EMNLP*, Sep. 2017, pp. 1784–1789.

[21] S. Jat, S. Khandelwal, and P. Talukdar. (Apr. 2018). "Improving distantly supervised relation extraction using word and entity based attention." [Online]. Available: https://arxiv.org/abs/1804.06987

[22] J. Xiaotian, Q. Wang, P. Li, and B. Wang, "Relation extraction with multi-instance multi-label convolutional neural networks," in *Proc. 26th Int. Conf. Comput. Linguistics Tech. Papers*, Dec. 2016, pp. 1471–1480.

[23] T. Liu, K. Wang, B. Chang, and Z. Sui, "A soft-label method for noise-tolerant distantly supervised relation extraction," in *Proc. EMNLP*, Sep. 2017, pp. 1790–1795.

[24] M. Zhang, Y. Liu, H. Luan, and M. Sun, "Adversarial training for unsupervised bilingual lexicon induction," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2017, pp. 1959–1970.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (Jan. 2013). "Efficient estimation of word representations in vector space." [Online]. Available: https://arxiv.org/abs/1301.3781

[26] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul./Aug. 2005.

[27] A. Radford, L. Metz, and S. N. Chintala. (Nov. 2015). "Unsupervised representation learning with deep convolutional generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1511.06434

[28] Y. Zhang and L. Carin, "Generating text via adversarial training," in *Proc. NIPS Workshop Adversarial Training*, 2016, p. 21.

[29] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. (Dec. 2015). "Autoencoding beyond pixels using a learned similarity metric." [Online]. Available: https://arxiv.org/abs/1512.09300

[30] B. Luo *et al.*, "Learning with noise: Enhance distantly supervised relation extraction with dynamic transition matrix," in *Proc. ACL*, Jul. 2017, pp. 430–439.

[31] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, "Knowledge-based weak supervision for information extraction of overlapping relations," in *Proc. ACL*, Jun. 2011, pp. 541–550.

[32] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning, "Multi-instance multi-label learning for relation extraction," in *Proc. EMNLP*, Jul. 2012, pp. 455–465.

[33] A. Liu, S. Soderland, J. Bragg, C. H. Lin, X. Ling, and D. S. Weld, "Effective crowd annotation for relation extraction," in *Proc. NAACL*, Jun. 2016, pp. 897–906.

[34] P. Qin, W. Xu, and W. Y. Wang, "DSGAN: Generative adversarial training for distant supervision relation extraction," in *Proc. ACL*, Jul. 2018, pp. 496–505.

[35] D. P. Kingma and J. Ba. (Dec. 2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

DANFENG YAN received the Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications, where she is currently a Professor with the State Key Laboratory of Networking and Switching Technology. Her current research interests include data mining, and big data and analytics.

BO HU is currently pursuing the master's degree with the Beijing University of Posts and Telecommunications, Beijing, China. His major is computer science and technology. His current research interests include machine learning and natural language processing.