Intelligent Data Analysis 24 (2020) 445–457
DOI 10.3233/IDA-194476
IOS Press

Knowledge-embodied attention for distantly supervised relation extraction

Kejun Deng[a], Xuemiao Zhang[b], Songtao Ye[c] and Junfei Liu[d,*]

[a]*School of Electronics Engineering and Computer Science, Peking University, Beijing, China*

[b]*School of Software and Microelectronics, Peking University, Beijing, China*

[c]*The College of Information Engineering, Xiangtan University, Xiangtan, Hunan, China*

[d]*National Engineering Research Center for Software Engineering, Peking University, Beijing, China*

Abstract. Knowledge bases (KBs) provide a large amount of structured information for entities and relations, which are successfully leveraged in many natural language processing tasks. However, distantly supervised relation extraction only utilizes KBs to automatically generate datasets, while ignoring the background information in KBs during the relation extraction process. We herein propose a knowledge-embodied attention that leverages knowledge information in KBs to reduce the impact of noisy data for distantly supervised relation extraction. Specifically, we pre-train distributed representations of KBs with the knowledge representation learning (KRL) model, and subsequently incorporate them into relation extraction to learn sentence-level attention weights. The experimental results demonstrate that our approach outperforms all baselines, thus indicating that we can focus our attention on valid data by leveraging background information in KBs.

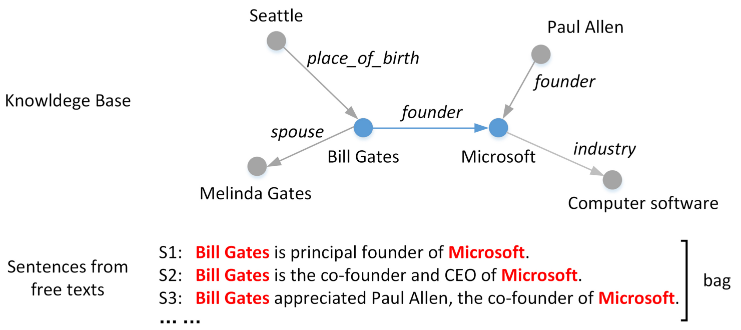
Keywords: Relation extraction, distant supervision, neural networks, sentence-level attention, knowledge representation learning, de-noising

1. Introduction

Extracting semantic relations between entities in text is an important and well-studied task in natural language processing (NLP). Traditional supervised methods [1,2] rely heavily on hand-labeled corpora, which is laborious and expensive. To address the insufficient training data issue, Mintz et al. [3] proposed the distant supervision (DS) paradigm for relation extraction to automatically generate training data via aligning knowledge bases (KBs) and texts. Specifically, for a triplet (e_1, r, e_2) in a KB, all sentences that mention both head entity e_1 and tail entity e_2 constitute a bag, and is labeled with relation r. Figure 1 shows this process.

Recently, neural networks methods have been widely explored on DS relation extraction [4–8]. These methods use all of the words as input without complicated preprocessing. First, words in sentences are transformed to low-dimensional vectors by looking up the word embeddings. Subsequently, sentence-level and bag-level features are learned using convolutional neural networks (CNNs) or recurrent neural networks (RNNs). Finally, the bag-level features are fed into a softmax classifier to predict the relationship between the head and tail entities. Compared with feature-based methods, neural networks methods achieve significant improvements without any features derived from lexical resources or NLP tools.

[*] Corresponding author: Junfei Liu, National Engineering Research Center for Software Engineering, Peking University, Beijing 100871, China. Tel.: +86 13901056728; E-mail: xxjh@pku.edu.cn.



S1: **Bill Gates** is principal founder of **Microsoft**.
S2: **Bill Gates** is the co-founder and CEO of **Microsoft**.
S3: **Bill Gates** appreciated Paul Allen, the co-founder of **Microsoft**.
… …

Fig. 1. Distant Supervision paradigm generates training data automatically by aligning triplets in a KB with free texts. However, some sentences (i.e., S3 in the figure) do not express the relation of the given triplet.

However, most conventional neural networks methods only utilize KBs to generate datasets, where the input of the neural networks is only text or a sequence of words, and numerous background knowledge in KBs is disregarded. The importance of background knowledge in NLP has long been recognized. Earlier NLP studies primarily exploited limited linguistic knowledge such as manually encoded syntactic patterns. With the development of knowledge-based construction, numerous semantic knowledge have become available. Recently, researchers have proposed several improved neural networks to introduce KBs into NLP tasks. Yang et al. [9] proposed KBLSTM , an extension to bidirectional long short-term memory neural networks (BiLSTMs) that can leverage continuous representations of KBs to enhance the learning of RNNs for machine reading. Wang et al. [10] incorporated KB representations into news recommendation by deep knowledge-aware networks.

Inspired by the wide success of leveraging KBs, we attempt to use the background knowledge in KBs to solve the wrong label problem [11] in DS relation extraction. The simple assumptions of distant supervision are not always consistent with the actual situation. There are a lot of wrong-labeling data in the automatically labeled dataset. A sentence that mentions two entities may not express the relation that links them in a KB, i.e., S3 in Fig. 1. Early approaches typically adopted multi-instance learning [12,13] to select the most likely valid sentence from a bag. Neural network-based methods often used the sentence-level attention mechanism [6,14,15] to solve this problem; it attempts to assign weights for instances by the similarity or relatedness between the current input and the target output. However, these sentence-level attentions only use the word embeddings features of the input entity pair or sentences to compute the target relation representation, which is very limited to predict an accurate relation representation.

We have noticed that knowledge representation learning (KRL) models, such as TransE [16] and TransH [17], build entity and relation embeddings by regarding a relation as translation from the head entity to tail entity, i.e., $\mathbf{l}_{e_1} - \mathbf{l}_{e_2} \approx \mathbf{l}_r$ when (e_1, r, e_2) holds in the KB. The difference vector of the head entity and tail entity are often used to predict missing links between the entity pair in the KB. Motivated by these properties of KB representations, we propose a knowledge-embodied attention model based on piecewise convolutional neural networks (denoted by PCNN+KeATT) that combines the neural network model and the KRL model in distantly supervised relation extraction. For a bag, we first use piecewise convolutional neural networks (PCNNs) to extract sentence vectors based on word embeddings, and subsequently automatically learn the weight for each sentence by comparing the sentence vector with the knowledge embeddings computed by $(\mathbf{l}_{e_1} - \mathbf{l}_{e_2})$. Our knowledge-embodied attention can dynamically assign higher weights for valid sentences and lower weights for the invalid ones, because the feature vector of valid sentences should contain a higher similarity (relatedness) with $(\mathbf{l}_{e_1} - \mathbf{l}_{e_2})$.

It is noteworthy that we do not use the knowledge embeddings of the target relation, as directly indicating the relation vector will lead to information leakage during the training process. In addition, we cannot directly calculate the semantic similarity between knowledge embeddings and word embeddings, because they are not learned in the same semantic space. To solve this problem, PCNN+KeATT uses a projection matrix [18] to transfer the knowledge embeddings of entities from the knowledge space into the word space. Finally, we jointly learn the relation extraction task with the projection matrix by adding constraints on the overall objective function. Our experimental results show that our PCNN+KeATT model achieves improvements in relation extraction compared with the state-of-the-art methods.

Our primary contributions in this study can be summarized as follows:

- We proposed an extension to PCNN with a knowledge-embodied attention mechanism that can incorporate KB representations into relation extraction to reduce the impact of noisy data in the DS paradigm.
- We evaluated our PCNN+KeATT model on a widely used dataset and discovered that it outperformed the comparative baselines.
- We further analyzed the effect of different training strategies in detail to explore the influence factors of the model performance. Further, we conducted a detailed analysis on a representative case, thus confirming the power of our attention in selecting valid sentences.

2. Related work

2.1. Distantly supervised relation extraction

Although distant supervision is an effective method to automatically label training data, it inevitably suffers from the wrong-labeling problem. To reduce the impact of noisy data, Riedel et al. [11] regarded distantly supervised relation extraction as a multi-instance single-label problem, where only the most likely sentence for each entity pair in training and prediction was selected. Hoffmann et al. [12] and Surdeanu et al. [13] used a probabilistic graphical model to select valid sentences.

In neural network methods, Zeng et al. [5] explored a PCNN to extract sentence-level features with a multi-instance learning paradigm. The multi-instance learning strategy only selects one sentence with the maximum probability, and does not fully utilize the supervision dataset. Therefore, Lin et al. [6] used a selective attention over multiple instance by generating the inner attentions for each sentence in a bag. Ji et al. [14] proposed a sentence-level attention model based on PCNNs (called APCNN) that directly uses the vector $(\mathbf{e}_1 - \mathbf{e}_2)$ to express the relation in sentences, where $\mathbf{e}$ is the word embedding of entity e. The basic idea of APCNN is from the properties of word embeddings such as $\mathbf{v}(\text{"Madrid"}) - \mathbf{v}(\text{"Spain"}) \approx \mathbf{v}(\text{"Paris"}) - \mathbf{v}(\text{"France"})$.

Our work is inspired by the APCNN. The major difference between PCNN+KeATT and APCNN is that the former attempts to introduce a knowledge representation of entities for a better expression of the relationships between entity pairs, while the latter execute the same task without any knowledge information from the KB. More than half of the entities in the distant dataset (about 40,000) with an occurrence of less than five in the training corpus, thus resulting in a highly sparse feature of their word embeddings. Therefore, introducing the knowledge representation of entities appears to be an effective method to solve this problem. In addition, PCNN+KeATT uses a projection matrix before the attention calculation, and jointly trains the projection matrix with the relation extraction task.

2.2. Knowledge representation learning

We further relate to prior works regarding KRL; the latter aims to project both entities and relations in a KB into a continuous low-dimensional semantic space. Translation-based methods could leverage both effectiveness and efficiency in KRL, and thus have attracted particular attention in recent years. TransE [16] encodes both entities and relations into the same vector space, with relations considered to be translating operations between the head and tail entities. For a triplet (e_1, r, e_2) in a KB, TransE assumes that the distributed representation of tail entity $\mathbf{l}_{e_2}$ should be in the neighborhood of $\mathbf{l}_{e_1} + \mathbf{l}_r$, i.e., $\mathbf{l}_{e_1} + \mathbf{l}_r \approx \mathbf{l}_{e_2}$. The energy function of TransE is as follows: $E(e_1, r, e_2) = ||\mathbf{l}_{e_1} + \mathbf{l}_r - \mathbf{l}_{e_2}||$.

TransE is both effective and efficient, while the simple translating operation may result in conflicts when modeling more complicated relations such as 1-to-N, N-to-1, and N-to-N. To address this issue, TransH [17] proposed a relation-specific hyperplane for translations between entity pairs. TransR [19] directly modeled entities and relations in separate entities and relation spaces.

Our method attempts to introduce knowledge representations into the distantly supervised relation extraction task that uses the TransE model to pre-train the knowledge embeddings of KBs. Compared with other improved KRL models, TransE is relatively simple and is faster for training large-scale knowledge graph. In future work, we will try to combine other KRL models with relation extraction task.

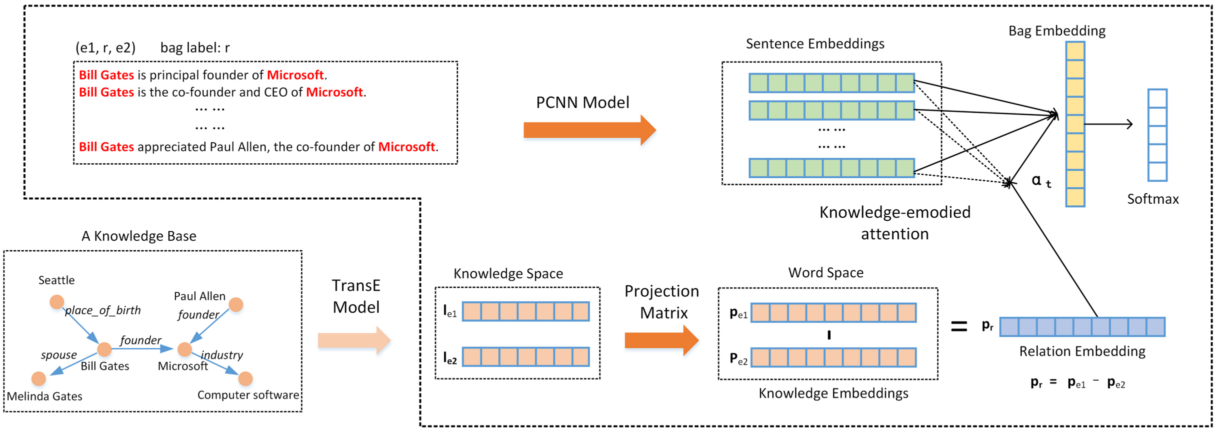
3. Methodology

In this section, we present our PCNN+KeATT model. Figure 2 shows our neural network architecture. The dashed box illustrates the procedure that handles a sentence bag consisting of sentences aligned with (e_1, r, e_2). This procedure consists of four parts: PCNN model, projection matrix, knowledge-embodied attention, and training objective. We describe these parts in detail below. We pre-train the knowledge embeddings on the KB with the TransE model, and use the knowledge embeddings of the entities as the external input information.

3.1. PCNN model

The PCNN model is a widely used neural network in the previous DS relation extraction work [5,6], which is used to extract the feature vector of a sentence in a bag. Suppose that a sentence consisting of t words $\{w_1, w_2, \ldots, w_t\}$ exists; the aim of the PCNN model is to learn the sentence embedding $\mathbf{s}$. The structure of the PCNN model is shown in Fig. 3.

Input vectors: Because we use neural networks, we must transform raw words into low-dimensional vectors as the input. In our method, each word vector is concatenated by word embeddings and position embeddings, similar to [4–6]. Word embeddings are distributed representations of words that map each word in texts to a low-dimensional vector. We employ the word2vec model [20] to pre-train the word embeddings. Position embeddings are defined as the combination of the relative distance from the current word to e_1 and e_2 in a sentence. For example, in the sentence "*Bill Gates* is the principle founder of *Microsoft*," the relative distance from the word "founder" to the head entity *Bill Gates* and tail entity *Microsoft* are 3 and 2, respectively. In our model, the relative distances are mapped to vectors. For word "founder", two distance vectors of length 3 and 2 are randomly initialized. Then, we combine the word embedding and two position embeddings as the input vector for each word. For the *i*-th word in the sentence, $\mathbf{w}_i$ indicates the corresponding word embedding, and $\mathbf{w}'_i$ indicates the input vectors for the PCNN consisting of word embeddings and position embeddings. Assume that the size of word



Bill Gates is principal founder of Microsoft.
Bill Gates is the co-founder and CEO of Microsoft.
Bill Gates appreciated Paul Allen, the co-founder of Microsoft.

Fig. 2. Neural networks architecture for PCNN+KeATT model. It comprises four parts: PCNN model, projection matrix, knowledge-embodied attention, and training objective. PCNN mode is used to learn sentence embeddings from word embeddings. Projection matrix transfers the knowledge embeddings of the head and tail entities from knowledge space into word space. Knowledge-embodied attention computes the bag embedding by assigning attention weights for each sentence, and subsequently feeds the bag embedding into a softmax. Training objective jointly learns DS relation extraction task with the projection matrix.

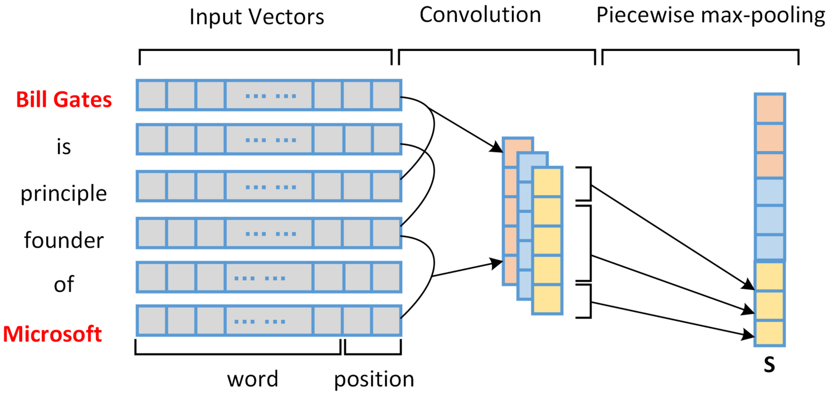


Fig. 3. Structure of PCNN model. Input a sentence with $\{w_1, w_2, \dots, w_t\}$; PCNN uses convolutions and piecewise max-pooling to extract the sentence embedding $\mathbf{s}$.

embedding is k_w and that of the position embedding is k_d; subsequently, the size of the input vector is $k = k_w + 2k_d$.

Convolution Layer: The convolution of $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{B} \in \mathbb{R}^{m\times n}$ is defined as

$$\mathbf{A}\otimes\mathbf{B} = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}b_{ij} \tag{1}$$

For the input sentence $S = \{\mathbf{w}'_1, \mathbf{w}'_2, \dots, \mathbf{w}'_t\}$, where $\mathbf{w}'_i$ is the input vector of the i-th word, we use $\mathbf{S}_{i:j}$ to represent the matrix concatenated by sequence $[\mathbf{w}'_i; \mathbf{w}'_{i+1}; \dots; \mathbf{w}'_j]$. Suppose that the length of the sliding window is l (Fig. 3 shows an example of $l = 3$); subsequently, the weight matrix of the filters in the convolution layer is $\mathbf{W} \in \mathbb{R}^{l\times k}$. Subsequently, the convolution operation between the filter $\mathbf{W}$ and the sentence S results in a vector $\mathbf{c} \in \mathbb{R}^{t-l+1}$:

$$c_j = \mathbf{W}\otimes\mathbf{S}_{(j-l+1):j} + \mathbf{b}, \tag{2}$$

where $1\leqslant j\leqslant t-l+1$, and $\mathbf{b}$ is a bias.

For sentence S, we use n filters $\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_n$ in the convolution layer (**Figure** 3 shows an example of $n = 3$), and obtain the results vectors $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n\}$.

Piecewise max-pooling: The size of the convolution output matrix $\mathbf{C}$ depends on the number of words in the sentence. To obtain the length-fixed feature embeddings for each sentence, max-pooling operations are often applied for this purpose, where the most significant features (with highest values) in each feature map are chosen.

Furthermore, to capture the fine-grained features and structure information, the PCNN divides a sentence into three segments according to the given entity pair, and subsequently perform max-pooling on each segment. For each filter result $\mathbf{c}_i$, the piecewise max-pooling procedure can be expressed as follows:

$$p_{ij} = \mathit{max}(\mathbf{c}_{ij}) 1\leqslant i\leqslant n, 1\leqslant j\leqslant 3 \tag{3}$$

where n is the number of filters in the convolution layer. Therefore, we can concatenate all the vectors: $\mathbf{p}_i = [p_{i,1}, p_{i,2}, p_{i,3}](i = 1, 2, \ldots, n)$ to obtain vector $\mathbf{p} \in \mathbb{R}^{3n}$, as shown in Fig. 3. Finally, we calculate the sentence embedding $\mathbf{s} = \mathit{tanh}(\mathbf{p})$ for sentence S, where $\mathbf{s} \in \mathbb{R}^{3n}$.

3.2. Projection matrix

As shown in Fig. 2, we use the KB as the external information resource, and use the TransE model to pre-train the knowledge embeddings for all entities and relations in the KB. That is, for a triplet (e_1, r, e_2), we transfer them into vectors $\mathbf{l}_{e_1}, \mathbf{l}_r, \mathbf{l}_{e_2}$. Based on the TransE assumption, their embeddings confirm that $\mathbf{l}_{e_2} - \mathbf{l}_{e_1} \approx \mathbf{l}_r$. We denote the size of the TransE embeddings as k_e.

To introduce the knowledge embeddings of the entities into the relation extraction task, we must first transfer the knowledge embeddings from the knowledge space to the word space. Inspired by [18], we transfer the entity embeddings from the knowledge space into the word space with a shared projection matrix. The knowledge embeddings $\mathbf{p}_{e_1}$ and $\mathbf{p}_{e_2}$ in the word space for the head e_1 and the tail entity e_2 are defined as follows:

$$\mathbf{p}_{e_1} = \mathbf{M}_p \cdot \mathbf{l}_{e_1} \tag{4}$$

$$\mathbf{p}_{e_2} = \mathbf{M}_p \cdot \mathbf{l}_{e_2} \tag{5}$$

in which $\mathbf{M}_p \in \mathbb{R}^{k_w \times k_e}$, and the size of $\mathbf{p}$ is k_w the same as that of the word embeddings.

3.3. Knowledge-embodied attention

We expect that our attention mechanism could reduce the impact of noisy data by using information in the KB; this implies that our attention must learn higher weights for valid sentences and lower weights for invalid ones in the bag. Once the bag embeddings have been calculated, we feed them into a softmax classifier.

Attention Mechanism: The simple assumption of distantly supervision inevitably accompanies with the wrong-labeling problem. To address this issue, we require an effective attention mechanism over sentences in a bag that is expected to dynamically reduce the weights of those noisy sentences.

As stated in Section 3.2, the TransE model entities and relations in a KB in the same vector space, with relations considered to translate between the head and tail entities. Hence, for a triplet (e_1, r, e_2) in the KB, the knowledge embeddings in the knowledge space conform with $\mathbf{l}_{e_2} - \mathbf{l}_{e_1} \approx \mathbf{l}_r$.

Because we use a shared matrix to project the knowledge embeddings of the head entity e_1 and tail entity e_2 from the knowledge space into the word space, the difference between the projected knowledge embeddings should still express the relationship r to some extent, or it can be considered as a collection of multiple relationships (because e_1 and e_2 may have multiple relations in the KB).

Motived by these ideas, we utilize the difference vector to represent the feature of the relation r between e_1 and e_2. Specifically, for a bag aligned by (e_1, r, e_2), we introduce a relation embedding $\mathbf{p}_r = \mathbf{p}_{e_2} - \mathbf{p}_{e_1}$ to represent the features of relation r. Each sentence in the bag may express the relation r or not. If a sentence is valid, its sentence embedding should exhibit a higher similarity or relatedness with relation embedding $\mathbf{p}_r$.

Suppose m sentences exist in the bag, and $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m\}$ are sentence embeddings. We use the general score (similarity or relatedness) function in [21] to compute the attention weight between each sentence embedding and relation embedding, and propose the following formulas:

$$\alpha_i = \frac{exp(\omega_i)}{\sum_{j=1}^{m} exp(\omega_j)} \tag{6}$$

$$\omega_i = score(\mathbf{s}_i, \mathbf{p}_r) = \mathbf{s}_i^{\top} \mathbf{M}_a \mathbf{p}_r + d_a \tag{7}$$

where $\mathbf{M}_a \in \mathbb{R}^{s \times k_w}$ is an intermediate matrix and $\mathbf{d}_a$ is an offset value. Subsequently, the bag embedding can be computed by $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_m]$ as follows:

$$\mathbf{b} = \sum_{i=1}^{m} \alpha_i \mathbf{s}_i \tag{8}$$

where $\mathbf{b} \in \mathbb{R}^{3n}$, the same size of sentence embeddings.

Softmax: Finally, we feed the sentence embedding into a softmax classifier as follows:

$$\mathbf{o} = \mathbf{M}_s \mathbf{b} + d_s \tag{9}$$

where $\mathbf{o} \in \mathbb{R}^{n_r}$ is the final output; n_r is the total number of relations. Let θ indicate all the parameters in our model; for bag B, the conditional probability of the *i*-th relation is

$$p(r_i|B;\theta) = \frac{exp(o_i)}{\sum_{j=1}^{n_r} exp(o_j)} \tag{10}$$

3.4. Training objective

For each entity e_i, we learn two types of semantic embeddings in our task. The first one is the word embedding $\mathbf{w}_{e_i}$ learned by word2vec, and the other one is projected knowledge embedding $\mathbf{p}_{e_i}$ learned by TransE and the projection matrix. In our method, we train the projection matrix by placing these two embeddings of entities close to each other. Therefore, we define an extra energy function as follows:

$$\mathcal{L}_e = \sum_{i=1}^{|E|} ||\mathbf{w}_i - \mathbf{p}_i|| \tag{11}$$

where $|E|$ is the number of all entities.

Assume that N bags in exist in the training dataset $\{B_1, B_2, \ldots, B_N\}$ in the distant supervision dataset, and their corresponding labels are $\{r_1, r_2, \ldots, r_N\}$. We use θ to indicate all parameters, and the objective function of the DS relation extraction is defined as follows, which is the same as that of previous work [5,6]:

$$min\mathcal{L}_D = \sum_{i=1}^{N} log(r_i|B_i;\theta) \tag{12}$$

To learn the relation extraction task using the projection matrix, the overall objective function of our method is

$$min\mathcal{L} = \mathcal{L}_D + \lambda\mathcal{L}_e \tag{13}$$

where $\lambda > 0$ is the weight of $\mathcal{L}_e$.

4. Experiment

The experiments are intended to prove that our PCNN+KeATT model can successfully use the semantic information in KBs to reduce the impact of noisy data on distant supervision. In this section, we first introduce the dataset and settings. Subsequently, we present the experimental results and analysis.

4.1. Dataset and evaluation metrics

In this section, we introduce our dataset and evaluation metrics.

Dataset: We evaluate our method on a widely used dataset developed by [11]. This dataset is generated by aligning the Freebase triplets with the New York Times corpus (NYT). The entity mentions are annotated with the Stanford NER and further linked to Freebase. The training dataset is aligned to the 2005–2006 NYT corpus, and the testing dataset is aligned to the 2007 NYT corpus. The dataset contains 53 possible relationships including the NA relation, which indicates that no relation exists between the entity pairs.

The training dataset contains 570,088 sentences, 63,428 entities, 291,010 entity pairs, and 19,601 relational triplets (except NA). The testing set contains 172,448 sentences, 16,705 entities, 96,678 entity pairs, and 1,950 relational triplets (except NA). We train the word embeddings with word2vec[1] and maintain the words that appear more than 100 times in the corpus as the word vocabulary and initial values, similar to [6].

Our methods use KBs as the external information by modeling entities and relations in a KB into a low-dimensional vector space. To capture richer semantic information, we combine all training triplets related to DS relation extraction with a subset of Freebase (called FB15K) as a new KB to train the knowledge embeddings. To prevent overfitting, we remove the triplets in the test set from FB15K. Finally, our KB dataset contains 78,639 entities, 1,371 relations, and 1,225,678 relation triplets. We pre-train the knowledge embeddings with KB2E,[2] an open tool for the TransE model.

Evaluation metrics: Following the previous work [3], we evaluate our method in two ways: the held-out evaluation and manual evaluation. The held-out evaluation automatically compares the extracted relation triplets against those in Freebase, and reports the result in precision/recall curves. Considering the incomplete nature of Freebase, we use human evaluation to manually verify the newly discovered relation triplets that are not in Freebase. We report the Precision@N for the manual evaluation.

4.2. Experimental settings

In our experiments, we use three-fold validation to tune all of the models on the training set, and use a grid search to determine the optimal parameters. We select the dimension of word embedding k_w and

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

[2] https://github.com/thunlp/KB2E.

knowledge embeddings k_e among $\{50, 150, 250\}$, the dimension of position embeddings k_d among {5, 10, 20}, the window size l among {3, 5, 7}, the number of filters n among $\{100, 150, 200, 230\}$, the weight λ among {0.01, 0.1, 1.0, 1.2}, the batch size among {40, 160, 640, 1280}, and the learning rate among {0.01, 0.001, 0.0001, 0.00001}. The best configurations are as follows: $k_w = 50$, $k_e = 50$, $k_d = 5$, $l = 3$, $n = 230$, $\lambda = 0.1$, the batch size is 160, and the learning rate is 0.01.

It is noteworthy that we use the TensorFlow[3] deep learning framework to implement the PCNN+KeATT model. In this specific implementation, we used a compromise technique that set a suitable bag size to fit the framework's constraints of fixed-dimensional input, because the number of sentences in each bag may be different. After statistics analysis, we found the bag size of 12 to be a good choice. We split the big bag into multiple bags when the bag size is larger than 12, and fill the small bag by resampling when the size is smaller than 12.

4.3. Comparison with traditional methods

To evaluate our PCNN+KeATT model, we select the following three feature-based methods and three neural network methods for comparison. **Mintz**: [3] represented a traditional distantly supervised model. **Hoffmann**: [12] is a multi-instance learning method. **MIMLRE**: [13] is a multi-instance and multi-relation model. **PCNN+ONE**: [5] used the PCNN to extract the feature vector for sentences and only selected the most likely valid sentence for each bag. **PCNN+ATT**: [6] used an inner selective attention to select informative sentences based on the PCNN model. **APCNN**: [14] proposed a sentence-level attention motived by the properties of word embeddings.

Held-out evaluation: Figure 4 shows the precision/recall curves for each method, where our PCNN+KeATT outperforms all traditional models over the entire range of recall. As PCNN+KeATT achieves better performance than all baselines, we conclude that our knowledge-embodied attention can effectively reduce the impact of noisy data by assigning lower attention weights for invalid sentences. In particular, PCNN+KeATT outperforms PCNN+ATT (using an inner sentence-level attention) and APCNN (using word-embedding-based attention), thus demonstrating that our attention can better focus on valid sentences by utilizing external information in KBs.

Manual evaluation: As shown in Fig. 4, a sharp decline occurs in the held-out precision/recall curves of most models at low recalls; this is due to the incomplete nature of Freebase. A manual verification for the misclassified instances with high confidence can eliminate the problems [4,6,11,14]. Table 1 shows the manual evaluation on the top-100, top-200, and top-500 extracted relation instances. The results show that PCNN+KeATT achieves the best performance and exhibits significant improvements over the traditional methods especially in the top-500 instances. The evaluation results have proven the effectiveness of our method.

4.4. Further discussion

As stated in Section 3, we use the general score function [21] to compute the attention weight for each sentence in a bag. It is noteworthy that two score functions are widely used [21]. Hence, for two vectors $\mathbf{v}$ and $\mathbf{w}$, their similarity or relatedness score could be calculated as follows:

$$\mathit{score}(\mathbf{u},\mathbf{v}) = \begin{cases} \mathbf{u}^\top \mathbf{W}_g \mathbf{v} & \mathit{general} \\ \mathbf{W}_c^\top (\mathit{tanh}[u; v]) & \mathit{concat} \end{cases}$$

[3] https://www.tensorflow.org/.

Table 1
Precision for the top 100, 200, 500 extracted relations

	Top 100	Top 200	Top 500	Average
Mintz	0.77	0.71	0.55	0.676
Hoffmann	0.83	0.74	0.59	0.720
MIMLRE	0.85	0.75	0.61	0.737
PCNN+ONE	0.84	0.77	0.64	0.750
PCNN+ATT	0.86	0.80	0.68	0.780
APCNN	0.87	0.82	0.72	0.802
PCNN+KeATT	**0.90**	**0.84**	**0.73**	**0.820**

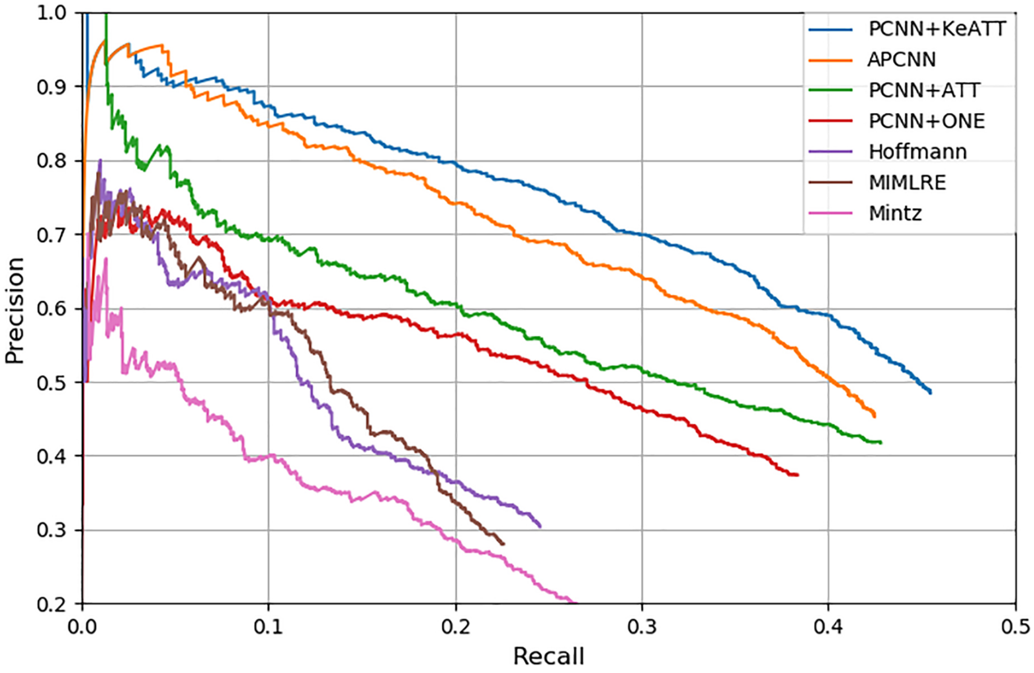


Fig. 4. The precision/recall curves of PCNN+KeATT and traditional models.

where $[\mathbf{u}; \mathbf{v}]$ denotes the vertical concatenation of $\mathbf{u}$ and $\mathbf{v}$; $\mathbf{W_g}$ and $\mathbf{W_c}$ are intermediate matrices.

In addition, we jointly learn the projection matrix with the relation extraction task. In fact, we can learn the projection matrix separately, and subsequently use the projected knowledge embeddings on the relation extraction task.

In this section, we further explore the effect of training strategies by conducting four experiments; Fig. 5 shows the precision/recall curves. As shown: (1) The performance of joint learning modes based on two score functions are better than that of separate learning models, demonstrating that the reverse transmission of loss from the space projection model in the joint learning methods can further feed the knowledge information to the master model. (2) Using the general score function, the performance of the DS relation extraction model has been improved. This shows that in our task, the general function can evaluate the similarity or relatedness between two vectors better than the concatenation function.

Table 2

An example of attention weights by KeATT. The bag is aligned by /location/location/contains (Ukraine, Kiev)

Bag label: /location/location/contains (Ukraine, Kiev)		
Sentences	KeATT	Manual
S1: "If there is a president in the country, and if the president loves **Ukraine** and respects Ukrainians, he will either dissolve the Parliament or lose the remaining support of society," Yulia V. Tymoshenko, once Mr. Yushchenko's prime minister, said Friday, the Russian News Agency Interfax reported from **Kiev**.	0.00001	Invalid
S2: Directed by Andrei Zagdansky Not rated; 72 minutes in the documentary "Orange Winter Orange" blooms throughout **Kiev**, **Ukraine**, the epicenter of dissent over that country's stolen 2004 presidential elections.	**0.11200**	Valid
S3: Miss America was born Leah Berliawsky in **Kiev**, **Ukraine**, to upper-middle-class Jews with ties to the lumber industry.	**0.31160**	Valid
S4: **Ukraine** Ukraine's crisis-defused battling political rivals agreed to hold new parliamentary elections, defusing a political crisis that had escalated with president Viktor A. Yushchenko's decision to order extra interior ministry troops to the capital, **Kiev**.	0.00003	Invalid
S5: Born on Sept.27, 1907, in **Kiev**, **Ukraine**, Mr.Maslow was the son of Raeesa and Saul Maslenkov.	**0.57131**	Valid
S6: Aleksey Kolupaev, 25, works for an internet company in **Kiev**, **Ukraine**, and in his spare time, with his friend Juriy Ogijenko, he develops and sells software that can thwart captchas by analyzing the images and separating the letters and numbers from the background noise.	**0.00401**	Valid
... ...		

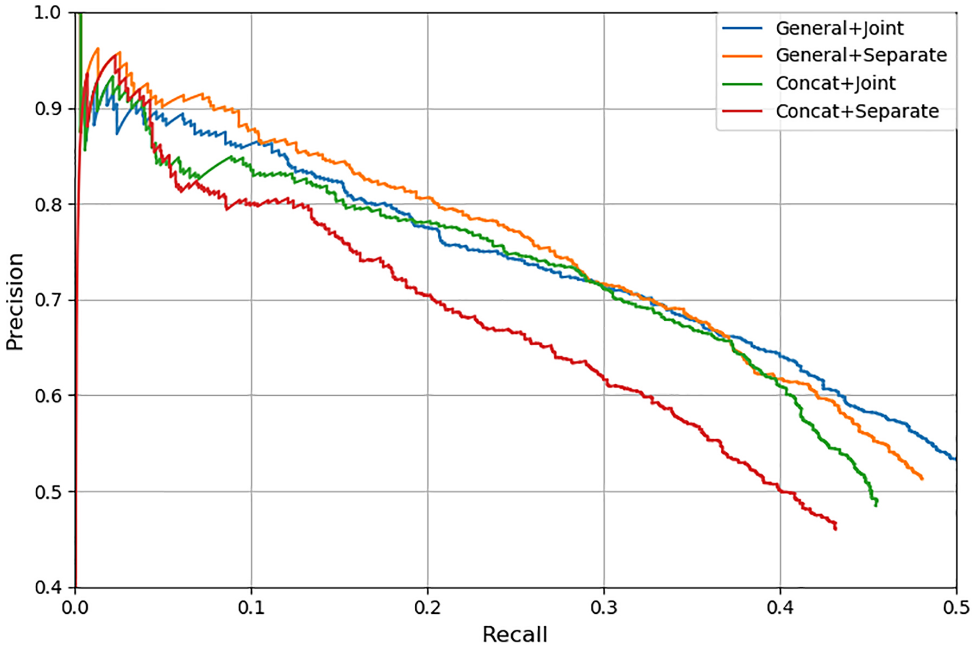


Fig. 5. Precision/recall curves of PCNN+KeATT with different training strategies. General and concat indicate two types of score functions. Joint and separate represent different learning methods for the projection matrix.

4.5. Case study

Table 2 shows a representative example of knowledge-embodied attention weights from the testing data. The bag contains 12 sentences, with four valid ones according to manual assessment. Table 2 shows that our attention assigns higher weights to valid sentences and lower weights to invalid ones.

It demonstrates that our attention can reduce the impact of noisy data. For valid sentences, we found that longer sentences are assigned with lower attention weights, although they do not exhibit obvious semantic differences with shorter sentences. (i.e., the weight of S6 is much lower than the other valid sentences, but still significantly higher than the invalid ones.) This shows that the PCNN model can still be improved when handling long sentences.

5. Conclusion and future work

We herein proposed the knowledge-embodied attention for distantly supervised relation extraction, using knowledge information from KBs to reduce the impact of noisy data. Our experimental results proved the effectiveness of our method.

We will explore the following in the future: (1) The quality of knowledge embeddings is critical in our model. We will utilize more sophisticated KRL models, such as TransR or TransH, to extract better knowledge features. (2) Our method is based on the PCNN that has shown its weak ability to handle long sentences in our case study. We will explore other sentence representation models in the future.

Acknowledgments

This research is supported by the National Natural Science Funds No. 61802327 and Natural Science Foundation of Hunan Province No. 2018JJ3511.

References

[1] M. Zhang, J. Zhang, J. Su and G. Zhou, A composite kernel to extract relations between entities with both flat and structured features, in: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Association for Computational Linguistics*, 2006, pp. 825–832.

[2] N. Bach and S. Badaskar, A review of relation extraction, Literature review for Language and Statistics II 2.

[3] M. Mintz, S. Bills, R. Snow and D. Jurafsky, Distant supervision for relation extraction without labeled data, in: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, Association for Computational Linguistics, 2009, pp. 1003–1011.

[4] D. Zeng, K. Liu, S. Lai, G. Zhou and J. Zhao, Relation classification via convolutional deep neural network, in: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 2335–2344.

[5] D. Zeng, K. Liu, Y. Chen and J. Zhao, Distant supervision for relation extraction via piecewise convolutional neural networks, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1753–1762.

[6] Y. Lin, S. Shen, Z. Liu, H. Luan and M. Sun, Neural relation extraction with selective attention over instances, in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1, 2016, pp. 2124–2133.

[7] N.T. Vu, H. Adel, P. Gupta and H. Schütze, Combining Recurrent and Convolutional Neural Networks for Relation Classification, HLT-NAACL 2016, pp. 534–539.

[8] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao and B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vol. 2, 2016, pp. 207–212.

[9] B. Yang and T. Mitchell, Leveraging knowledge bases in lstms for improving machine reading, in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1, 2017, pp. 1436–1446.

[10] H. Wang, F. Zhang, X. Xie and M. Guo, Dkn: Deep knowledge-aware network for news recommendation, arXiv preprint arXiv:180108284.

[11] S. Riedel, L. Yao and A. McCallum, Modeling relations and their mentions without labeled text, in: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2010, pp. 148–163.

[12] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer and D.S. Weld, Knowledgebased weak supervision for information extraction of overlapping relations, in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, Association for Computational Linguistics, 2011, pp. 541–550.

[13] M. Surdeanu, J. Tibshirani, R. Nallapati and C.D. Manning, Multi-instance multi-label learning for relation extraction, in: *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, Association for Computational Linguistics, 2012, pp. 455–465.

[14] G. Ji, K. Liu, S. He, J. Zhao et al., Distant supervision for relation extraction with sentence-level attention and entity descriptions, in: AAAI, 2017, pp. 3060–3066.

[15] X. Feng, J. Guo, B. Qin, T. Liu and Y. Liu, Effective deep memory networks for distant supervised relation extraction, in: *Proceedings of the TwentySixth International Joint Conference on Artificial Intelligence*, IJCAI, 2017, pp. 19–25.

[16] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: *Advances in neural information processing systems*, 2013, pp. 2787–2795.

[17] Z. Wang, J. Zhang, J. Feng and Z. Chen, Knowledge graph and text jointly embedding, in: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1591–1601.

[18] R. Xie, Z. Liu, H. Luan and M. Sun, Image-embodied Knowledge Representation Learning, IJCAI, 2017, pp. 3140–3146.

[19] Y. Lin, Z. Liu, M. Sun, Y. Liu and X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: AAAI, Vol. 15, 2015, pp. 2181–2187.

[20] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, in: *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[21] M.-T. Luong, H. Pham and C.D. Manning, Effective approaches to attention-based neural machine translation, arXiv preprint arXiv:150804025.