REKER: Relation Extraction with Knowledge of Entity and Relation

Hongtao Liu[1], Yian Wang[2], Fangzhao Wu[3], Pengfei Jiao[4]✉, Hongyan Xu[1], and Xing Xie[3]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
{htliu, hongyanxu}@tju.edu.cn
[2] School of Physics, University of Science and Technology of China, Hefei, China
wya16@mail.ustc.edu.cn
[3] Microsoft Research Asia, Beijing, China
wufangzhao@gmail.com, xingx@microsoft.com
[4] Center for Biosafety Research and Strategy, Tianjin University, Tianjin, China
pjiao@tju.edu.cn

Abstract. Relation Extraction (RE) is an important task to mine knowledge from massive text corpus. Existing relation extraction methods usually purely rely on the textual information of sentences to predict the relations between entities. The useful knowledge of entity and relation is not fully exploited. In fact, off-the-shelf knowledge bases can provide rich information of entities and relations, such as the concepts of entities and the semantic descriptions of relations, which have the potential to enhance the performance of relation extraction. In this paper, we propose a neural relation extraction approach with the knowledge of entity and relation (REKER) which can incorporate the useful knowledge of entity and relation into relation extraction. Specifically, we propose to learn the concept embeddings of entities and use them to enhance the representation of sentences. In addition, instead of treating relation labels as meaningless one-hot vectors, we propose to learn the semantic embeddings of relations from the textual descriptions of relations and apply them to regularize the learning of relation classification model in our neural relation extraction approach. Extensive experiments are conducted and the results validate that our approach can effectively improve the performance of relation extraction and outperform many competitive baseline methods.

Keywords: relation extraction, entity concept, relation description

1 Introduction

Relation extraction (RE) aims to identify semantic relations between known entities from plain text corpus [10]. For example, as shown in Fig.1, given a sentence "Steve Jobs was the co-founder and CEO of Apple" and two entities "Steve Jobs" and "Apple" in it, the goal of relation extraction is to identify that there is

a `Co-Founder` relation between the person entity `Steve_Jobs` and the company entity `Apple`. Relation extraction is an important task in information extraction field, and is widely used in many real-world applications such as knowledge base completion [13], question answering [17] and so on.

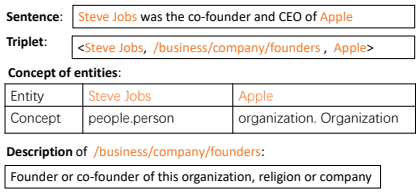


Fig. 1. An illustrative example of relation extraction with the knowledge of both entity (i.e., the concepts of entities) and relation (i.e., the description of relation).

Many methods have been proposed for relation extraction. Early methods for relation extraction mainly depended on human-designed lexical and syntactic features, e.g., POS tags, shortest dependency path and so on [2, 3]. Designing these handcrafted features relies on a large number of domain knowledge [19]. In addition, the inevitable errors brought by NLP tools may hurt the performance of relation extraction [8]. In recent years, many neural network based methods have been proposed for relation extraction [20, 19, 8]. For example, Zeng et al. [19] utilized Piece-Wise Convolutional Neural Network (PCNN) to encode sentences and adopt multi-instance learning to select the most informative sentences for an entity pair. Lin et al. [8] built a sentence-level attention model to make use of the information of all sentences about an entity pair. These methods can automatically extract semantic features of sentences with various neural networks and then feed them into a relation classifier to predict the relation labels of entity pairs in these sentences.

However, most previous works merely consider the relation extraction as a sentence classification task and focus on the text encoder but ignore the off-the-shelf and rich knowledge about entities and relations in knowledge bases. As shown in Fig.1, entity concepts, also known as the categories of entities, can intuitively provide extra guidance information for the classifier to identify the relation between two entities [11]. For example, the relation between an entity of `Person` and an entity of `Organization` would be related with positions (e.g., `business.company.founders` between "Steve Jobs" and "Apple"). As for knowledge about relations, most existing works merely regard relations as labels in classification, i.e., meaningless one-hot vectors, which would cause a loss of information. In fact, relations contain rich potential semantic information such as their textual descriptions. Considering that human beings can easily understand the meaning of a relation according to its description, we should attach the semantic knowledge from descriptions into relations and integrate it to relation extraction models rather than regarding relations merely as meaningless labels.

To address these issues, in this paper we propose a novel approach that can integrate two additional information (i.e., entity concept and relation description) into the neural relation extraction model named REKER: Relation Extraction with Knowledge of Entity and Relation. First, we encode entity concepts into feature vectors in an embedding way which can capture the context semantic of concepts. Afterwards, we integrate these embedding vectors into the sentence encoder directly to enhance text representation. Second, in order to attach semantics into relation labels, we extract relation textual descriptions from knowledge bases and learn the semantic feature vectors of relations. Then we apply the semantic representation in the relation classification part and regard it as a regular item in loss function. In this way, we can introduce the information of relation descriptions into the relation extraction model.

The major contributions of this paper are summarized as follows:

(1) We propose an effective neural approach to integrate additional knowledge including concepts of entities and textual descriptions of relations to enhance neural relation extraction.

(2) We utilize entity concept embedding to capture the semantic information of entity concepts, and we encode the relation description to attach semantics into relation labels rather than regarding them as meaningless one-hot vectors.

(3) We conduct extensive experiments on two relation extraction benchmark datasets and the results show our approach can outperform the existing competitive methods.

2 Related Work

Relation extraction is one of the most important tasks in NLP and many works have been proposed so far. Traditional approaches depend on the designed features and regard relation extraction as a multi-class classification problem. Kambhatla et al. [7] designed handcrafted features of sentences and then feed them into a classifier (e.g., SVM). The supervised approaches may suffer from the lack of labeled training data. To address this problem, Mintz [10] proposed a distant supervision method which align text corpus to the given knowledge base and automatically label training data. However, distant supervision would result in wrong labeling problem. Hoffmann et al. [4] relaxed the strong assumption of distance supervision and adopted multi-instance learning to reduce noisy data.

Recently with the development of deep learning, many works in RE based on the neural network have achieved a significantly better performance compared with traditional methods. Zeng et al. [20] adopted a Convolutional Neural Network to extract semantic features of sentences automatically. In addition, to reduce the impact of noisy data in distant supervision, Zeng et al. [19] combined the multi-instance learning with a convolutional neural network encoder, Lin et al. [8] adopted a sentence-level attention over multiple instances to pay attention to those important sentences.

Besides, a few works utilize the knowledge about entities or relations in relation extraction. Vashishth et al. [15] employs Graph Convolution Networks for

encoding syntactic information of sentences and utilize the side information of entity type and relation alias. However, the information used in [15] may be limited. For example the alias of a relation is always too short to represent the relation semantic. To utilize the rich knowledge of both entity and relation more effectively, we propose a flexible approach that integrates the concepts of entities and descriptions of relations into neural relation extraction models.

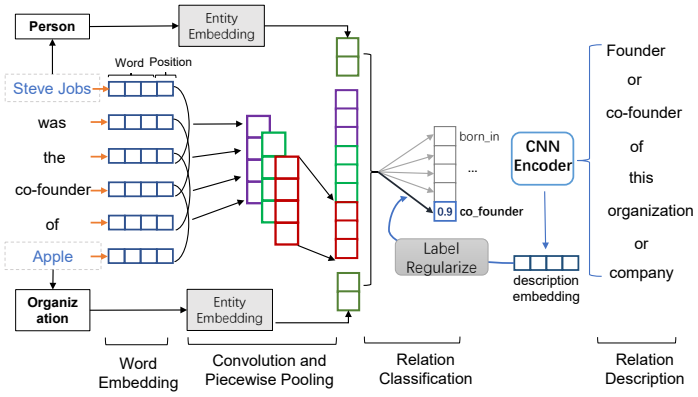


Fig. 2. The architecture of our REKER model.

3 Proposed Method

This section describes our proposed approach. We first give the problem definition and then introduce our approach REKER in detail as shown in Fig. 2. Given a sentence s_i and the mentioned entity pair (e_1, e_2) in s_i, the goal of relation extraction is to predict the relation r_j existing between e_1 and e_2 . Our model estimates the probability $p(r_j|s_i, e_1, e_2)$, $j = 1, 2, 3 \cdots N_r$, where N_r denotes the number of relations.

In this paper, we focus on integrating additional knowledge about entity and relation. As a result, our method can be used both in fully supervised learning and distant supervised learning (i.e., sentence-level and bag-level relation extraction tasks). We will next present our approach in sentence-level relation extraction, which is almost the same as in bag-level tasks.

3.1 Sentence Representation Module

As shown in the overview, we adopt a Convolutional Neural Network (CNN) to encode a sentence into a vector. This module includes three parts: word representation layer, convolution and pooling layer and the entity concept enhancement part.

Word Representation We transform the words in sentences into distributed representation including word embedding and position embedding which captures the position information.

Word Embedding Given a sentence $s = \{w_1, w_2, \cdots, w_n\}$, each word w_i in s is represented as a low-dimensional real-valued vector $\mathbf{w_i} \in \mathcal{R}^{d_w}$, where d_w is the dimension.

Position Embedding Zeng et al. [20] proposed the Position Feature (PF) of words to specify distance from words to entity mentions. The idea is that words closer to the two entity words are more informative. PF is the combination of the relative distance of the current word to `entity1` and `entity2`. The distance is then encoded into low-dimensional vector as position embedding.

Then a word w_i can be represented by concatenating its word embedding and two position embeddings, denoted as $\mathbf{w_i} \in \mathcal{R}^{(d_w + d_p \times 2)}$, where d_p is the dimension of the position embedding vector. Hence, the final word embedding of sentence $s = \{w_1, w_2, \cdots, w_n\}$ can be expressed as a matrix: $\mathbf{X} = [\mathbf{w_1}, \mathbf{w_2}, \cdots, \mathbf{w_n}]$.

Convolution and Pooling We apply a convolutional layer into the sentence matrix $\mathbf{X}$ to extract senmatic features of sentence. Then a pooling layer is adopted to combine all these features and filter the important features.

Convolution Given the input matrix $\mathbf{X} = [\mathbf{w_1}, \mathbf{w_2}, \cdots, \mathbf{w_n}]$ and the window size of convolutional filter l, we adopt multiple convolutional filters $\mathbf{F} = \{\mathbf{f_1}, \cdots, \mathbf{f_K}\}$ to capture semantic features of sentences. The convolution operation between the i-th filter and the j-th window is computed as:

$$c_{ij} = \mathbf{f_i} \odot \mathbf{w_{j:j+l-1}} \ , \tag{1}$$

where $\odot$ is the inner-product operation. The output of the convolution layer is a sequence of vectors $\mathbf{C} = \{\mathbf{c_1}, \mathbf{c_2}, \cdots, \mathbf{c_K}\}$.

Pooling Traditional max-pooling can only extract the most significant feature from the whole sentence, which would lose some structural information about two entity mentions. As a result, Following Piece-wise Pooling in [19], each $\mathbf{c_i}$ is divided into three segments $\{c_{i1}, c_{i2}, c_{i3}\}$ by the two entities. Then the output of the pooling layer:

$$p_{ij} = \max(\mathbf{c_{ij}}) \quad 1 \le i \le K, j = 1, 2, 3 \ . \tag{2}$$

The pooling result of the i-th filter is a 3-dimensional vector $\mathbf{p_i} = [p_{i1}, p_{i2}, p_{i3}]$. We concatenate all K filters vectors $\mathbf{P} \in \mathcal{R}^{3K}$ and feed $\mathbf{P}$ into a nonlinear layer, e.g., tanh to get the output vector for this sentence s_i: $\mathbf{s_i} = \tanh(\mathbf{P})$.

Entity Concept Enhancement The representation $\mathbf{s_i}$ only considers the feature of sentences and neglects the useful extra knowledge about the entities in the sentence. Here we utilize the concept embedding to enhance representation of the sentence.

Similar to word embedding and position embedding, entity concept embedding is the distributed representation of entity concept and each concept is encoded into a low-dimension and real-value vector. Given a entity pair $\mathtt{e_1}$ and $\mathtt{e_2}$

in sentence s_i, the concepts are denoted as $\mathtt{t}_1, \mathtt{t}_2$ respectively. Then we integrate the concept information into sentence representation via concatenating the concept embedding with the sentence representation $\mathbf{s_i}$. For entities with multiple concepts, we just adopt the average of the concept embeddings.

The final representation of sentence s_i enhanced by entity concepts is:

$$\hat{\mathbf{s_i}} = [\mathbf{t_1}, \mathbf{s_i}, \mathbf{t_2}] \ , \tag{3}$$

where $\mathbf{t_1}, \mathbf{t_2} \in \mathcal{R}^{d_t}$ are the embeddings of t_1, t_2, and d_t is the dimension of the entity concept embedding.

3.2 Relation Classification Module

As illustrated in Fig. 2, we define the final output $\mathbf{o}$ of our model which corresponds to the scores of s_i associated to all relations:

$$\mathbf{o} = \mathbf{R}\hat{\mathbf{s_i}} + \mathbf{b} \ , \tag{4}$$

where $\mathbf{o} \in \mathcal{R}^{N_r}$ and N_r is the number of relations and $\mathbf{b} \in \mathcal{R}^{N_r}$ is a bias item. Especially $\mathbf{R}$ is the `relation prediction embeddings matrix`. The j-th row vector of $\mathbf{R}$ correspond to the predictive vector of the relation r_i which is used to evaluate the score of s_i associated with relation r_j, denoted as $\mathbf{r_j^p}$.

Afterwards, the conditional probability that the sentence s_i can express the relation r_j is defined with a soft-max layer as:

$$p(r_j|s_i, \theta) = \frac{\exp(\mathbf{o}_{r_j})}{\sum_{j=1}^{N_r}(\mathbf{o}_j)} \ , \tag{5}$$

where θ is the set of learned parameters in our model. To train our model, we define two kinds of loss function as below:

Cross-Entropy Loss We use cross entropy to define the loss function and maximize the likelihood of all instances in the training data:

$$\mathcal{L}_b = -\sum_{j=1}^{T} \log p(r_j|s_i, \theta). \tag{6}$$

Mean Squared Error for Relation Embeddings In order to attach semantics to relation labels rather than meaningless one-hot vector, we introduce relations description embedding to regularize the loss function. We adopt a Convolutional Neural Networks to extract semantic features from the textual descriptions for each relation. The description embedding for relation r_j is denoted as $\mathbf{r_j^d}$.

Hence, we have defined two embedding vectors for each relation: `description embedding` $\mathbf{r_d}$ and `prediction embedding` $\mathbf{r_p}$ which are two views about each relation. They should be, to some extent, closer to each other in the vector space. As a result, we constrain that `description embedding` $\mathbf{r_d}$ is similar

Table 1. Statistics of NYT+Freebase and GDS datasets.

Dataset	# relations	# sentences	# entity-pair	# entity concept
NYT+Freebase Dataset				
Train	53	455,771	233,064	55
Dev	53	114,317	58,635	55
Test	53	172,448	96,678	55
GDS Dataset				
Train	5	11,297	6,498	25
Dev	5	1,864	1,082	25
Test	5	5,663	3,247	25

with `prediction embedding` $\mathbf{r_p}$. So we define another loss function using mean squared error as a constraint item:

$$\mathcal{L}_r = \sum_{j=1}^{N_r} ||\mathbf{r_j^p} - \mathbf{M}\mathbf{r_j^d}||_2^2, \tag{7}$$

where M is a harmony parameter matrix. In this way, the useful semantic information in relation descriptions can be integrated into relation classification module effectively.

The final loss function is defined :

$$\mathcal{L} = \mathcal{L}_b + \lambda \mathcal{L}_r \ , \tag{8}$$

where $\lambda > 0$ is a balance weight for the two loss components.

4 Experiments

4.1 Dataset and Evaluation Metrics

Our experiments are conducted on NYT+Freebase and Google Distant Supervision (GDS) datasets. The details of the two datasets are followed:

NYT+Freebase[5]**:** The dataset is built by [12] and generates by aligning entities and relations in Freebase [1] with the corpus New York Times (NYT).‘ The articles of NYT from year 2005-2006 are used as training data, and articles from 2007 are used as testing data. NYT+Freebase is widely used as a benchmark dataset in relation extraction field [19, 8, 18, 15].

GDS[6]**:** This dataset is recently built in [5] by Google Research. Different from NYT+Freebase, GDS is a human-judged dataset and each entity-pair in the dataset is judged by at least 5 raters.

The statistics of the two datasets is summarized in Table 1.

In addition, we extract off-the-shelf descriptions of the relations from WikiData [7] [16] (a migration project of Freebase).

[5] http://iesl.cs.umass.edu/riedel/ecml/
[6] https://ai.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html
[7] https://www.wikidata.org/

Evaluation Metrics Following previous works [8, 19], we evaluate our approach with held-out evaluation, which automatically compares the relations extracted in our model against the relation labels in Freebase. Similar to previous works [19, 8], we also present the precision-recall curves and P@Top N metrics to conduct the held-out evaluation.

4.2 Experimental Settings

Embedding Initialization For position embedding and entity concept embedding, we just randomly initialize them from a uniform distribution. For word embedding, we adopt word2vec[8] [9] to train word embeddings on corpus.
Parameter Settings We tune hyper parameters using the validation datasets in experiments. The best parameter configuration is: the dimension of word embedding $d_w = 50$, the dimension of type embedding $d_t = 20$, the dimension of position embedding $d_p = 5$, the window size of the filter $w = 3$, the number of filters $N_f = 230$, the mini batch size $B = 64$, the weight $\lambda = 1.0$, the dropout probability $p = 0.5$.

4.3 Performance Evaluation

To demonstrate the effect of our approach, in this section, we conduct held-out evaluation and compare with three traditional feature-based methods and five competitive neural network based methods as follows:
Feature-based methods:

Mintz [10] is a traditional method which extracts features from all sentences.

MultiR [4] adopts multi-instance learning which can handle overlapping relations.

MIMLRE [14] is a multi-instance and multi-label method for distant supervision.

Neural network based methods:

PCNN [19] utilizes a convolutional neural network to encode sentences and uses the most-likely sentence for the entity pair during training.

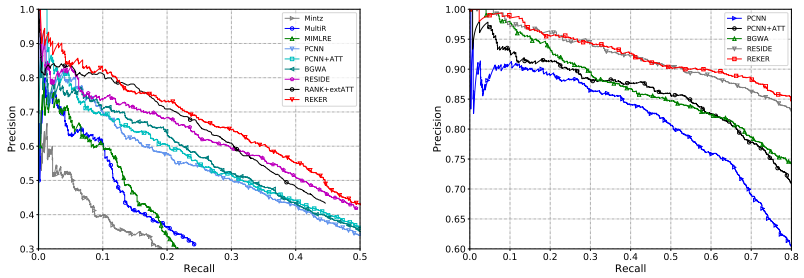
PCNN+ATT [8] adopts a sentence-level attention to reduce the noise data in distant supervision.

BGWA [5] proposes an entity-aware attention model based on Bi-GRU to capture the important information.

RANK+extATT [18] makes joint relation extraction with a general pairwise ranking framework to learn the class ties between relations, which achieves state-of-art performance.

RESIDE [15] employs graph convolution networks to encode the syntactic dependency tree of sentences and utilizes the side information of entity type and relation alias to enhance relation extraction.

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



(a) NYT+Freebase dataset. (b) GDS dataset.

Fig. 3. Precision-recall curves.

Table 2. Precision@Top K of extracted relations in NYT+Freebase dataset.

	Top 100	Top 200	Top300	Top 500	Top600	Top800	Average
Mintz	0.53	0.51	0.49	0.42	0.37	0.34	0.44
MultiR	0.62	0.63	0.63	0.48	0.44	0.40	0.53
MIML	0.68	0.64	0.62	0.51	0.46	0.42	0.56
PCNN	0.78	0.72	0.67	0.60	0.57	0.55	0.65
PCNN+ATT	0.81	0.71	0.69	0.63	0.60	0.57	0.67
BGWA	0.82	0.75	0.72	0.67	0.62	0.54	0.69
RESIDE	0.84	0.79	0.76	0.70	0.65	0.62	0.72
RANK+extATT	0.83	0.82	0.78	0.73	0.68	0.62	0.74
REKER	**0.85**	**0.83**	**0.78**	**0.73**	**0.70**	**0.65**	**0.76**

The Precision-Recall curves of the the NYT+Freebase dataset and GDS[9] dataset are shown in Fig. 3. We can see that our method REKER can outperform all other methods in almost the entire range of recall. From the comparison in terms of the Presion Recall curves, we can observe that: (1) all the neural network based methods outperform feature-based methods a lot. This is because that the neural network methods can learn better representation about sentences and then benefit for the relation extraction task. (2) our method REKER outperforms the basic model PCNN, which shows that the two components (i.e., entity and relation knowledge) in our method can indeed improve the performance of relation extraction tasks. (3) our method REKER achieves better performance than RESIDE in the NYT+Freebase dataset, which also incorporates the entity and relation information. This is because the information of relation alias used in RESIDE is limited, and our method REKER adopts the richer relation description to conduct relation label embedding, which can play a direct and global role for the model. (4) since there are only 5 relations in GDS dataset, most methods perform much better than in NYT+Freebase and our method REKER can achieve a slightly better performance than the state-of-art method (i.e., RESIDE). Especially when the recall is beyond 0.5, our model can keep a higher precision.

[9] We compare with recent baselines in GDS since the dataset is newly released in 2018.

Table 3. Examples for the effectiveness of Entity Concept.

Sentences	Entity Concept	PCNN	PCNN+ENT
... said [e_1]**Kamal_Nath** , the commerce minister of [e_2]**India** ...	[e_1]:**Person** [e_2]:**Country**	**/business/company**	**/people/nationality**
...a former minister of [e_1]**Thailand**, [e_2]**Thaksin_Shinawatra** ...	[e_1]:**Country** [e_2]:**Person**	**/location/contains**	**/people/nationality**

Besides Precision Recall curves, we evaluate models using P@N metric in held-out evaluation under NYT+Freebase dataset following previous works [8, 15, 19], which can provide the precision about the top-ranked relations. According to the P@N result in Table 2, we can see that our model REKER performs best at the entire P@N levels, which further demonstrates that our additional rich knowledge including entity concept and relation description are all useful in the relation extraction task and can extract more true positive relations.

4.4 Effectiveness of Entity and Relation Knowledge

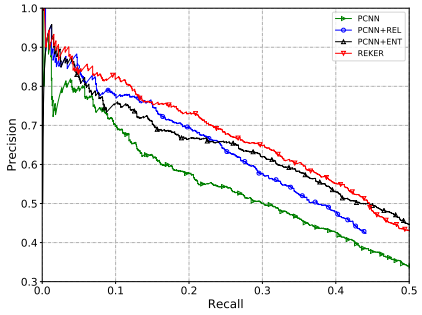


Fig. 4. Comparison of our model variants in terms of Precision-Recall curves.

We further conduct extra experiments under NYT+Freebase dataset with each component alone (i.e., PCNN+ENT and PCNN+REL respectively). The Precision Recall Curve is shown in Fig. 4, we have the following conclusions: (1) models with ENT, REL alone all consistently outperform the basic model PCNN, which denotes that the knowledge about entity concepts and relation descriptions are all beneficial to the relation extraction task. (2) our final model REKER performs best among those methods respectively, which shows that the combination of entity concepts and relation descriptions can bring a further improvement compared with the two components alone.

4.5 Case Study

To indicate the impact of additional knowledge more intuitively, in this section we show some case studies for qualitative analysis.

Entity Concept In this part, we find out some examples to show the intention of entity concept. As the first sentence in Table 3, the method PCNN extracted `/business/company` from a sentence containing a `Person` entity and a `Country` entity, which didn't make sense. Our method integrating the knowledge of entity concept can effectively avoid these wrong cases and improve the quality of extracted relations.

Relation Description As introduced above, we believe that relation description can attach semantic information into the relation labels. Hence, we select 25 relations from three largest domains to visualize their feature vectors distribution by projecting them into 2-Dimension space using PCA [6] algorithm. Fig. 5(a) shows the visualization result of the 25 relations after training a model without description information. And Fig. 5(b) is the result of the model with relation description. We can find out that relations from the same domains lie closer to each other in Fig. 5(b). In other words, different domains of relations are well separated in the model with the help of relation description information. As a result, incorporating the relation description can capture more semantic correlation between relations, which could benefit the relation classification task.

(a) without textual description. (b) with textual description

Fig. 5. Visualization of relation prediction embedding

5 Conclusion

In this paper, we propose a novel approach REKER that integrates the rich knowledge including the relation description and entity concept into a neural relation extraction model. We adopt entity concept embedding to enhance the representation of sentences. Besides, we utilize relation description information to attach semantics into relation labels rather than meaningless one-hot vectors. Experimental results on two benchmark datasets show that our approach REKER can effectively improve the performance of the relation extraction task.

Acknowledgement

This work was supported by the National Key R&D Program of China (2018YFC0831005), the Science and Technoloy Key R&D Program of Tianjin (18YFZCSF01370) and the National Social Science Fund of China (15BTQ056).

References

1. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD. pp. 1247–1250. AcM (2008)
2. Bunescu, R.C., Mooney, R.J.: A shortest path dependency kernel for relation extraction. In: HLT. pp. 724–731. Association for Computational Linguistics (2005)
3. GuoDong, Z., Jian, S., Jie, Z., Min, Z.: Exploring various knowledge in relation extraction. In: ACL. pp. 427–434. Association for Computational Linguistics (2005)
4. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: ACL. pp. 541–550. Association for Computational Linguistics (2011)
5. Jat, S., Khandelwal, S., Talukdar, P.: Improving distantly supervised relation extraction using word and entity based attention. CoRR **abs/1804.06987** (2018)
6. Jolliffe, I.: Principal component analysis. In: International encyclopedia of statistical science, pp. 1094–1096. Springer (2011)
7. Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: ACL. p. 22. Association for Computational Linguistics (2004)
8. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: ACL. vol. 1, pp. 2124–2133 (2016)
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS. pp. 3111–3119 (2013)
10. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: ACL. pp. 1003–1011. Association for Computational Linguistics (2009)
11. Ren, X., Wu, Z., He, W., Qu, M., Voss, C.R., Ji, H., Abdelzaher, T.F., Han, J.: Cotype: Joint extraction of typed entities and relations with knowledge bases. In: WWW. pp. 1015–1024 (2017)
12. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: ECML-PKDD. pp. 148–163. Springer (2010)
13. Riedel, S., Yao, L., McCallum, A., Marlin, B.M.: Relation extraction with matrix factorization and universal schemas. In: NAACL. pp. 74–84 (2013)
14. Surdeanu, M., Tibshirani, J., Nallapati, R., Manning, C.D.: Multi-instance multi-label learning for relation extraction. In: EMNLP. pp. 455–465. Association for Computational Linguistics (2012)
15. Vashishth, S., Joshi, R., Prayaga, S.S., Bhattacharyya, C., Talukdar, P.: RESIDE: improving distantly-supervised neural relation extraction using side information. In: EMNLP. pp. 1257–1266 (2018)
16. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)
17. Xu, K., Reddy, S., Feng, Y., Huang, S., Zhao, D.: Question answering on freebase via relation extraction and textual evidence. In: ACL (2016)
18. Ye, H., Chao, W., Luo, Z., Li, Z.: Jointly extracting relations with class ties via effective deep ranking. In: ACL. pp. 1810–1820 (2017)
19. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: EMNLP. pp. 1753–1762 (2015)
20. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: COLING. pp. 2335–2344 (2014)