IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received January 24, 2019, accepted February 15, 2019, date of publication February 26, 2019, date of current version March 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2901544

Representation Learning of Knowledge Graphs via Fine-Grained Relation Description Combinations

MING HE, XIANGKUN DU, AND BO WANG
Faulty of Information Technology, Beijing University of Technology, Beijing 100124, China
Corresponding author: Ming He (heming@bjut.edu.cn)

This work was supported in part by the Beijing Natural Science Foundation under Grant 4192008 and in part by the General Project of Beijing Municipal Education Commission under Grant KM201710005023.

ABSTRACT Knowledge representation learning attempts to represent entities and relations of knowledge graph in a continuous low-dimensional semantic space. However, most of the existing methods such as TransE, TransH, and TransR usually only utilize triples of knowledge graph. Other important information such as relation descriptions with relevant knowledge is still used ineffectively. To address these issues, in this paper, we propose a *relation text-embodied knowledge representation learning* method, in which relation descriptions are adopted as side information for representation learning. More specifically, we explore a convolutional neural model to build representations of fine-grained relation descriptions. Furthermore, knowledge representations of triples and representations of fine-grained relation descriptions are jointly embedding. Our model is evaluated on the tasks of both link prediction and triple classification. The experiment results show that our model exhibits a superior performance than other baselines, which demonstrates the availability of our method with fine-grained relation descriptions and knowledge graph jointly embedding.

INDEX TERMS Knowledge representation, representation learning, knowledge graph.

I. INTRODUCTION

Knowledge graphs, which organize human knowledge into structured information, are crucial resources for some intelligent applications such as question answering, personalized recommendation system and Web search [19]. A typical knowledge graph usually arranges multi-relational data in the form of triple facts (*head entity, relation, tail entity*) which is abbreviated as (*h,r,t*).

In the last several decades, there have made a great progress in building large-scale knowledge graphs. However, applications of knowledge graphs are subjected to challenges of computational inefficiency as well as data sparsity with the increase in knowledge graph size. To address this issue, a new approach has been proposed which attempts to project a knowledge graph into a continuous vector space while retaining certain properties of the original graph [3], [5]–[7], [18], [24]. Furthermore, many representation learning methods concern only structured information in knowledge graphs, while ignoring the rich textual information related to knowledge base.

Recently, combinations of multiple source information have been shown to be an effective approach in improving the performance of knowledge representation. Reference [25] proposes an original representation learning model for knowledge graphs using entity descriptions. However, the entity descriptions are only considered when utilizing textual information for representation learning, while there is various textual information of relation. Reference [14] proposes a model that captures the compositional structure of textual relations, and jointly optimizes entity, knowledge base, and textual relation representations. Nevertheless, only the compositional structure of relations was used rather than the rich texts about relations.

There are lots of information related to relation of knowledge base on the Internet. For example, as shown in Fig. 1, we exhibit the relation descriptions in a fact triple which extracted from Freebase, a large-scale knowledge graph maintained by Google.

To utilize rich relation descriptions as a supplement to triple facts that helps to represent knowledge graphs, we present a new representation learning model for knowledge graphs, named as Relation Text-embodied Knowledge Representation Learning (RTKRL), which is inspired

The associate editor coordinating the review of this manuscript and approving it for publication was Chang Choi.

by [25]. Since relations of Freebase have three layers—domain, type and topic—whose ranges are in descending order. In this paper, we mainly consider the accuracy of modeling relations and treat the fine-grained descriptions of topic as the descriptions of the entire relation as shown in Fig. 1.

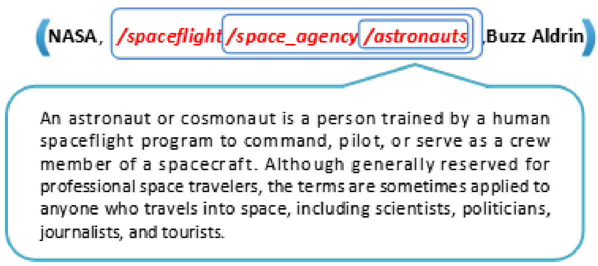


FIGURE 1. Example of relation descriptions in Freebase We treat the descriptions of astronauts as fine-grained descriptions for astronauts is the third layer of this relation. The descriptions explain what astronauts are well. Even those who have little knowledge of astronauts would be able to understand that *Buzz Aldrin* is a person called astronaut worked at *NASA* based on the descriptions. We want to extract the features of the descriptions with a convolutional neural network (CNN) for improving the performance of link prediction and other tasks.

In the RTKRL model, we model the fact triples along with their relation descriptions for the embedding of relations. We acquire text-based representations with considering the relation descriptions, and the structure-based representations are obtained by translation-based methods. The relation representations are modeled by projecting text-based and structure-based representations into a same continuous low-dimensional vector space. The main contributions of this work are summarized as follows:

1) We investigate the new problem of exploiting fine-grained relation descriptions, which promote the representation learning of knowledge for describing relation in triples.
2) We propose a novel model named RTKRL, which captures the features of fine-grained relation descriptions to improve the performance of representation learning.
3) We conduct experiments on real-world dataset to evaluate the effectiveness on both tasks of link prediction and triple classification and acquire promising performances.

The rest of this paper is organized as follows. Section 2 presents the related work and Section 3 describes the methodology of the RTKRL model. Section 4 discusses the experimental results, and Section 5 presents the conclusion and future work of this paper.

II. RELATED WORK

A. TRANSLATION-BASED MODELS

There are various methods in representation learning for knowledge graphs and great advances have been made in these methods in recent years. TransE [5] considers the relations as translating operations from head entities to tail entities in a low-dimensional vector space. This means that (**t**) should be the closest to (**h**+**r**) given a triple (*h,r,t*). Therefore, the score function of TransE is defined as

$$f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|, \tag{1}$$

fr(*h,t*) is small if (*h,r,t*) exists, and large otherwise. TransE makes a good effect in 1-to-1 relation but has problems with modeling 1-to-N, N-to-1 and N-to-N relations. TransH [23] is presented to solve the issues of TransE by treating the relation as a hyperplane with a normal vector w_r. The entity embeddings **r** and **h** are projected to the hyperplane of w_r, making entities to have diverse distributed representations in distinct relations. TransR [16] models relations and entities in different vector spaces and projects entities from an entity space to a relation space with a relation-specific matrix. TransD [11] introduces a dynamic mapping matrix for different types of entities while considering the diversities of both entities and relations. In this paper, we integrate the structural knowledge representation obtained by TransE, TransH and TransR with the textual representation for fully leveraging the relation descriptions.

B. REPRESENTATION LEARNING WITH TEXTUAL INFORMATION

Textual information is significant in representation learning of knowledge graphs, which has attracted great attention recently. It can be considered as a supplement to triple facts that has shown significant contribution to represent knowledge graphs. Reference [22] makes an attempt to project words and entities into a same continuous vector space jointly by alignment models in Wikipedia anchors and entity names. Reference [28] proposes a joint learning approach that learns the vectors of entities by leveraging resources of both graph knowledge and text data. However, those methods neglect the word orders of descriptions and struggle with ambiguity while using entity names. Reference [25] proposes description-based representations for entities constructed from entity descriptions with CBOW or CNN, which can model entities in zero-shot scenario. Reference [14] proposes a model that captures the compositional structure of textual relations, and jointly optimizes entity, knowledge base, and textual relation representations. Different from previous works which make use of text data about entities, we present a model to integrate the structural knowledge of triples with text about relations for representation learning.

C. CNN FOR MODELING SENTENCE

The bag-of-word (BOW) model [1], [21] uses a fixed length vector of word counts to represent text. However, the model disregards word sequences with ignoring word orders in sentences and is vulnerable to the sparsity problem, which results in poor generalization performance. Originally proposed for computer vision, CNN model is widely applied in natural language processing and has achieved excellent results in sentence modeling [12], search query retrieval [17], semantic parsing [26] and other traditional NLP tasks [8]. Zhang *et al.* [12] present a convolutional framework called the Dynamic Convolutional Neural Network (DCNN) which can be used for the semantic modeling of sentences. In [29], the authors explore the application of temporal ConvNets to text which are considered as a kind of raw signal at the character level. Reference [13] explores sentence-level classification tasks with convolutional neural networks trained on word vectors, and obtains excellent results in the experiments. Conneau *et al.* [9] propose a novel framework where small convolutions and pooling operations are used only on the character for text processing. In our CNN encoder, we use the word embeddings learned from significant amounts of unlabeled data other than the randomly initialized embeddings inspired by [8].

III. METHODOLOGY

A. OVERALL ARCHITECTURE

We introduce the overall architecture of the RTKRL model. Enlightened by the model shown in [25], we define the energy function as follows:

$$E(h, r, t) = E_s + E_t, \tag{2}$$

where the energy function of the structure-based representations is denoted as E_s, which is the same as the energy function of TransE shown in Equation 1. The energy function of the text-based representations is E_t, which has good effect in capturing textual information contained in relation descriptions. We define E_t as follows:

$$E_t = \|\mathbf{h} + \mathbf{r_t} - \mathbf{t}\|, \tag{3}$$

where the head and tail are structure-based representations, and the relation is replaced by $\mathbf{r_t}$ which is text-based representation obtained from the CNN encoder. The energy function is used to project two types of relation representations into a same continuous vector space while the entity representations are the same for the two energy functions.

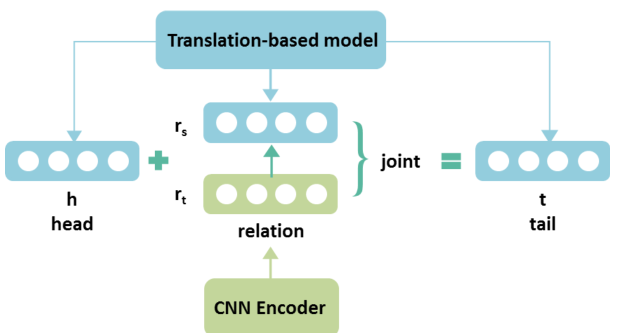


FIGURE 2. Overall architecture of the RTKRL model.

The overall framework of the RTKRL model is shown in Fig. 2. To utilize both fact triples and relation descriptions, we propose two types of relation representations, which are text-based and structure-based representations. Firstly, we acquire the structure-based representations of relations and entities by translation-based model like TansE. We build text-based representations of fine-grained relation descriptions using the CNN encoder. Furthermore, text-based and structure-based representations are jointly embedding with the energy function shown in (2).

B. WORD REPRESENTATION

In the CNN encoder, we take the word embeddings of relation descriptions as inputs. As in [27], the word representations consist of two parts: word features (WF) and position features (PF).

C. WORD FEATURES

The word features could be learned from vast amounts of mostly unlabeled training data, as the word embeddings learned from it are far more satisfactory than the randomly initialized embeddings. We select the trained embeddings provided by [20].

D. POSITION FEATURES

Reference [27] proposes position features which are considered as a combination of the relative distances of the current word to w_1 and w_2. Given a sentence represented as a sequence $s = (w_1, w_2, \ldots, w_n)$, in which w_i is the i-th word of this sentence, we mark the relation name as 0, and the position features of other words are relevant to the distance to relation name. The words on the left are marked positive values, and negative values for the right words The position features having a distance larger than d are marked as *-d* or d. Relative distances are projected into a low-dimensional vector $\mathbf{d}$, and $PF = [\mathbf{d}]$. The word representations are represented as $[WF, PF]^T$, which is treated as the input of the CNN encoder.

E. CONVOLUTIONAL NEURAL NETWORK ENCODER

Convolutional neural network is a high-performance feed-forward neural network with convolution layers and pooling operations. It has recently been used to improve the state of the art in large-scale image classification and speech recognition by a large margin [10], [15] and has achieved excellent results in some natural language processing tasks such as sentence modeling, semantic parsing, prediction, search query retrieval and classification. In view of CNN is capable of capturing global and local features and significantly enhancing the efficiency and accuracy, we use the CNN encoder to capture the implicit information contained in the relation descriptions.

The convolutional neural network encoder in this paper is shown in Fig. 3. In the CNN encoder, there have five layers. We process the descriptions of relations and treat them as inputs of the CNN encoder. The text-based representations of relations are output through the encoder, and the entity embeddings should then be acquired to minimize the energy function of the RTKRL.

F. CNN WITH CONVOLUTION

There are two convolution layers in the RTKRL model. The convolution layer is the core of the neural network. In every

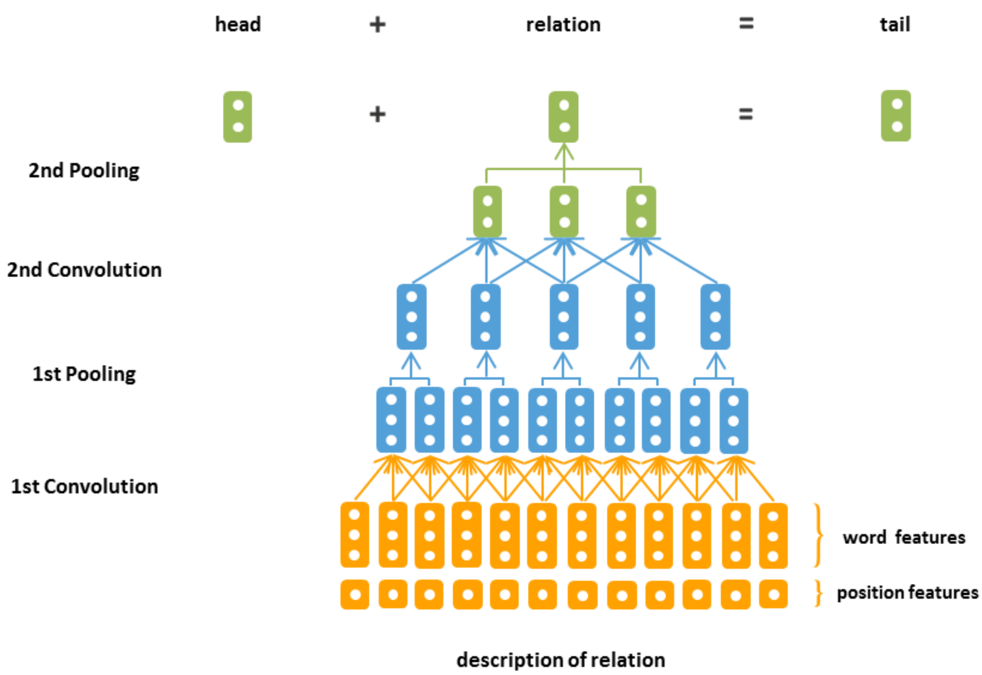


FIGURE 3. The convolutional neural network encoder.

convolution layer, there exist several convolution kernels with receptive fields and different features are extracted with the convolution kernels slide through the original input. The parameters of each convolution unit are optimized by back propagation algorithm. After these operations, filters can be learnt and will activate when they extract some specific type of features in the input.

As in [25], we set $X_i^{(l)}$ as the input of the i-th vector in l-th convolution layer. $A_i^{(l)}$ represents the output of the i-th vector in the l-th convolution layer. In the first layer, $X^{(l)}$ is a set of vectors $(x_0, x_1, \cdots, x_n)$ that represents the relation descriptions. We first acquired $X^{(l)}$ using a window of size k that slides through $X^{(l)}$. For the window operation, we have

$$x_i^{'(1)} = x_{(i:i+k-1)} = [x_i^T, x_{(i+1)}^T, \ldots x_{(i+k-1)}^T]^T, \tag{4}$$

where $x_i^{'(1)}$ is acquired with k column vectors that are concatenated in the i-th window of the input sentence. Considering when proceeds window process, the length of the inputs change, we add zero-padding at the boundary of input matrix. We obtain the i-th vector of the output in the convolution layer as follows:

$$z_i^{(l)} = \sigma(W^{(l)} x_i^{'(l)} + b_i^{(l)}), \tag{5}$$

in which $b_i^{(l)}$ is the optional bias and $W^{(l)} \in R^{(n_2^{(l)} * n_1^{(l)})}$ is considered as the convolution kernel for $x_i^{'(1)}$ which is obtained from the window operation. $n_2^{(l)}$ is the number of feature maps and is treated as the dimension of the output vectors. $n_1^{(l)} = k * n_0^{(l)}$ where $n_0^{(l)}$ is the dimension of the input vectors. σ is the activation function which is usually ReLU or tanh. As zero-padding vectors should neither contribute to forward propagation nor be updated in back propagation, we can also make the variable length of the input sentences to be aligned as well as prevent the possible side effects of zero-padding.

G. CNN WITH POOLING

The pooling layer is inserted between consecutive convolution layers in a convolutional neural network. A pooling layer can effectively reduce the spatial size of representations thus reducing the number of parameters and computations of the network, as well as preventing overfitting. The filter of a pooling layer merely affects those slices of the input at the same depth and resizes it spatially, using different pooling strategies for different layers.

We use the max operation in the first pooling layer. The output vectors of the previous convolution layer are divided into n non-overlapped windows. For every window, we make up a new vector by selecting the maximum of every feature map. The depth dimension does not change.

$$x_i^{(2)} = max(z_{n*i}^{(1)}, \ldots z_{n*(i+1)-1}^{(1)}), \tag{6}$$

The pooling process can narrow the size of feature representations by n times, which lowers the complicacy of the CNN encoder. Unfortunately, merely using max-pooling results in information loss, because the maximum of every feature map is selected while ignoring other input vectors that contain different local information. Considering this, we take mean-pooling before activation to build relation representations for the second pooling layer. All the input vectors are added to build representations.

$$x^{(3)} = \Sigma_{i=1,\ldots,m} \frac{z_i^{(2)}}{m}, \tag{7}$$

H. TRAINING

Our training objective is based on a margin-based score function, and we define the overall score function as follows:

$$L = \sum_{(h,r,t)\in T} \sum_{(h',r',t')\in T'} max(\gamma + d(h+r,t) - d(h'+r',t'), 0), \tag{8}$$

in which $\gamma > 0$ and is a margin hyper parameter. $d(h+r,t)$ represents the dissimilarity function score of the positive triple, while $d(h'+r',t')$ represents the dissimilarity function score of the negative triple. T' is the negative triples sampling from T, which is a positive triple set. Owing to a lack of negative triples in the knowledge graphs, we construct T' as follows:

$$T' = (h', r, t) \mid h' \in E \cup (h, r, t') \mid t' \in E \cup (h, r', t) \mid r' \in R \tag{9}$$

where the head entity, tail entity and relation are randomly substituted by other relations in **R** or entities in **E**. Moreover, if new triples exist in T, they are not treated as negative instances. There are two types of relation representations, which include structure-based and text-based representations.

We use the energy function in Equation (2) to learn structure-based representations and text-based representations into the same vector space. The objective formalization is then used to train our model.

I. IMPLEMENTATION DETAILS

The CNN encoder takes texts obtained from Wikipedia as inputs and outputs the entity embeddings. The parameters of this model can be denoted as $\mathbf{W}$, $\mathbf{W^{(1)}}$, $\mathbf{W^{(2)}}$, $\mathbf{E}$ and $\mathbf{R}$. $\mathbf{W^{(1)}}$ and $\mathbf{W^{(2)}}$ represent the convolution kernels in different layers and are randomly initialized, and **E**, **R** and **W** represent the embeddings of entities, relations and words.

In this paper, we use the stochastic gradient descent (SGD) for the learning of the RTKRL model. We initiate all the transfer matrices as identity matrices and initiate **E** and **R** with the results of TransE in order to avoid overfitting and speed up the convergence. **W** is pre-trained through Word2Vec using the Wikipedia corpus.

IV. EXPERIMENTS

A. DATASETS AND EXPERIMENT SETTINGS

1) DATASETS

Most of experiments in previous work made use of FB15K and WN18 [5] for evaluation. However, WN18 is unsuitable for our method because the relation descriptions of relations in WN18 is inappropriate to extract. In the experiments of this paper, we evaluate our model with FB15K [5], a subset of Freebase which provides general facts of the world.

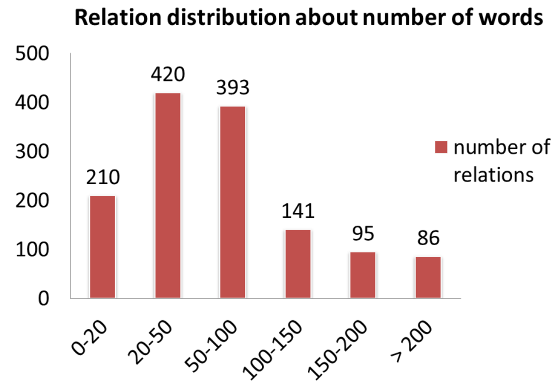


FIGURE 4. The statistics of relation descriptions.

FB15K comprises 1345 relations. For each relation, we use Wikipedia as the corpus to extract the related description. The statistics of relation descriptions are shown in Fig. 4. From Fig. 4, we can observe that the number of words in the majority of relations are between 20 and 100. There are 94(6.9%) relations with no descriptions that cannot be extracted from Wikipedia, which is the part we do not use. We process the raw texts by removing all stop words and mark the position of the relation name. After the preprocessing, the description of every relation comprises more than five words, the description contains a maximum of 291 words, and the average number of words in relation descriptions is 86. The detailed statistics of FB15K are presented in Table 1.

TABLE 1. Statistics of data sets.

Dataset	#Rel	#Ent	#Train	#Valid	#Test
FB15K	1345	14951	483142	50000	59071

2) PARAMETER SETTINGS

We execute TransE, TransH and TransR as our baselines following the same experimental settings reported in their papers, and we use the RTKRL method along with one of the three translation-based models for the purpose of comparison in evaluation. The dimensions of entities and relations in all baselines are the same for ensuring a fair comparison. We select the entity/relation dimension n among {50, 80, 100}, the margin γ among {1, 2} and the learning rate for the stochastic gradient descent among {0.005, 0.001, 0.002}. As for the CNN encoder, we set up the dimension of the feature map n_f among {50, 100, 150} and the dimension of word embedding n_w among {50, 80, 100}. In this five-layer architecture, there are two pooling layers in which the first pooling layer performs 4-max-pooling and the second pooling layer performs mean-pooling. In the convolution layer, we select a different window size k among {1, 2, 3}.

B. LINK PREDICTION

Link prediction aims at predicting the missing h or t when a triple (h,r,t) is given, used in [3] and [5]. This task ranks a set of candidate entities other than seeking one best answer from the knowledge graph. We use the data set FB15K that is used in TransE [3], [5].

Evaluation Protocol: In this paper, we follow the protocol used in TransE [3]. For every test triple (*h,r,t*), we substitute the head/tail entity *h/t* with every entity *e* in turn in the dataset and calculate the score of dissimilarity of the corrupted triplet (*h,r,e*) by function f_r. Acquiring the scores, we rank it in an ascending sort order and obtain the rank of the original correct triplet. Following [5], we treat two measures as our evaluation metric: (1) the proportion of correct entities in top-10 ranked entities (Hits@ 10); (2) the mean rank of the correct entities. A good embedding model should achieves a high Hits@10 and a low mean rank. This evaluation setting is called "Raw." However, a corrupted triplet may exist in the training, validation and test datasets, which should also be considered as correct. Taking this into consideration, we remove the corrupted triplets that appear in the knowledge graph before the ranking. This setting is called "Filter." In this paper, we evaluate the results of both these settings.

TABLE 2. Evaluation results on entity prediction.

Metric	Mean Rank		Hits@10(%)	
	Raw	Filter	Raw	Filter
TransE [5]	243	125	34.9	47.1
TransH [23]	212	87	45.7	64.4
TransR [16]	198	77	48.2	68.7
RKTKL(TransE)	212	114	47.9	64.6
RKTKL(TransH)	195	83	48.8	65.7
RKTKL(TransR)	183	72	50.1	69.9

TABLE 3. Evaluation results on relation prediction.

Metric	Mean Rank		Hits1(%)	
	Raw	Filter	Raw	Filter
TransE [5]	2.91	2.53	69.5	90.2
TransH [23]	2.71	2.32	69.9	90.8
TransR [16]	2.49	2.09	70.2	91.6
RKTKL(TransE)	2.74	2.33	70.0	91.0
RKTKL(TransH)	2.45	2.14	70.4	91.9
RKTKL(TransR)	2.38	2.12	70.8	92.7

Implementation: As the dataset of our task is used in the other baselines, we copy the evaluation results from literatures of [4], [16], and [22]. In order to get the best configuration, we have tried several settings on the validation dataset. The optimal configurations under the "bern." sampling strategy are: learning rate $\lambda = 0.001$, the entity/relation dimension $n = 100$, the dimension of word embedding $n_w = 100$, the dimension of feature map $n_f = 100$.

Results: Evaluation results on FB15K are listed in Table 2, Table 3 and Table 4. From the tables we can observe that: (1) Our RTKRL model obviously outperforms all baselines significantly in entity prediction and relation prediction, which demonstrates the availability and robustness of our model. (2) The results indicate that the CNN encoder has successfully extracted the textual information contained in the fine-grained relation descriptions, thus representing a promising improvement in the representation learning of knowledge graphs. (3) RTKRL achieves a superior performance in all mapping categories of relations (1-to-1, N-to-1, 1-to-N, N-to-N), which indicates that RTKRL is capable of dealing with complex relations and promotes the performance in the case of simple relations.

The improvement of our model over the baselines could be attributed to the following reasons: (1) CNN encoder extracts text-based representations successfully from the textual information in description and could supply implicit information for representation learning of knowledge graph. In some circumstance, the use of only structural information is insufficient for capturing the details. For example, just as the triple shown in Fig. 5, when the head entity is missing, it is more likely to predict that the head entity is *NASA* from among the candidates with the relation descriptions as we extract the features from it. (2) With the rich descriptions of relations, RTKRL acquires more accurate representations for both relations and entities and their complex relevance. (3) Joint embedding provides a superior performance in the task of predicting facts than text and knowledge graph embedding and it has been validated in large scale experiments on Freebase.

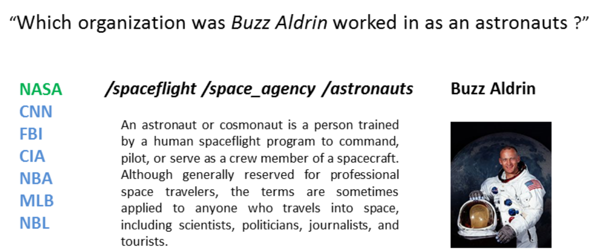


FIGURE 5. Example of link prediction.

C. TRIPLE CLASSIFICATION

Triple classification is a task to judge the correctness of a given triple (*h,r,t*). It has been explored in [18] and [23] and be considered as a binary classification task. We evaluate this task on FB15K in our experiment.

Evaluation Protocol: We comply with the protocol used in NTN [18]. Since there needs negative labels in this classification task and there are no explicit negative instances in FB15K, we construct the negative triples following the setting using in [18], wherein every golden triple is corrupted to obtain one negative triple.

In this classification task, we set a threshold δ_r related to relation for every relation. Given a triple (*h,r t*), the triple is classified as positive if the dissimilarity score acquired from f_r is less than δ_r and negative otherwise. The relation-specific threshold δ_r is optimized by maximizing the classification accuracies on the validation set.

Implementation: Since the dataset is alike, we can refer to the evaluation results of other baselines directly from the paper, as in [4], [16], and [22]. In order to acquire the best configuration, we have tried several kinds of settings on

TABLE 4. Experimental results on FB15K by mapping properities of relations (%).

Tasks	Predicting Head(Hit 10)				Predicting Tail(Hit 10)			
Relation Category	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
TransE [5]	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH [23]	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR [16]	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
RTKRL(TransE)	49.6	71.7	23.3	51.3	49.5	22.9	71.2	52.4
RKTKL(TransH)	70.7	89.4	31.9	66.8	71.1	38.9	83.8	68.5
RKTKL(TransR)	83.5	92.6	36.7	70.8	83.8	38.1	90.8	72.9

validation dataset for all evaluated models. Under the "bern." sampling strategy, we find the following configurations are the best: the entity/relation dimension $n = 100$, learning rate $\lambda = 0.001$, the dimension of feature map $n_f = 100$, the dimension of word embedding $n_w = 100$.

TABLE 5. Evaluation results on triple classification.

Method	FB15K
TransE [5]	79.2
TransH [23]	80.2
TransR [16]	83.9
RKTKL(TransE)	80.8
RKTKL(TransH)	82.1
RKTKL(TransR)	85.4

Results: The results of the triple classification are listed in Table5. We can conclude from the Table 5 that: Our RTKRL model obviously outperforms all baselines in triple classification including TransE, TransH and TransR, which indicates the capability and extensibility of our model in triple classification.

The reasons for the above conclusions can be summarized as follows: (1) CNN features have the capacity to capture triple type information to a greater extent and achieve superior robustness. (2) Translation-based methods only concern the structural information while the CNN encoder makes use of both textual information in relation descriptions and structural information in knowledge graphs with a superior performance.

V. CONCLUSION AND FUTURE WORK

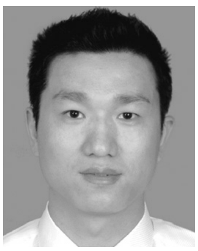
In this paper, we present the RTKRL model for the knowledge representation learning of knowledge graphs, which jointly considers triple facts and relation descriptions. Based on translation-based models, RTKRL takes relation descriptions into consideration and learns the text-based representations via the convolutional neural model for obtaining an improvement in knowledge representation. In experiments, we evaluated this model on the tasks of triple classification and link prediction. Experimental results reveal that RTKRL model provides superior performance as compared with other baselines, which indicates that RTKRL is capable of extracting features of relations from fine-grained relation descriptions.

In the future, we intend to explore the following research: (1) The RTKRL model considers only the relation descriptions, while implicit information is contained in the relation types, which can be utilized to enhance the performance of our present model. (2) We intend to explore more sophisticated models to better extract textual features for a better understanding of knowledge graphs.

REFERENCES

[1] D. Agarwal and B.-C. Chen, "Regression-based latent factor models," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 19–28.
[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. SIGMOD*, 2008, pp. 1247–1250.
[3] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Mach. Learn.*, vol. 94, pp. 1–27, Feb. 2012.
[4] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. (2013). "Irreflexive and hierarchical relations as translations." [Online]. Available: https://arxiv.org/abs/1304.7158
[5] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26. Red Hook, NY, USA: Curran Associates, 2013, pp. 2787–2795.
[6] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 301–306.
[7] K.-W. Chang, W.-T. Yih, and C. Meek, "Multi-relational latent semantic analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Washington, DC, USA: Assoc. Comput. Linguistics, 2013, pp. 1602–1612.
[8] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011.
[9] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun. (2016). "Very deep convolutional networks for text classification." [Online]. Available: https://arxiv.org/abs/1606.01781
[10] G. Hinton *et al.*, "deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
[11] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 687–696.
[12] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. (2014). "A convolutional neural network for modelling sentences." [Online]. Available: https://arxiv.org/abs/1404.2188
[13] Y. Kim. (2014). "Convolutional neural networks for sentence classification." [Online]. Available: https://arxiv.org/abs/1408.5882
[14] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2015, pp. 1499–1509.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105.
[16] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI*, 2015, pp. 2181–2187.
[17] Y. Shen, X. He, J. Gao, J. Deng, and G. Mesnil, "Learning semantic representations using convolutional neural networks for web search," in *Proc. 23rd Int. Conf. World Wide Web*. New York, NY, USA: ACM, 2014, pp. 373–374.

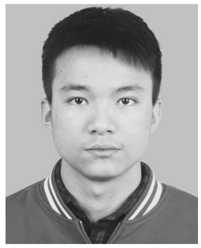
[18] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Proc. NIPS*, 2013, pp. 926–934.
[19] S. Szumlanski and F. Gomez, "Automatically acquiring a semantic network of related concepts," in *Proc. CIKM*, 2010, pp. 19–28.
[20] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: A simple and general method for semi-supervised learning," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*, 2010, pp. 384–394.
[21] H. Wang, J. Wang, M. Zhao, J. Cao, and M. Guo, "Joint topic-semantic-aware social recommendation for online voting," in *Proc. ACM Conf. Inf. Knowl. Manage.* New York, NY, USA: ACM, 2017, pp. 347–356.
[22] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph and text jointly embedding," in *Proc. EMNLP*, 2014, pp. 1591–1601.
[23] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI*, 2014, pp. 1112–1119.
[24] J. Weston, A. Bordes, O. Yakhnenko, and N. Usunier, "Connecting language and knowledge bases with embedding models for relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2013, pp. 1134–1137.
[25] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proc. AAAI*, 2016, pp. 189–205.
[26] W. T. Yih, X. He, and C. Meek, "Semantic parsing for single-relation question answering," in *Proc. Meeting Assoc. Comput. Linguistics*, 2014, pp. 643–648.
[27] 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 (COLING)*, 2014, pp. 2335–2344.
[28] D. Zhang, B. Yuan, D. Wang, and R. Liu, "Joint semantic relevance learning with text data and graph knowledge," in *Proc. 3rd Workshop Continuous Vector Space MODELS Their Compositionality*, 2015, pp. 32–40.
[29] X. Zhang, J. Zhao, and Y. Lecun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 649–657.



MING HE received the B.S. degree in computer and application and the M.S. degree in oil-gas well engineering from the Xi'an Petroleum Institute, Xi'an, Shaanxi, China, in 1999 and 2002, respectively, and the Ph.D. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, in 2005.

From 2006 to 2011, he was a Research Assistant with the College of Computer Science, Beijing University of Technology, Beijing, China, where he has been an Assistant Professor with the Faulty of Information Technology, since 2012. He has authored more than 60 articles. His research interests include recommendation systems, data mining, and machine learning.

Dr. He is a member of the International Association of Computer Science and Information Technology. His awards and honors include the Excellent Teachers Award, in 2015, and the Beijing University of Technology Best Academic Paper Award, in 2010.



XIANGKUN DU was born in Ganzhou, China, in 1996. He received the B.S. degree from the College of Water Conservancy and Environmental Engineering, Zhengzhou University, in 2017. He is currently pursuing the master's degree in information technology with the Beijing University of Technology, under the supervision of Asst. Prof. M. He.

His research interests include representation learning and recommender systems.

BO WANG was born in Langfang, Hebei, China, in 1994. He received the B.S. degree from the School of Internet of Things Engineering, Xingtai University, in 2017. He is currently pursuing the master's degree with the Beijing University of Technology, under the supervision of Asst. Prof. M. He.

His research interests include knowledge graph and recommender systems.