Representation Learning of Knowledge Graphs with Entity Descriptions

Ruobing Xie^{1,2}, **Zhiyuan Liu**^{1,2,3}, **Jia Jia**^{1*}, **Huanbo Luan**^{1,2}, **Maosong Sun**^{1,2,3}
^{1} Department of Computer Science and Technology,
^{2} State Key Lab on Intelligent Technology and Systems,
National Lab for Information Science and Technology, Tsinghua University, Beijing, China
^{3} Jiangsu Collaborative Innovation Center for Language Ability,
Jiangsu Normal University, Xuzhou 221009 China

Abstract

Representation learning (RL) of knowledge graphs aims to project both entities and relations into a continuous low-dimensional space. Most methods concentrate on learning representations with knowledge triples indicating relations between entities. In fact, in most knowledge graphs there are usually concise descriptions for entities, which cannot be well utilized by existing methods. In this paper, we propose a novel RL method for knowledge graphs taking advantages of entity descriptions. More specifically, we explore two encoders, including continuous bag-of-words and deep convolutional neural models to encode semantics of entity descriptions. We further learn knowledge representations with both triples and descriptions. We evaluate our method on two tasks, including knowledge graph completion and entity classification. Experimental results on real-world datasets show that, our method outperforms other baselines on the two tasks, especially under the zero-shot setting, which indicates that our method is capable of building representations for novel entities according to their descriptions. The source code of this paper can be obtained from `https://github.com/xrb92/DKRL`.

Introduction

Knowledge graphs (KG) provide effective structured information and have been crucial resources for several intelligent applications including Web search (Szumlanski and Gomez 2010) and question answering. A typical KG usually describes knowledge as multi-relational data and represent as triple facts (*head entity*, `relation`, *tail entity*), also denoted as (h, r, t), indicating the relation between two entities.

Based on the symbolic representation of KGs with triples, people have to design various graph-based methods for KG applications. As KG size increases, these methods are becoming infeasible on large-scale KGs due to computation inefficiency and data sparsity. To address the challenge, representation learning (RL) for KGs has been proposed to embed KGs including both entities and relations into a continuous low-dimensional vector space (Dong et al. 2014) (embeddings). As a supplement to symbolic representation, the embeddings in latent space can significantly promote knowledge acquisition and inference (Bordes et al. 2013; Yang et al. 2014; Neelakantan, Roth, and McCallum 2015).

Most existing RL methods solely learn from fact triples of KGs (Bordes et al. 2013). In fact, in most KGs there are also concise descriptions for entities, with rich semantic information about these entities. For example, in Fig. 1 we show the descriptions of two entities in a fact triple sampled from Freebase, a large-scale KG maintained by Google.

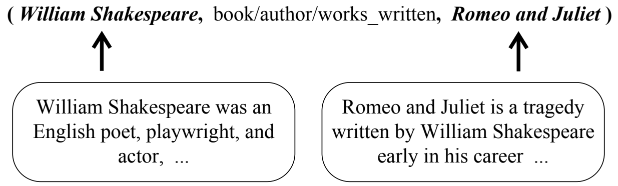
(William Shakespeare, book/author/works_written, Romeo and Juliet)
William Shakespeare was an English poet, playwright, and actor, ...
Romeo and Juliet is a tragedy written by William Shakespeare early in his career ...

Figure 1: Example of entity descriptions in Freebase.

It is non-trivial for existing RL methods of KGs to utilize entity descriptions. To address this problem, we propose a novel RL method for KGs, which is able to take advantages of both fact triples and entity description. We name the method as Description-Embodied Knowledge Representation Learning (DKRL). In the DKRL model, the embedding of an entity is responsible for both modeling the corresponding fact triples and modeling its description.

For fact triples, we follow a typical RL method TransE (Bordes et al. 2013) and regard the relation in each triple as a translation from head entity to tail entity. In this way, the entity and relation embeddings are learned to maximize the likelihood of these translations.

Meanwhile, given an entity we will also learn to maximize the likelihood of predicting its description. For this, we explore two encoders to represent semantics of entity descriptions, including continuous bag-of-words (CBOW) model and deep convolutional neural model. As compared with CBOW ignoring word orders in text, the convolutional model takes word orders, i.e., complicated local interactions of words in text, into consideration.

We evaluate the effectiveness of the DKRL model on two tasks, including knowledge graph completion and en-

[*] Corresponding author: J. Jia (jjia@tsinghua.edu.cn).

tity type classification. Experimental results on real-world datasets show that, the DKRL model consistently outperforms other baselines on the two tasks. Especially, we also consider the zero-shot scenario, where some entities are novel to existing KGs with only descriptions. Existing RL methods of KGs are incapable of those novel entities, since no embeddings have been learned for them. However, the DKRL model can build representations for those novel entities automatically from their descriptions. The experiments in zero-shot setting show that, the DKRL model can still achieve relatively favorable results on the two tasks. This indicates the good generalization ability and robustness of the DKRL model, which is particularly important for large-scale KGs and their applications in Web domain.

Related Work

Recent years there are a variety of methods modeling multi-relational data in knowledge graphs, many of which encode both entities and relations into a continuous low-dimensional vector space. TransE (Bordes et al. 2013) interprets the relations as translating operations between head and tail entities on the low-dimensional vector space. The energy function is defined as

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

which indicates that the tail embedding $\mathbf{t}$ should be the nearest neighbour of $\mathbf{h} + \mathbf{r}$. TransE performs well in 1-to-1 relations while has issues for modeling 1-to-N, N-to-1 and N-to-N relations. TransH (Wang et al. 2014b) attempts to solve the problem of TransE by modeling relations as hyperplanes and projecting $\mathbf{h}$ and $\mathbf{t}$ to the relational-specific hyperplane, allowing entities playing different roles in different relationships. TransR (Lin et al. 2015b) models entities and relations in distinct semantic space and projects entities from entity space to relation space when learning embeddings. PTransE (Lin et al. 2015a) proposes a multiple-step relation path-based representation learning model.

Most existing translation-based RL methods of KGs only concentrate on the structural information between entities, regardless of rich information encoded in entity descriptions. Moreover, because of the limitation of entity representations, these models are also not able to validate a triple when at least one of the entities is out of KGs. However, this situation can be handled with our DKRL model.

There are several methods using textual information to help KG representation learning. (Socher et al. 2013) proposes NTN and represents an entity as the average of its word embeddings in entity name, allowing the sharing of textual information located in similar entity names. (Wang et al. 2014a) combines entity embeddings with word embeddings into a joint continuous vector space by alignment models using entity names or Wikipedia anchors. (Zhong et al. 2015) extends the joint model and aligns knowledge and text embeddings by entity descriptions. These two works represent new entities using word embeddings of the corresponding entity names. (Zhang et al. 2015) represents entities with entity names or the average of word embeddings in descriptions. However, their use of descriptions neglects word orders, and the use of entity names struggles with ambiguity. Moreover, in practical zero-shot scenario, word embeddings of new entity names are usually missing in training data. Our model can directly build representations from descriptions to avoid such issues, not merely using entity descriptions as additional information.

Problem Formulation

We first introduce the notations used in this paper. Given a triple $(h, r, t) \in T$ while $h, t \in E$ stand for entities and $r \in R$ stands for relation. E is the set of entities and R is the set of relationships. T stands for the training set. Each entity and relation embedding takes values in $\mathbb{R}^k$.

Definition 1. Structure-based Representations: $\mathbf{h_s}$ and $\mathbf{t_s}$ are the structure-based representations for head and tail which can directly represent entities. This kind of representations is the same as those learned from existing translation-based models like TransE .

Definition 2. Description-based Representations: $\mathbf{h_d}$ and $\mathbf{t_d}$ are the description-based representations for head and tail which are built from entity descriptions. We will propose two encodes to construct this kind of representations in the following section.

Methodology

To utilize both fact triples and entity descriptions and be capable of dealing with zero-shot scenario, we propose two types of representations for entities, i.e., structure-based representations and description-based representations. Structure-based representations do better in capturing information in fact triples of KGs, while description-based representations do better in capturing textual information in entity descriptions. We learn the two entity representations simultaneously into the same continuous vector space, but do not force the representations to be unified for the consideration of better representing ability. The energy function of DKRL is then defined as

$$E = E_S + E_D, \tag{2}$$

where E_S is the energy function of structure-based representations, which shares the same formulation as TransE's in Equation 1, while E_D is the energy function of description-based representations. E_D can be defined by a variety of measurements. To make the learning process of E_D to be compatible with E_S ,we define E_D as follow:

$$E_D = E_{DD} + E_{DS} + E_{SD}, \tag{3}$$

where $E_{DD} = ||\mathbf{h_d} + \mathbf{r} - \mathbf{t_d}||$ in which head and tail are description-based representations. Also we have $E_{DS} = ||\mathbf{h_d} + \mathbf{r} - \mathbf{t_s}||$ and $E_{SD} = ||\mathbf{h_s} + \mathbf{r} - \mathbf{t_d}||$, in which one of $\mathbf{h}$ or $\mathbf{t}$ uses description-based representation and the other uses structure-based representation. The energy function will project the two types of entity representations into the same vector space with relation representations shared by all four energy functions, which will have mutual promotion between the two types of representations.

In this paper, we propose two encoders to build description-based representations in the following subsections. We first propose a continuous bag-of-words encoder for entity construction, then we propose a deep convolutional neural network encoder for a better understanding of textual information.

Continuous Bag-of-words Encoder

From each short description, we can generate a set of keywords which are usually capable of capturing the main ideas of entities. We assume that similar entities should have similar descriptions, and correspondingly have similar keywords. Those relations cannot be directly detected through structural information may be found in the internal contact of their keywords.

In the continuous bag-of-words encoder (CBOW), we select top n keywords in the description for each entity as the input (some classical textual features like TF-IDF could be used for ranking keywords). Then we simply sum up the embeddings of keywords to get the entity embedding ignoring word orders:

$$\mathbf{e_d} = \mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_k, \tag{4}$$

where $\mathbf{x}_i$ is the i-th word embedding belonging to the keyword set of entity e, and $\mathbf{e_d}$ will be used to minimize E_D. Fig. 2 shows the framework of the CBOW Encoder.

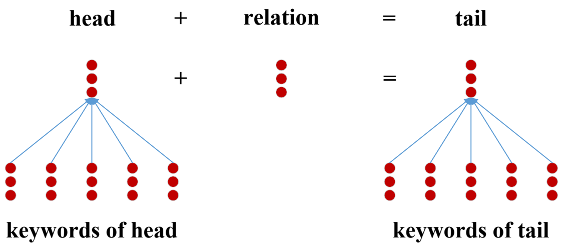

Figure 2: The CBOW Encoder

Convolutional Neural Network Encoder

Convolutional neural network (CNN) is an efficient model widely used on image and proven to be effective on some natural language processing tasks such as part-of-speech tagging, chunking, named entity recognition and semantic role labeling (Collobert et al. 2011). Recently CNN models are also proposed for relation classification (Zeng et al. 2014; dos Santos, Xiang, and Zhou 2015). Since CBOW has the shortage of ignoring the information of word orders and is easy to be influenced by the quality of keywords extraction, we propose a convolutional neural network encoder to further understand the descriptions and exploit the internal textual information hidden in word orders.

Overall Architecture Fig. 3 shows the overall architecture of the CNN Encoder. The CNN architecture has five layers, taking the whole description of a certain entity as the input after preprocessing, and output description-based representations of this entity. The entity embedding will then be learned to minimize the energy function of DKRL.

Preprocessing and Word Representation In preprocessing we first remove all stop words from raw texts, then we mark all phrases in descriptions (we simply select all entity names in training set as phrases) and consider those phrases to be words. Afterwards, each word is represented by a word embedding as the input of convolution layer. In our experiments, we use the word embeddings trained on Wikipedia by word2vec (Mikolov et al. 2013) as inputs for the CNN Encoder.

Convolution In convolution layer, we set $\mathbf{Z}^{(l)}$ to be the output of l-th convolution layer and $\mathbf{X}^{(l)}$ to be the input of l-th convolution layer. First a size k window will slide through the input vectors in $\mathbf{X}^{(l)}$ to get $\mathbf{X}'^{(l)}$. Specially in the first layer, $\mathbf{X}^{(1)}$ is preprocessed descriptions represented as a set of vectors $(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_n)$, and the window process has

$$\mathbf{x}'^{(1)}_i = \mathbf{x}_{i:i+k-1} = [\mathbf{x}_i^T, \mathbf{x}_{i+1}^T, \cdots, \mathbf{x}_{i+k-1}^T]^T, \tag{5}$$

where the i-th vector of $\mathbf{x}'^{(1)}$ is obtained by concatenating k column vectors in i-th window of input sentences. Due to the variable length of inputs when proceeding window process, we also add all-zero padding vectors at the end of every input vector. The i-th output vector of convolution layer will be:

$$\mathbf{z}_i^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{x}'^{(l)}_i + \mathbf{b}_i^{(l)}), \tag{6}$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{n_2^{(l)} \times n_1^{(l)}}$ is the convolution kernel for all input vectors of l-th convolution layer after window process and $\mathbf{b}_i^{(l)}$ is the optional bias. $n_2^{(l)}$ is the dimension of output vectors which could be considered as the number of feature maps. $n_1^{(l)} = k \times n_0^{(l)}$ where $n_0^{(l)}$ is the dimension of input vectors. σ is the activation function such as $\tanh$ or ReLU. Note that all zero-padding vectors should have no contribution in forward propagation nor be updated in back propagation. In this way we can align the variable length of input sentences while avoiding the possible side effects of all-zero paddings.

Pooling We use pooling after every convolution layer to shrink the parameter space of CNN and filter noises. Since articles are taken as our inputs, we propose different pooling strategies for different layers.

For the first pooling layer. we split the output vectors of the convolution layer with size n non-overlapped windows. In each window, we pick up the max value of every feature map to make up a new vector. The n-max-pooling is defined to determine the most significant feature values in each dimension of the input vectors within a size n window:

$$\mathbf{x}_i^{(2)} = \max(\mathbf{z}^{(1)}_{n\cdot i}, \cdots, \mathbf{z}^{(1)}_{n\cdot(i+1)-1}). \tag{7}$$

The n-max-pooling can shrink n times the size of feature representations, thus it will lower the complexity of CNN encoder and the cost of parameter learning.

However, some descriptions are so complicated that different sentences in a description may have different aspects of local information. Merely using max-pooling will lead to enormous information loss. In this case for the last pooling

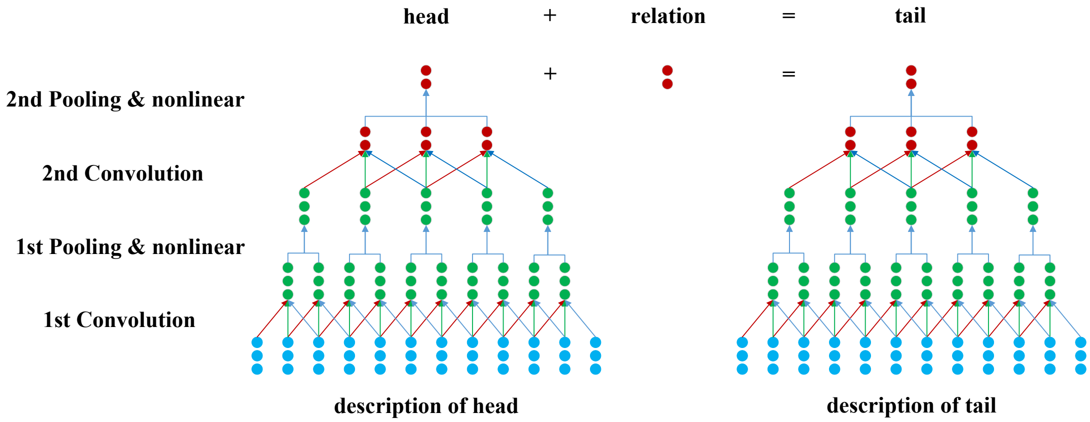

Figure 3: The Convolutional Neural Network Encoder

layer, we use mean-pooling instead of max-pooling before activation to build entity representations. We have

$$\mathbf{x}^{(3)} = \sum_{i=1,\cdots,m} \frac{\mathbf{z}_i^{(2)}}{m}, \tag{8}$$

that all m input vectors containing different local information should have contribution to the final entity embedding and can be updated during back propagation. Due to the different pooling strategies, we are capable of dealing with variable length inputs and get fixed-length representations for every entity without too much information loss.

Training

The DKRL model can be stated as a parameter set $\theta = (\mathbf{X}, \mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{E}, \mathbf{R})$ where $\mathbf{X}$, $\mathbf{E}$, $\mathbf{R}$ stand for the embeddings of words, entities and relations, and $\mathbf{W^{(1)}}$, $\mathbf{W^{(2)}}$ stand for the convolutional kernels in different layers. We minimize the following margin-based score function as objective for training:

$$\begin{aligned} L = \sum_{(h,r,t)\in T} \sum_{(h',r',t')\in T'} & \max(\gamma + d(h+r,t) \\ & -d(h'+r',t'), 0), \end{aligned} \tag{9}$$

where $\gamma > 0$ is a margin hyperparameter, $d(h+r,t)$ is the dissimilarity function between $\mathbf{h}+\mathbf{r}$ and $\mathbf{t}$. We test $L1$-norm and $L2$-norm and find that $L1$-norm performs better in our tasks. T' is the negative sampling set of T, we have

$$\begin{aligned} T' = \{(h', r, t) | h' \in E\} \cup \{(h, r, t') | t' \in E\} \\ \cup \{(h, r', t) | r' \in R\}, \end{aligned} \tag{10}$$

in which the head, tail or relation are randomly replaced by another entity or relation in a triple. Note that a triple will not be considered as a negative sample if it is already in T. Since there are two types of representations for both h and t, entities in the margin-based score function could either be structure-based representations or description-based representations.

Model Initialization The CBOW and CNN Encoders take plain texts as input and entity embeddings as output to minimize the score function stated above. $\mathbf{W^{(1)}}$, $\mathbf{W^{(2)}}$ are randomly initialized and $\mathbf{X}$ is pre-trained by Word2Vec learned on Wikipedia. $\mathbf{E}$ and $\mathbf{R}$ could either be initialized randomly or by the pre-trained embeddings with existing translation-based models such as TransE. For the consideration of efficiency, we also employ a multi-thread version of CNN to learn representations.

Optimization The optimization is a standard back propagation using stochastic gradient descent(SGD). The back propagation will be blocked when meets all-zero paddings or the current feature value was not considered in pooling during forward propagation. The chain rule is applied top-down through the DKRL model until the word embedding layer. The learning rate could be different for different combinations of entity representations.

Experiments

Datasets and Experiment Settings

Datasets In this paper, we adopt FB15K (Bordes et al. 2013), a dataset extracted from a typical large-scale KG Freebase (Bollacker et al. 2008), to evaluate the DKRL model on knowledge graph completion and entity classification. To confirm that every entity should have description for the description-based representation learning, we remove 47 entities from FB15K which have shorter than 3 words after preprocessing or even have no descriptions, and take away all triples containing those entities in FB15K. The average number of words in descriptions is 69 after preprocessing, and the longest description contains 343 words. The remaining training set has 472,860 triples and 1,341 relations, and test set has 57,803 triples.

For zero-shot learning, we build a new dataset FB20K which takes FB15K as the seed and shares the same relations. We select all entities in Freebase which have relations with entities in FB15K as candidates, then randomly select new entities from those candidates which have rich descriptions. We also extract all triples with relations in FB15K whose head or tail is a new entity and the other is in FB20K into the origin test set of FB15K. We split the test set into 4 types: both the head and the tail are in training set ($e - e$), the head is a new entity but the tail is not ($d - e$), the tail is a new entity but the head is not ($e - d$), both the head and the tail are new entities ($d - d$). FB20K shares the same training and validation set with FB15K. The statistics of datasets are listed in Table 1.

Table 1: Statistics of data sets

Dataset	#Rel	#Ent	#Train	#Valid	#Test
FB15K	1,341	14,904	472,860	48,991	57,803

Dataset	#Ent	$\#e - e$	$\#d - e$	$\#e - d$	$\#d - d$
FB20K	19,923	57,803	18,753	11,586	151

Parameter Settings We implement TransE, CBOW and CNN for comparison. We train those model with entity/relation dimension n in $\{$50, 80, 100$\}$. Following (Bordes et al. 2013), we use a fixed learning rate λ among $\{$0.0005, 0.001, 0.002$\}$, and margin γ among $\{$0.5, 1.0, 1.5, 2.0$\}$. For CBOW Encoder, we try different number of top N keywords and choose top 20 keywords to build entity embeddings for the best overall performances. For the CNN Encoder, we use 4-max-pooling for the first pooling layer and mean-pooling for the second pooling layer to achieve the best performance. We try different window size k among $\{$1, 2, 3$\}$ for different convolution layer. Also we set the dimension of word embedding n_w among $\{$50, 80, 100$\}$ and the dimension of feature map n_f among $\{$50, 100, 150$\}$. The optimal configurations of CNN are : $\lambda = 0.001$, $\gamma = 1.0$, $k = 2$, $n = 100$, $n_w = 100$, $n_f = 100$.

Knowledge Graph Completion

The task of knowledge graph completion aims to complete a triple (h, r, t) when one of h, t, r is missing based on minimizing the score function $S(h, r, t) = ||\mathbf{h} + \mathbf{r} - \mathbf{t}||$.

Evaluation Protocol We conduct our evaluation on FB15K and consider the knowledge graph completion task as two sub-tasks: entity prediction and relation prediction. Following (Bordes et al. 2013), we use two measures as our evaluation metrics: (1) mean rank of correct entities; (2) proportion of valid entities ranked in top 10 (for entity) or top 1 (for relation). We also follow the two evaluation settings named as ”Raw” and ”Filter”. We implement TransE as a baseline which performs better than results reported in (Bordes et al. 2013). In DKRL, both representations can validate triples separately. Since the results of structure-based representation learned in DKRL are similar to TransE, we report the results of CBOW and CNN which only use description-based representations. CNN+TransE is a union model of CNN and TransE which predict by weighting the representations of two models.

Results The results of entity prediction and relation prediction are in Table 2 and Table 3. From the results we observe that: (1) CNN+TransE significantly outperforms TransE and CBOW in mean rank and Hits@N on both entity and relation prediction, and CNN outperforms TransE and CBOW in mean rank on entity prediction, which indicate the robustness of CNN representations. The results show that the textual information in description, which has been successfully encoded in description-based representations by CNN, could provide a good supplement for RL of KGs. (2) Case study shows that simply using structural information may fail to capture details. For example, it’s hard to tell whether a soccer player is a goalkeeper or a forward if there is no explicit relation, but through some latent information embedded in words we may find the answer. (3) DKRL may not have huge advantages over TransE since structure-based representations already work well on this task. But DKRL will show its power in zero-shot scenario which cannot be handled by existing translation-based models.

Table 2: Evaluation results on entity prediction

Metric	Mean Rank		Hits@10(%)	
	Raw	Filter	Raw	Filter
TransE	210	119	48.5	66.1
DKRL(CBOW)	236	151	38.3	51.8
DKRL(CNN)	200	113	44.3	57.6
DKRL(CNN)+TransE	**181**	**91**	**49.6**	**67.4**

Table 3: Evaluation results on relation prediction

Metric	Mean Rank		Hits@1(%)	
	Raw	Filter	Raw	Filter
TransE	2.91	2.53	69.5	90.2
DKRL(CBOW)	2.85	2.51	65.3	82.7
DKRL(CNN)	2.91	2.55	**69.8**	89.0
DKRL(CNN)+TransE	**2.41**	**2.03**	**69.8**	**90.8**

Entity Classification

The task of entity classification is a multilabel classification task aiming to predict entity types, which is crucial and widely used in many NLP tasks (Neelakantan and Chang 2015). Almost every entity has types in Freebase (e.g. The entity *Washington County* has types including *locations/us_county* and *location/administrative_division*).

We extract all types of entities in FB15K from Freebase and get 4,054 types. We rank those types by their frequency and select top 50 types for classification (we remove the type of *common/topic* which almost all entities have). The top 50 types cover 13,445 entities. We randomly split them into training set and test set, the training set has 12,113 entities while the test set has 1,332 entities.

Evaluation Protocol In training, we use entity representations learned by TransE, CBOW and CNN trained on FB15K as features. We use Logistic Regression as classifier and one-versus-rest for multilabel classification. To make further comparison, we use a classical textual feature bag-of-words(BOW) as the baseline. For evaluation following (Neelakantan and Chang 2015), We use mean average precision (MAP) which is commonly used in multilabel classification as the evaluation method for entity classification.

Table 4: Evaluation results on entity classification

Metric	FB15K	FB20K
TransE	87.9	-
BOW	86.3	57.5
DKRL(CBOW)	89.3	52.0
DKRL(CNN)	**90.1**	**61.9**

Results From Table 4 we observe that CNN outperforms all other models in FB15K. It indicates that CNN features are more capable of catching entity type information and have better robustness. The reason is that, it is natural for CNN to encode both structural information in KGs and textual information in descriptions to get a better understanding of entities. CBOW also makes use of both information but performs weaker than CNN. However, BOW merely takes textual information into consideration, regardless of the relationships between entities, while TransE only focuses on the structural information, failing to encode the textual information embedded in descriptions.

Zero-shot Scenario

The tasks in zero-shot scenario focus on the situation when at least one of entities in test triples is out of KGs. All existing models based on structure-based representations cannot deal with this situation because they have no representations for entities which are out of KGs. However, the DKRL model is naturally capable of this situation.

We use FB20K to simulate a zero-shot scenario that all entities in FB15K are in-KG entities which can be learned through training, while 5,019 new-added entities are considered as out-of-KG entities which are built from their descriptions. As for entity classification, we use the same top 50 types in FB15K, put all 13,445 entities covered by those types in FB15K into training set and 4,050 out-of-KG entities into test set.

Knowledge Graph Completion in Zero-shot Scenario We consider CBOW as our baseline since all existing models using structure-based representations cannot represent new entities. We propose two evaluation methods for both encoders. In CBOW and CNN, all entities use description-based representations, while in Partial-CBOW and Partial-CNN, entities in training set use structure-based representations. The test set is split into 4 types: $e-e$, $d-e$, $e-d$ and $d-d$. In zero-shot scenario, we only focus on the latter three types which contain at least one out-of-KG entity. The results are shown in Table 5 and Table 6.

From Table 5 and Table 6 we observe that: (1) CNN significantly outperforms other models on all three types of test triples which achieves approximately 4.2% improvement on entity prediction and 7.9% improvement on relation prediction compared to CBOW. It indicates that even in zero-shot scenario, the DKRL model can still achieve relatively favorable results on knowledge graph completion. And the representations built with CNN have better performance than those built with CBOW. (2) The results of partial models are relatively good, which confirm that the two representations share the same vector space and could be learned into an unified one. However, There are still existing some incompatibility between the two representations which will weaken the performances, comparing to the corresponding models of CNN and CBOW which use description-based representations for all entities.

Table 5: Evaluation results on entity prediction in zero-shot scenario

Metric	$d-e$	$e-d$	$d-d$	Total
Partial-CBOW	26.5	20.9	67.2	24.6
CBOW	27.1	21.7	66.6	25.3
Partial-CNN	26.8	20.8	69.5	24.8
CNN	**31.2**	**26.1**	**72.5**	**29.5**

Table 6: Evaluation results on relation prediction in zero-shot scenario

Metric	$d-e$	$e-d$	$d-d$	Total
Partial-CBOW	49.0	42.2	0.0	46.2
CBOW	52.2	47.9	0.0	50.3
Partial-CNN	56.6	52.4	4.0	54.8
CNN	**60.4**	**55.5**	**7.3**	**58.2**

Entity Classification in Zero-shot Scenario For entity classification, we only test on CNN, CBOW and BOW since TransE has no representations for out-of-KG entities. From Table 4 we observe that CNN has the best performance which achieves 4.4% and 9.9% improvement compared with BOW and CBOW in FB20K. It indicates that description-based representations of CNN can function well for entity classification even though we cannot directly use structural information of triples in test set since they are out-of-KG.

Conclusion and Future Work

In this paper, we propose the DKRL model for representation learning of knowledge graphs with entity descriptions. We explore two encoders including continuous bag-of-words and deep convolutional neural network to extract semantics of entity descriptions. In experiments, we evaluate our model on two tasks including knowledge graph completion and entity classification. Experimental results show that our model achieves better performances than other baselines on both tasks especially in zero-shot scenario, which indicates the capability of building representations from entity descriptions.

We will explore the following research directions in future: (1) The DKRL model only consider entity descriptions for representation learning, while there are various of information like textual information of relations or entity types which could be added to our model. We may take advantages of those rich information in future. (2) We verify the effectiveness of description-based representations only with TransE, and it is not difficult for further explorations with more sophisticated extension models of TransE (e.g. TransH, TransR and PTransE). We will explore to extend DKRL to these models for a better understanding of knowledge graphs.

Acknowledgments

This work is supported by the 973 Program (No. 2014CB340501), the National Natural Science Foundation of China (NSFC No. 61133012 and 61572273) and the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office. We thank all anonymous reviewers for their constructive comments.

References

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of KDD*, 1247–1250.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, 2787–2795.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *JMLR* 12:2493–2537.

Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; and Zhang, W. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of KDD*, 601–610.

dos Santos, C.; Xiang, B.; and Zhou, B. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of ACL*, 626–634.

Lin, Y.; Liu, Z.; Luan, H.; Sun, M.; Rao, S.; and Liu, S. 2015a. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of EMNLP*, 705–714.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015b. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*.

Neelakantan, A., and Chang, M.-W. 2015. Inferring missing entity type instances for knowledge base completion: New dataset and methods. In *Proceedings of NAACL*.

Neelakantan, A.; Roth, B.; and McCallum, A. 2015. Compositional vector space models for knowledge base completion. *Proceedings of EMNLP*.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of NIPS*, 926–934.

Szumlanski, S., and Gomez, F. 2010. Automatically acquiring a semantic network of related concepts. In *Proceedings of CIKM*, 19–28.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014a. Knowledge graph and text jointly embedding. In *Proceedings of EMNLP*, 1591–1601.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014b. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*, 1112–1119.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2014. Embedding entities and relations for learning and inference in knowledge bases. *Proceedings of ICLR*.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, 2335–2344.

Zhang, D.; Yuan, B.; Wang, D.; and Liu, R. 2015. Joint semantic relevance learning with text data and graph knowledge. *ACL-IJCNLP 2015* 32–40.

Zhong, H.; Zhang, J.; Wang, Z.; Wan, H.; and Chen, Z. 2015. Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of EMNLP*, 267–272.