Knowledge Graph Representation Learning via Generated Descriptions

Miao Hu(✉), Zhiwei Lin, and Adele Marshall

School of Mathematics and Physics, Queen's University Belfast, Belfast, UK
{mhu05,z.lin,a.h.marshall}@qub.ac.uk

Abstract. Knowledge graph representation learning (KGRL) aims to project the entities and relations into a continuous low-dimensional knowledge graph space to be used for knowledge graph completion and detecting new triples. Using textual descriptions for entity representation learning has been a key topic. However, the current work has two major constraints: (1) some entities do not have any associated descriptions; (2) the associated descriptions are usually phrases, and they do not contain enough information. This paper presents a novel KGRL method for learning effective embeddings by generating meaningful descriptive sentences from entities' connections. The experiments using four public datasets and a new proposed dataset show that the **N**ew **D**escription-Embodied **K**nowledge **G**raph **E**mbedding (NDKGE for short) approach introduced in this paper outperforms most of the existing work in the task of link prediction. The code and datasets of this paper can be obtained from GitHub (https://github.com/MiaoHu-Pro/NDKGE.)

Keywords: Knowledge graph embedding · Entity description · Constructing new descriptions · Link prediction

1 Introduction

A *knowledge graph* $G = \{(h, r, t)|h, t \in E, r \in R\}$ [4,10] contains a set of nodes E for entities and a set of edges R for the relations between the entities. A *triple* (h, r, t) in a knowledge graph, $(h, r, t) \in G$, where $h, t \in E$ and a $r \in R$. Triple (h, r, t) is usually used to denote a fact where a head entity h has a relation of r with a tail entity t. For example, as shown in Fig. 1, ('Tom Cruise', '/film/producer/film', 'Mission: Impossible') is a triple where the head entity is 'Tom Cruise', the relation is 'film/producer/film', and the tail entity is 'Mission: Impossible'. Large scale knowledge graphs, such as FreeBase, have played key roles in supporting intelligent question answering, recommendation systems, and searches engines. However, most of them were built collaboratively by humans, where emerging relationships and entities may not be included. This is so-called *incompleteness* and *sparseness* of knowledge graph [10]. Thus, it is important to enrich knowledge graphs automatically to reduce those issues.

Knowledge graph representation learning (KGRL), also known as *knowledge graph embedding* (KGE), aims to automatically enrich knowledge graphs by representing entities and their relations into a continuous low-dimensional vector space so that the missing entities and relations can be inferred using those embeddings [4]. Two key tasks,

E. Métais et al. (Eds.): NLDB 2023, LNCS 13913, pp. 365–378, 2023.
https://doi.org/10.1007/978-3-031-35320-8_26

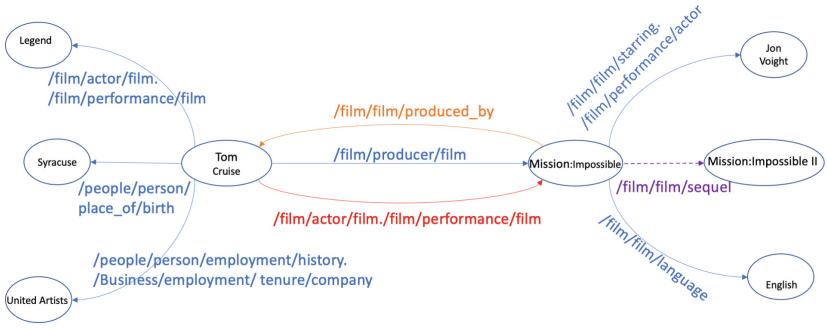

Fig. 1. An example of a sub-graph from the FB15K [4] dataset, where the nodes are entities, and the directed edges represent relationships between the entities. The head entity 'Tom Cruise' has two relations pointing at the tail entity 'Mission:Impossible' while the entity 'Mission:Impossible' is the head entity to the tail entity of 'Tom Cruise' with a relation of '/film/film/produced_by'. Most entities in the FB15K dataset contain a mention to explain the entity. For example, the entity 'Tom Cruise' has a mention of '`American actor and film producer`' and the entity 'Mission:Impossible' has a mention of '`1996 film directed by Brian De Palma`'.

link prediction and triple classification, have been proposed by [4] to consolidate a knowledge graph G, where both tasks are about making sure if a triple (h, r, t) exists in the knowledge graph, i.e., $(h, r, t) \in G$.

The early work, such as *translation-based models* [1,4,5,7,23], treats a triple (h, r, t) as a translation operation from head entity h to tail entity t via a relation r. Recently, researchers realised the importance of using textual information for learning effective embeddings [2,9,22]. These approaches use the associated descriptions for entities, or they extract relevant entity description information from external sources to help learning knowledge graph embedding. Although these methods have improved the performance in the link prediction task by using external information for the entities, they have the following key constraints:

1. the associated descriptions, also called *mention* are usually a phrase, which does not have enough meaningful information without enough context. For example, the mention for 'Tom Cruise' is '`American actor and film producer`' as show in Fig. 1. This does not provide enough detail to explain who 'Tom Cruise' is as it does not have any information regarding what films he had been involved in.
2. the associated description is not always available. For example, in the FB15K dataset, some entities do not have the associated descriptions;
3. the associated description obtained from external sources may not be accurate and may introduce noise into the training data.

To address these problems, this paper proposes a novel description-based KGE approach, known as New Description-Embodied Knowledge Graph Embedding (NDKGE for short), by creating a new description from their neighbours for each entity. The difference from all previous methods is: we use entities' neighbours to construct a

sentence-level description and then learn meaningful embeddings from the text. This NDKGE approach does not rely on external sources and it is believed that the generated description will help the algorithm to learn more meaningful and effective knowledge graph embeddings. The contributions of this paper include:

1. Sentence-level semantic description for entities generated by aggregating neighbourhood information;
2. A new data structure including an ID, name, mention, and a generated description introduced to represent an entity and a relation;
3. Experiments conducted to show that the sentence-level description is very useful for learning effective embeddings.

2 Related Work

This section presents key notation and related work in knowledge graph representation learning. For a triple $(h, r, t) \in G$, $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^n$ is used to denote their embeddings, respectively.

TransE [4] interprets each relation as a translating operation from a head entity to a tail entity, i.e., $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. The learning objective is to minimise the loss of the score function

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

for all the triples in G, where we take to be the L_1-norm. Studies have shown that the TransE performs well for 1-to-1 relations, but its performance drops significantly for 1-to-N, N-to-1, and N-to-N relations [28]. TransH [28] tries to solve the issues in the TransE by projecting h and t to the relationship-specific hyperplane, in order to allow entities to play different roles in different relationships. The PTransE [14] believes that multi-step relation paths contain rich inference patterns between entities. It considers relation paths as translations between entities. The TransE-EMM [17] introduced a neighbourhood mixture model for knowledge base completion by combining neighbour-based vector representations for entities. Compared with the TransE-EMM, our method relies on the generated entity descriptions to conduct embedding rather than computing entity representations directly based on neighbourhood entities and relations. The RotatE [23] treats the relation r as a rotating operation from h to t. The HAKE [35] models semantic hierarchies map entities into the polar coordinate system. It is inspired by the fact that concentric circles in the polar coordinate system can naturally reflect hierarchy. The BoxE [1] encodes relations as axis-aligned hyper-rectangles (or boxes) and entities as points in the d-dimensional euclidian space. The PairRE [7] uses two vectors for relation representation. These vectors project the corresponding head and tail entities to Euclidean space, where the distance between the projected vectors is minimized. The DualE [5] uses dual quaternion to unify translation and rotation in one model, where the new model can solve symmetry, antisymmetry, inversion, composition and multiple relations problems.

RESCAL [18] represents each relation as a full rank matrix and defines the score function as $f_{\mathbf{r}}(\mathbf{h}, \mathbf{t}) = \mathbf{h}^\top \mathbf{M}_{\mathbf{r}} \mathbf{t}$. As full rank matrices are prone to over-fitting, recent work turns to make additional assumptions on $\mathbf{M}_{\mathbf{r}}$. For example, DistMult [33] assumes

$\mathbf{M_r}$ to be a diagonal matrix, which also utilizes the multi-linear dot product as the scoring function. However, for general knowledge graphs, these simplified models are often less expressive and powerful. To better model asymmetric and inverse relations, DistMult was extended by introducing complex-valued embeddings, followed by the proposal of ComplEx [26]. The SimplE [11] uses the same diagonal constraint as DistMult. It models each fact in two forms (a direct and an inverse form). To represent such forms, It embeds each entity e in separate head and tail vectors e_h and e_t, and each relation r in individual direct and inverse vectors V_r and V_{-r}, which is fully expressive and can successfully model asymmetric relations. KGE-CL [32] proposed a simple yet efficient contrastive learning framework, which can capture the semantic similarity of the related entities and entity-relation couples in different triples, thus improving the expressiveness of embeddings.

The CNN-based approaches, such as ConvE [8] and ConvKB [16], improve the expressive power by increasing the interactions between entities and relations. CapsE [27] employs a capsule network to model the entries in the triple at the same dimension.

The Graph Convolutional Network-based methods (GCNs) were proposed to do embedding, such as R-GCN [20], which is the first to show that the GCNs can be applied to model relational data. This method aims to conduct the central node embedding by aggregating its neighbourhood information [12]. To explicitly and sufficiently model the Semantic Evidence into knowledge embedding, a new method SE-GNN [13] was proposed, where the three-level Semantic Evidence (entity level, relation level and triple-level) are modelled explicitly by the corresponding neighbour pattern and merged sufficiently by the multi-layer aggregation, which contributes to obtaining more extrapolative knowledge representation.

Text-based models take advantage of entity descriptions to help knowledge graph embedding. The majority of knowledge graphs include a brief entity description, called mention, for entities. Each mention, usually in a phrase, briefly explains its associated entity. Jointly (desp) [36] utilized an alternative alignment model that is not dependent on Wikipedia anchors and is based on text descriptions of entities. DKRL [30] employed two encoder methods, continuous Bag-of-words (CBOW) and convolutional neural network (CNN), to embed entity description and then to train models based on TransE. Jointly (LSTM) [31] used three encoder methods for joint knowledge graph embedding with structural and entity description and set the gating mechanism to integrate representations of structure and text into a unified architecture. ConMask [22] used the CNN attention mechanism to mark which words in the entity description are related to the relations and then generate target entity embedding. An et al. [2] proposed an accurate text-enhanced KG representation framework (AATE_E), which can utilize accurate textual information extracted from additional text to enhance the knowledge representations. Shah et al. [21] proposed an open-word detection framework, OWE, based on any pre-trained embedding model, such as TransE [4]. This framework aims to establish a mapping between entity descriptions and their pre-trained embeddings. Hu et al. [9] proposed to model the whole auxiliary text corpus with a graph and present an end-to-end text-graph enhanced KG embedding.

The above textual-based methods must satisfy a precondition, which is the entity descriptions, or available relevant texts. In other words, if the entity descriptions or

related text are missing or can not be obtained, these methods will be unable to perform knowledge graph embedding. This paper proposes a model, NDKGE, which is a textual-based model. NDKGE aims to solve the problem of unavailable descriptions and creates a high-quality description for entities.

3 Constructing New Entity Description for Knowledge Graph Embedding

This section presents a novel method to create descriptions for entities by aggregating the entity's neighbours' information in order to learn effective representations for entities and their relations.

3.1 Word-Level and Sentence-Level Semantics

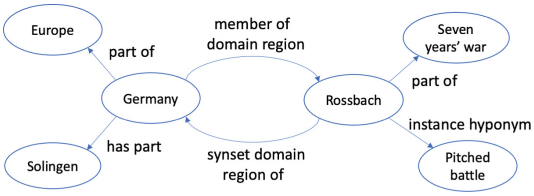

Fig. 2. A sub-graph of the WN18 dataset. The data use the format of (ID, entity name, mention). For example, the format for entity 'Germany' is (08766988, 'Germany', 'a republic in central Europe') and the format for entity 'Rossbach' is (01292928, 'Rossbach', 'a battle in the Seven Years' War (1757)').

The existing textual-based methods, such as SSP [29], AATE_E [2], and Teger-TransE [9] use the pre-defined descriptions which are associated with the entities in a knowledge graph. For example, in the WN18 dataset, the format for entity 'Germany' is (08766988, 'Germany', 'a republic in central Europe') as shown in Fig. 2, where 'a republic in central Europe' is the *mention* that is associated to 'Germany' and 08766988 is its unique ID.

However, not every entity has its associated description and the associated mention can be too brief to provide enough detail about the entity. With brief mentions or without any associated mentions, the performance could be compromised. As such, this paper aims to represent entities using more informative descriptions generated from their neighbours.

In this work, a new entity representation consists of four components (ID, Name, Mention, and Description), as shown in Fig. 3, where ID, Name, and Mention can be obtained from those existing knowledge graphs. Here, the Name will refer to either the actual entity or relation and the Mention is usually a phrase to interpret an entity (when the Name refers to an entity).

Fig. 3. The structure for representing entity or relation x. An entity may contain all four components but a relation will only contains (ID, Name), where name is the relation name.

The component of Description for each entity x is obtained by generating a set of sentences $D(x)$, where $D(x) = \{x \texttt{ has a relation of } r \texttt{ with } y : \forall(x, r, y) \in G\}$, from which k sentences are randomly picked ($k \leq |D(x)|$) and concatenated to construct a Description for entity x. For example, according to Fig. 2, if entity x is 'Germany', then $D(x)$={`Germany has a relation of part of with Europe.', `Germany has a relation of member of domain region with Europe.', `Germany has a relation of has part with Solingen.'}, which has three sentences. If $k = 3$, then the component of Description is generated by concatenating the three sentences. An entity x is represented with 4 components as shown in Fig. 3. The embedding for x shall also consider 4 components

$$\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}, \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^n$ for $1 \leq i \leq 4$, and $\mathbf{x}_1$ corresponds to the ID in Fig. 3. For $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_4$, as their corresponding components x_2, x_3, x_4 in Fig. 3 contain tokens/words, word embeddings are used to initiate $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_4$. Let $W \subseteq \mathbb{R}^n$ be the set of word embeddings and suppose x_i ($2 \leq i \leq 4$) contains n tokens $(x_{i_1}, \ldots, x_{i_n})$, then

$$\mathbf{x}_i = \frac{1}{n} \sum_{j=1}^{n} \mathbf{x}_{i_j} \tag{3}$$

where $\mathbf{x}_{i_j} \in W$ is the embedding for token x_{i_j}.

Algorithm 1 shows a function to calculate the embeddings $(\mathbf{h}, \mathbf{r}, \mathbf{t})$ for a triple $(h, r, t) \in G$, where the representation of entities ($\mathbf{h}$ or $\mathbf{t}$) and relations ($\mathbf{r}$) will be calculated according to different settings such as '*mention*' and '*description*'. For example, $\mathbf{h}$ =representation(h,'*description*') and $\mathbf{t}$ =representation(t,'*description*') will use the contextual information in the Description that are generated as shown in Fig. 3. As a relation does not have a Mention and no Description is generated, $\mathbf{r}$ is obtained using $\mathbf{r}$ =representation(r,'*name*'). The entities h or t, and relations r denote $\mathbf{x}$ represented by the Eq. (2). After $(\mathbf{h}, \mathbf{r}, \mathbf{t})$ is obtained, the TransE score function Eq. (1) is used for optimizing $(\mathbf{h}, \mathbf{r}, \mathbf{t})$.

3.2 Training

Our model uses the vectors constructed above as input. The embedding of the entities and relations is obtained after the model training is completed. We use the max-margin criterion [4] for training, and define the following loss function to optimize the model:

Algorithm 1: Generating initial embedding vectors for entities or relations x

Data: A knowledge graph G; A set of word embeddings W
Input: Entity or relation x (represented with Eq. (2)), method y for combining embedding vectors
Result: Embedding vector $\mathbf{v}_x$ for input x
Function `representation`(x, y)**:**
  Initialize $\mathbf{x}_1$ by a random vector or pre-trained embeddings [16];
  Calculate $\mathbf{x}_2$ using Eq. (3) ;
  if $y =$*'name'* **then**
    $\mathbf{x}_2 = \mathbf{x}_2$;
  end
  if $y =$*'mention'* **then**
    $\mathbf{x}_2 = \mathbf{x}_2 + \mathbf{x}_3$;
  end
  else if $y =$*'description'* **then**
    $\mathbf{x}_2 = \mathbf{x}_2 + \mathbf{x}_3 + \mathbf{x}_4$;
  end
  return $\mathbf{v}_x = \mathbf{x}_1 \oplus \mathbf{x}_2$ (The $\oplus$ denotes that two vectors are concatenated.);

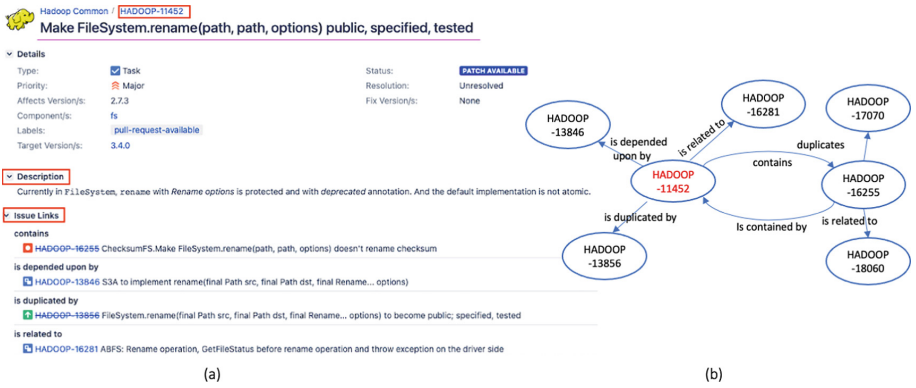

Fig. 4. A summary of the issues in Issue Tracker System. For the given issue, HADOOP-11452, has several attributes, such as Type, Status, and Priority, as shown in (a). On the other hand, the description (known as Mention in this work) and Issue Links were provided. And these Issue Links can be represented to a graph as shown in (b). In this paper, the links between issues are known as relations, and the issues are known as entities.

$$L = \sum_{(h,r,t)\in G} \sum_{(h',r,t')\in G'} \max(\gamma + f_{\mathbf{r}}(\mathbf{h},\mathbf{t}) - f_{\mathbf{r}}(\mathbf{h}',\mathbf{t}'), \mathbf{0}) \tag{4}$$

where (h', r, t') is the negative triple, and γ is a hyper-parameter representing the max-margin between positive triples scores and negative triples scores. G' is the negative

triple set generated by positive triples G with head or tail randomly replaced by another entity. Most importantly, the head and tail can not be replaced at the same time [4].

$$G' = \{(h', r, t) \,|h' \in E\} \cup \{(h, r, t') \,|t' \in E\} \tag{5}$$

In the training process, our model needs to learn the parameter set $\theta = \{\mathbf{E}, \mathbf{R}\}$ where $\mathbf{E} = \{\mathbf{h}, \mathbf{t}|\forall(h, r, t) \in G\}$, $\mathbf{R} = \{\mathbf{r}|\forall r \in R\}$ stand for the embeddings for entities and relations.

4 Experiments

4.1 Datasets

In this paper, four commonly used datasets, FB15K [4], WN18 [29], FB15K237 [8], WN18RR [8], and a new dataset Hadoop16K proposed by this work are used to evaluate NDKGE model on link prediction task. FB15K and WN18 are extracted from the FreeBase[1] and WordNet[2] respectively. FB15K237 and WN18RR were considered as challenging datasets, which is a subset of FB15K and WN18 where inverse relations are removed.

We collect the Hadoop16K from a popular Issue Tracking System[3] that is used to manage and track issues [15]. For an example of a given issue, HADOOP-11452, as shown in Fig. 4 (a), its details and Issue Links can be obtained. The Issue Links represent the relationships between this issue and other issues, such as 'contains', 'is related to', and 'is duplicated by', and we can show that with a graph, as shown in Fig. 4 (b). In practice, we found that a lot of links/relations between issues are missing, and these missing links should be included immediately to facilitate the orderly progress and maintenance of software development. Table 1 illustrates the number of entities and relations about the datasets.

Table 1. Summary of datasets.

Dataset	#Rel	#Ent	#Train	#Valid	#Test
FB15K	1345	14951	483142	50000	59071
WN18	18	40943	141442	5000	5000
FB15K237	237	14541	272115	17535	20466
WN18RR	11	40943	86835	3034	3134
Hadoop16K	31	12249	15791	1974	1974

4.2 Parameter Settings

The experiments use different margin γ from {0.5, 1, 3, 5} and the learning rate λ is set among {0.01, 0.05, 0.5, 1}. Also, we set the dimension of ID embedding $\mathbf{x}_1$ in

[1] www.freebase.com.
[2] https://wordnet.princeton.edu/.
[3] https://issues.apache.org/.

Algorithm 1 among {20, 50, 100, 200}, and the dimension of textual embedding $\mathbf{x}_2$ among {50, 100, 200, 300}. The number k of neighbours for generating descriptions is $|D(x)|$. The measure of dissimilarity is L_1 distance. At the same time, the experiment conducts a setting $description$ for using Name, Mention and generated sentence-level Description.

4.3 Link Prediction

Link prediction aims to complete a triple (h, r, t) with h or t missing. For example, to predict t given an in-complete triple $(h, r, ?)$ or predict h given $(?, r, t)$. We use two evaluation metrics in accordance with [4]: (1) the mean rank of correct entities; (2) the proportion of valid entities in the top 10 for the entity. In addition, we use the evaluation settings "Filter" [4,28]. Tables 2, 3 show the results of entity prediction.

As illustrated in Table 2, compared with *translation-based models* such as RotatE [23], PairRE [7], and DualE [5] that only encode entity/relation ID, our method can achieve high performances by using not only entity/relation ID but also textual information (entity name, mention and description). This indicates that related textual information is very helpful for effective knowledge graph embeddings. Also, we observe that our method is better than other text-based method, such as ConMask [22], and Teger-TransE [9], this indicates that the newly constructed entity description is reasonable and better than the original text. For WN18 dataset, our method achieves the best performance in Mean Rank (MR) and Hits@10 compared with all baselines. It even also surpasses the latest method such as PairRE [7] and DualE [5] in Hits@10.

Table 3 shows that our model NDKGE significantly outperforms the state-of-the-art models on the WN18RR. Our NDKGE with the `description` setting can obtain 0.699 for Hits@10, which is 10% higher than the state-of-the-art RESCAL-CL [32] to obtain 0.597. Also, our method achieved comparable performance to the benchmark models on the FB15K237, less than the latest method such as DualE [5], and ComplEx-CL [32]. The main reason could be that the FB15K237 is significantly density: 1) The multi-relationships between entities are common: for example, multi-relational facts (that is, N-to-N relations type) account for more than 70% in the test set [25]; 2) According to statistics of FB15K237, the average number of neighbours for entities is 18.8, and the maximum number of neighbours for entities is 1325, which is denser than WN18RR. The latter has the average number of neighbours at 2.1 and the maximum number of neighbours at 462. As a result, our NDKGE achieves higher performance on WN18RR than FB15K237.

On Hadoop16K, our method achieves the best performance in MR and Hits@10 compared with other state-of-the-art benchmarks. Compared with FB15K237, the Hadoop16K has a sparse structure. For example, the statistics of the test set found that the proportions of N-1, 1-N and N-N relation types were 11.5%, 10.8% and 0%, respectively. At the same time, counting the number of neighbours of entities, we found the maximum number of neighbours of its entities is 84, and the average number of neighbours is 1.2, much smaller than 1325 and 18.8 in FB15K237.

From all the results on the five datasets we report above, we find that connecting newly created sentence descriptions can obtain good experimental results, which

Table 2. Results of link prediction on FB15K and WN18.

Datasets	FB15K		WN18	
Metric	Mean Rank	Hits@10	Mean Rank	Hits@10
TransE [4]	119	0.661	280	0.899
TransH [28]	87	0.644	303	0.867
Jointly(desp) [36]	39	0.773	-	-
DKRL(CNN) [30]	91	0.674	-	-
Jointly(A-LSTM) [31]	73	0.755	123	0.909
SSP(Joint) [29]	82	0.790	156	0.932
AATE_E [2]	76	0.761	123	0.941
ConMask [22]	98	0.620	-	-
RotatE [23]	40	0.884	309	0.959
RPJE [19]	40	**0.903**	-	0.951
Teger-TransE [9]	72	0.763	168	0.947
PairRE [7]	37	0.896	-	-
DualE [5]	**21**	0.896	156	0.962
NDKGE	45	0.842	**13**	**0.976**

Table 3. Results of link prediction on FB15K237 and WN18RR.

Datasets	WN18RR		FB15K237		Hadoop16K	
Metric	Mean Rank	Hits@10	Mean Rank	Hits@10	Mean Rank	Hits@10
TransE [4]	3526	0.477	234	0.480	401	0.738
TransH [28]	6356	0.350	334	0.395	559	0.823
DistMult [33]	7000	0.504	512	0.446	530	0.586
ComplEx [26]	7882	0.530	546	0.450	555	0.793
R-GCN [20]	6700	0.207	600	0.300	-	-
ConvE [8]	4464	0.531	245	0.497	-	-
ConvKB [16]	3433	0.524	309	0.421	282	0.855
QuatE [34]	3472	0.564	176	0.495	-	-
RotatE [23]	3340	0.571	177	0.533	385	0.859
TuckER [3]	-	0.526	-	0.544	-	-
HAKE [35]	-	0.582	-	0.545	-	-
GC-OTE [24]	-	0.583	-	0.550	-	-
ATTH [6]	-	0.573	-	0.540	-	-
BoxE [1]	3117	0.523	163	0.538	481	0.851
PairRE [7]	-	-	160	0.544	379	0.850
DualE [5]	2270	0.492	**91**	0.559	1144	0.854
SE-GNN [13]	3211	0.572	157	0.549	-	-
RESCAL-CL [32]	-	0.597	-	0.554	-	0.812
ComplEx-CL [32]	-	0.595	-	**0.564**	-	0.882
NDKGE	**166**	**0.699**	187	0.547	**219**	**0.900**

Table 4. Ablation study for WN18, FB15K, WN18RR, and FB15K237.

Datasets	WN18		FB15K		WN18RR		FB15K237	
Metric	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10
NDKGE($name$)	158	0.848	330	0.558	1530	0.486	273	0.449
NDKGE($mention$)	40	0.948	76	0.725	307	0.621	263	0.516
NDKGE($description$)	13	0.976	45	0.842	166	0.699	187	0.547

Table 5. Ablation study for Hadoop16K.

Datasets	Hadoop16K	
Metric	MR	Hits@1
NDKGE($name$)	289	0.744
NDKGE($mention$)	275	0.766
NDKGE($description$)	219	0.778

means that the sentence-level description can provide the model with richer semantic information and help to learn more effective knowledge embeddings for application tasks.

4.4 Ablation Study

Algorithm 1 shows a function to calculate the embeddings $(\mathbf{h}, \mathbf{r}, \mathbf{t})$ for a triple $(h, r, t) \in G$, where we conduct ablation study on five datasets using three different settings: $name$, $mention$, and $description$. Table 4 shows the result of ablation study on WN18, FB15K, WN18RR and FB15K237. For WN18, The MR is reduced from 158 to 13 and Hits@10 rises from 0.848 to 0.976 when using the $name$ and $description$ settings, respectively. For FB15K, in Hits@10, using the $description$ setting to obtain 0.842 is 28.4% higher than using $name$ setting to obtain 0.558, and 11.7% higher than using $mention$ setting to obtain 0.725. The MR is reduced from 1530 to 166, Hits@10 increases from 0.486 to 0.699 in WN18RR and Hits@10 achieves 0.699 using $description$ setting and 21.3% higher than using $name$ setting to obtain 0.486, and 7.8% higher than using the $mention$ setting to obtain 62.1. For FB15K237, Hits@10 increases from 0.449 to 0.547 by using $name$ and $description$ settings, respectively.

Table 5 shows the ablation study result on Hadoop16K. Statistics show that about 45% of entities have only one link, so getting higher Hits@1 makes more sense in industrial practice. We report the results of MR and Hits@1 under the three settings, $name$, $mention$, and $description$. With the addition of the newly created description, MR decreases from 289 to 219, and Hits@1 rises from 0.744 to 0.778. The ablation study on five datasets shows that the link prediction performance increases with newly created sentence-level descriptions, which means adding new descriptions to help knowledge graph embedding is meaningful in practice.

5 Conclusion and Future Work

This paper introduces the NDKGE approach, which uses neighbour information to create a description for an entity. The method helps to address the issue in the existing text-based methods where some entities may not have their associated mentions or the related text description can not be obtained from external sources. We conduct the link prediction task on five datasets, FB15K, FB15K237, WN18, WN18RR, and Hadoop16K. The experimental results show that the knowledge graph embeddings with the generated descriptions can outperform the existing work when each entity has fewer relations with other entities, such as in the WN18RR and Hadoop16K. This paper only focused on using the score function from TransE, which already shows promising results. We will consider the other score functions in our future work. Future work will also focus on extending the generated description for detecting unknown entities that are introduced from out-of-the KG.

References

1. Abboud, R., Ceylan, İ.İ., Lukasiewicz, T., Salvatori, T.: Boxe: A box embedding model for knowledge base completion. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, pp. 9649–9661 (2020)
2. An, B., Chen, B., Han, X., Sun, L.: Accurate text-enhanced knowledge graph representation learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, Louisiana, USA, vol. 1 (Long Papers), pp. 745–755 (2018)
3. Balazevic, I., Allen, C., Hospedales, T.M.: Tucker: tensor factorization for knowledge graph completion. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Hong Kong, China, pp. 5184–5193 (2019)
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS), pp. 1–9 (2013)
5. Cao, Z., Xu, Q., Yang, Z., Cao, X., Huang, Q.: Dual quaternion knowledge graph embeddings. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, Virtual Event, pp. 6894–6902 (2021)
6. Chami, I., Wolf, A., Juan, D., Sala, F., Ravi, S., Ré, C.: Low-dimensional hyperbolic knowledge graph embeddings. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, pp. 6901–6914 (2020)
7. Chao, L., He, J., Wang, T., Chu, W.: Pairre: knowledge graph embeddings via paired relation vectors. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, (vol. 1: Long Papers), Virtual Event, pp. 4360–4369 (2021)
8. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, pp. 1811–1818 (2018)
9. Hu, L., et al.: Text-graph enhanced knowledge graph representation learning. Front. Artif. Intell. **4**, 118–127 (2021)
10. Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S.Y.: A survey on knowledge graphs: representation, acquisition, and applications. IEEE Trans. Neural Netw. Learn. Syst. **33**, 1–21 (2021)

11. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. In: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, Montréal, Canada, pp. 4289–4300 (2018)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, Toulon, France, Conference Track Proceedings, pp. 1–14 (2017)
13. Li, R., et al.: How does knowledge graph embedding extrapolate to unseen data: a semantic evidence view. In: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI, pp. 5781–5791 (2022)
14. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 705–714 (2015)
15. Montgomery, L., Lüders, C.M., Maalej, W.: An alternative issue tracking dataset of public jira repositories. In: IEEE/ACM 19th International Conference on Mining Software Repositories, Pittsburgh, PA, USA, pp. 73–77 (2022)
16. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.Q.: A novel embedding model for knowledge base completion based on convolutional neural network. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 2 (Short Papers), pp. 327–333 (2018)
17. Nguyen, D.Q., Sirts, K., Qu, L., Johnson, M.: Neighborhood mixture model for knowledge base completion. In: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, pp. 40–50 (2016)
18. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning, Bellevue, Washington, USA, pp. 809–816 (2011)
19. Niu, G., et al.: Rule-guided compositional representation learning on knowledge graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 2950–2958 (2020)
20. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European semantic web conference, pp. 593–607 (2018)
21. Shah, H., Villmow, J., Ulges, A., Schwanecke, U., Shafait, F.: An open-world extension to knowledge graph completion models. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3044–3051 (2019)
22. Shi, B., Weninger, T.: Open-world knowledge graph completion. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, pp. 1957–1964 (2018)
23. Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: knowledge graph embedding by relational rotation in complex space. In: 7th International Conference on Learning Representations, New Orleans, LA, USA, pp. 1–18 (2019)
24. Tang, Y., Huang, J., Wang, G., He, X., Zhou, B.: Orthogonal relation transforms with graph context modeling for knowledge graph embedding. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, pp. 2713–2722 (2020)
25. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, pp. 57–66 (2015)
26. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning, pp. 2071–2080 (2016)

27. Vu, T., Nguyen, T.D., Nguyen, D.Q., Phung, D., et al.: A capsule network-based embedding model for knowledge graph completion and search personalization. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), pp. 2180–2189 (2019)
28. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, Québec, Canada, pp. 1112–1119 (2014)
29. Xiao, H., Huang, M., Meng, L., Zhu, X.: SSP: semantic space projection for knowledge graph embedding with text descriptions. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA, pp. 3104–3110 (2017)
30. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA, pp. 2659–2665 (2016)
31. Xu, J., Qiu, X., Chen, K., Huang, X.: Knowledge graph representation with jointly structural and textual encoding. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, pp. 1318–1324 (2017)
32. Xu, W., Luo, Z., Liu, W., Bian, J., Yin, J., Liu, T.: KGE-CL: contrastive learning of knowledge graph embeddings, pp. 1–14 (2022)
33. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: 3rd International Conference on Learning Representations, San Diego, CA, USA, Conference Track Proceedings, pp. 1–12 (2015)
34. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, pp. 2731–2741 (2019)
35. Zhang, Z., Cai, J., Zhang, Y., Wang, J.: Learning hierarchy-aware knowledge graph embeddings for link prediction. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, pp. 3065–3072 (2020)
36. Zhong, H., Zhang, J., Wang, Z., Wan, H., Chen, Z.: Aligning knowledge and text embeddings by entity descriptions. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 267–272 (2015)