IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 12, 2017, accepted September 27, 2017, date of publication October 4, 2017, date of current version October 25, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2759139

Knowledge Graph Embedding by Dynamic Translation

LIANG CHANG[1], MANLI ZHU[1], TIANLONG GU[2], CHENZHONG BIN[2], JUNYAN QIAN[1], AND JI ZHANG[3], (Senior Member, IEEE)

[1]Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China
[2]Guangxi Experiment Center of Information Science, Guilin University of Electronic Technology, Guilin 541004, China
[3]Faculty of Health, Engineering and Sciences, University of Southern Queensland, Toowoomba, QLD 4350, Australia

Corresponding author: Liang Chang (changl@guet.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61363030, Grant 61572146, and Grant U1501252, in part by the Natural Science Foundation of Guangxi Province under Grant 2015GXNSFAA139285 and Grant 2016GXNSFDA380006, in part by the Innovation Project of GUET Graduate Education under Grant 2016YJCX13, and in part by the High Level of Innovation Team of Colleges and Universities in Guangxi and Outstanding Scholars Program.

ABSTRACT Knowledge graph embedding aims at representing entities and relations in a knowledge graph as dense, low-dimensional and real-valued vectors. It can efficiently measure semantic correlations of entities and relations in knowledge graphs, and improve the performance of knowledge acquisition, fusion and inference. Among various embedding models appeared in recent years, the translation-based models such as TransE, TransH, TransR and TranSparse achieve state-of-the-art performance. However, the translation principle applied in these models is too strict and can not deal with complex entities and relations very well. In this paper, by introducing parameter vectors into the translation principle which treats each relation as a translation from the head entity to the tail entity, we propose a novel dynamic translation principle which supports flexible translation between the embeddings of entities and relations. We use this principle to improve the TransE, TransR and TranSparse models respectively and build new models named TransE-DT, TransR-DT and TranSparse-DT correspondingly. Experimental results show that our dynamic translation principle achieves great improvement in both the link prediction task and the triple classification task.

INDEX TERMS Dynamic translation, embeddings, knowledge graph, translation-based models.

I. INTRODUCTION

Knowledge graph is one of the most popular approaches for representing knowledge on the current Web. A typical knowledge graph often describes the knowledge as multi-relational data and stores factual information in the form of triple (*head entity, relation, tail entity*) ((h, r, t) for short), where *head* and *tail* are entities and *relation* represents the relationship between the two entities, e.g., (*Bill Gates, Founder, Microsoft*). Knowledge graph is an important basic technology to promote the development of artificial intelligence and support the application of intelligent information services, such as question answering [1], web search [2], [3], and information extraction [4].

With the advent of big data era, various large-scale knowledge graphs such as WordNet [5], Yago [6], Freebase [7] and NELL [8] are available. There are lots of information and knowledge contained in knowledge graphs. However, due to the large scale and rapid increase of knowledge graphs, it is almost impossible to make full use of the knowledge by traditional logic-based methods [9], [10]. Recently, embedding-based approaches have shown strong feasibility and robustness. The basic idea of these approaches is to project (or embed) entities and relations in knowledge graphs into a continuous, real-valued and low-dimensional vector space, and then make use of the knowledge contained in knowledge graphs by efficient numerical calculations on the vector space. This kind of approaches performed well in dealing with the data sparseness problem. It achieved promising results in various tasks such as knowledge graph completion [11], classification [12], entity resolution [13], [14] and relation extraction [15]–[17].

Among various embedding-based models, translation-based models achieve the state-of-the-art performance. The most typical translation-based model is TransE [18] which proposed by Bordes *et al.* in 2013. In TransE, relationships between entities are treated as translations in the embedding space: if (h, r, t) holds, then the embedding of the tail entity t should be close to the embedding of the head entity h plus the embedding of the relationship r. In another word, TransE only relies on a reduced set of parameters, by learning a low-dimensional vector for each entity and each relationship in such a way that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ for every triple (h, r, t) in the knowledge graph, where the letters in boldface denote the vector embeddings of the corresponding entities or relationship. TransE achieves significant performance on the benchmark tasks of link prediction and triple classification in general cases. However, it is not good at dealing with complex relations which are reflexive, transitive, 1-to-N, N-to-1, or N-to-N. To solve this problem, various improved models were proposed in the past three years.

The TransH model [19] proposed by Wang *et al.* treats a relation as a translating operation on a hyperplane. It sets two vectors for each relation: a relation-specific hyperplane $\mathbf{w}_r$ (also called norm vector), and a relation-specific translation vector $\mathbf{d}_r$. For each triple (h, r, t), the embeddings of h and t are firstly projected to the hyperplane of r (i.e. $\mathbf{w}_r$) respectively, then these two projections are connected by the relation-specific translation vector $\mathbf{d}_r$ on $\mathbf{w}_r$. Since TransH makes each entity to have distinct representations in different relations, it achieves better performance on complex relations compared with TransE.

Lin *et al.* think that entities and relations are different types of objects, and it is not enough to represent them in the same space as done by the TransE model and the TransH model. Based on this idea, they proposed the TransR/CTransR model [20] which models entities and relations in different semantic spaces. The TransR/CTransR model first projects entity vectors from entity space into the corresponding relation space, and then builds translating operations between projected entities. TransR achieves significant improvements on complex relations compared to TransE and TransH.

Ji *et al.* think that the types and attributes of entities linked by the same relation are various and therefore it is not good to let all entities share the same mapping parameters as done by the TransR/CTransR model. They proposed the TransD model [21] which uses two distinct projection matrices to respectively project the head entity and the tail entity into the relation space. In another work, they replaced the dense matrices used in TransR by sparse matrices and proposed a model named TranSparse [22].

Feng *et al.* think that the translation principle $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ used in the above models is too strict to model complex and diverse entities and relations. They proposed the FT model [23] which uses a new translation principle $\mathbf{h} + \mathbf{r} \approx \alpha\mathbf{t}$. This new translation principle is good at dealing with 1-to-N relations, since it only requires the embeddings of multiple tail entities to be at the same direction rather than be equal vectors.

In this paper, we think that the translation principle $\mathbf{h} + \mathbf{r} \approx \alpha\mathbf{t}$ is still too strict to deal with complex relations and diverse entities well. For dealing with complex relations, instead of requiring the embeddings of multiple relations to be equal vectors or at the same direction, we only require them to be at the same plane. At the same time, in order to deal with diverse entities, we relax the constraints on the embeddings of multiple head entities or tail entities, by permitting them to be at the same plane rather than be equal vectors or be at the same direction. By unifying these two ideas in a framework, we propose a new translation principle named Dynamic Translation (or DT for short). We use this dynamic translation principle to improve TransE, TransR and TranSparse respectively, and present new models TransE-DT, TransR-DT and TranSparse-DT correspondingly. Experimental results show that our dynamic translation principle achieves great improvement.

The remainder of this paper is organized as follows. A formal description of related models is presented in Section 2. In Section 3, we first introduce the dynamic translation principle, and then apply it to improve TransE, TransR and TranSparse respectively. In section 4, detailed experimental results and comparisons are presented. Section 5 concludes the paper.

II. RELATED WORK

We first describe some common notations. We denote a triple by (h, r, t) and their column vectors by bold lower case letters $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$. Score function is represented by $f_r(h, t)$. We will describe other notations in the appropriate sections.

A. TRANSLATION-BASED MODELS

TransE: As shown in Fig. 1, basic idea of TransE [18] is to calculate vector embeddings for all entities and relations in a single space, with the target that $\mathbf{h}+\mathbf{r}=\mathbf{t}$ holds for each triple (h, r, t) contained in knowledge graph. Hence, score function used for training the vector embeddings is

$$f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|^2_{\ell_{1/2}}. \tag{1}$$

TransE is not good at dealing with complex relations. For example, suppose r is a 1-to-N relation, and there are many triples $(h, r, t_1), \ldots, (h, r, t_n)$. By TransE, we might get the same embeddings for different entities $t_1, \ldots, t_n$.

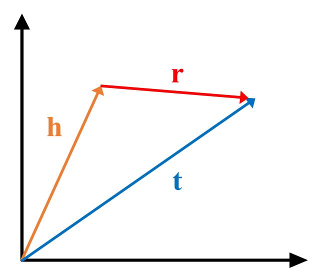


FIGURE 1. Simple illustration of TransE.

TransH: TransH [19] makes an improvement to TransE by enabling an entity to have distinct representations when the entity is involved in different relations. As illustrated in Fig. 2, for each triple (h, r, t), TransH further projects the embeddings $\mathbf{h}$ and $\mathbf{t}$ to a relation-specific hyperplane by a normal vector $\mathbf{w}_r$, and correspondingly gets the projected

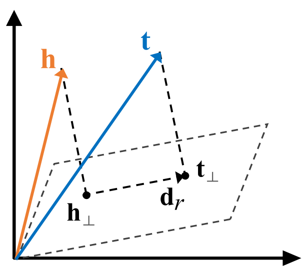


FIGURE 2. Simple illustration of TransH.

vectors $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^\top \mathbf{h} \mathbf{w}_r$ and $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^\top \mathbf{t} \mathbf{w}_r$. Then, score function used for training is

$$f_r(h,t) = \|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_{\ell_{1/2}}^2. \tag{2}$$

TransR: TransR/CTransR [20] first extends the single vector space used in TransE and TransH to many vector spaces: an entity space for embedding all entities, and a group of relation spaces that each space is used for a distinct relation. Then, for each relation r, it constructs a projection matrix $\mathbf{M}_r$ and by which projects the embeddings of entities from the entity space to the relation space of r. The basic idea of TransR is illustrated in Fig. 3, where circles denote entities, and the surrounding triangles denote the entities which is similar to them. For a given triple (h, r, t), the goal of TransR is $\mathbf{M}_r\mathbf{h} + \mathbf{r} = \mathbf{M}_r\mathbf{t}$ with the ideal embedding in the relation-specific space. The score function is

$$f_r(h,t) = \|\mathbf{M}_r\mathbf{h} + \mathbf{r} - \mathbf{M}_r\mathbf{t}\|_{\ell_{1/2}}^2. \tag{3}$$

A disadvantage of TransR/CTransR is that it can not deal with large-scale knowledge graphs since the calculation on matrixes used by it is very time-consumption.

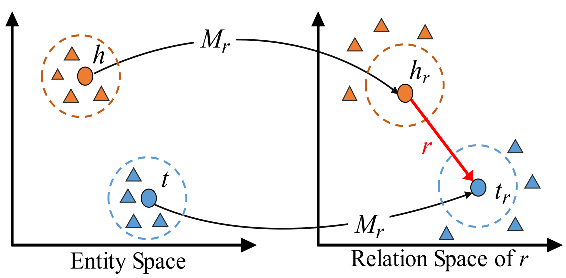


FIGURE 3. Simple illustration of TransR.

TransD: TransD [21] generates two vectors for each entity and relation, where the first vector is the embedding of the entity (or relation), and the second vector is used for constructing projecting matrices. Basic idea of TransD is illustrated in Fig. 4, where each shape denotes an entity pair involved in a relation r. Score function of TransD is as follows

$$f_r(h,t) = \|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_{\ell_{1/2}}^2, \tag{4}$$

where $\mathbf{h}_\perp = \mathbf{M}_{rh}\mathbf{h}$, $\mathbf{t}_\perp = \mathbf{M}_{rt}\mathbf{t}$, $\mathbf{M}_{rh} = \mathbf{r}_p\mathbf{h}_p^\top + \mathbf{I}^{m\times n}$, $\mathbf{M}_{rt} = \mathbf{r}_p\mathbf{t}_p^\top + \mathbf{I}^{m\times n}$, $\mathbf{h}_p$, $\mathbf{r}_p$, $\mathbf{t}_p$ are projection vectors and $\mathbf{I}^{m\times n}$ is an identity matrix.

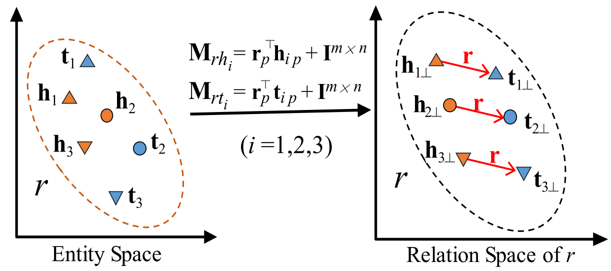


FIGURE 4. Simple illustration of TransD.

TranSparse: TranSparse [22] makes an improvement to TransR. Since all relations are trained with the same number of parameters in TransR, it is possible that simple relations might be overfitting while complex relations are underfitting. To deal with this problem, TranSparse uses sparse matrices instead of projection matrices to model diverse relations.

FT: As shown in Fig. 5, FT [23] makes an improvement to TransE by using a flexible translation principle $\mathbf{h} + \mathbf{r} \approx \alpha\mathbf{t}$ instead of $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$, so that the embedding of t is not fixed on a point but a plane when $\mathbf{h}$ and $\mathbf{r}$ hold. FT uses the following score function

$$f_r(h,t) = (\mathbf{h} + \mathbf{r})^\top \mathbf{t} + \mathbf{h}^\top(\mathbf{t} - \mathbf{r}). \tag{5}$$

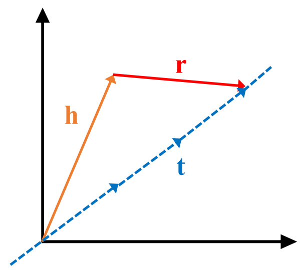


FIGURE 5. Simple illustration of FT.

B. OTHER METHODS

Besides translation-based models, there are many other approaches for knowledge graph embedding. Here, we introduce some typical models which will be used as our baselines.

Structured Embedding (SE): Be corresponding to each triple (h, r, t), SE [24] sets two relation-specific matrices $\mathbf{M}_{r,h}$ and $\mathbf{M}_{r,t}$ for the head and tail entities respectively, and uses the following score function

$$f_r(h,t) = \|\mathbf{M}_{r,h}\mathbf{h} - \mathbf{M}_{r,t}\mathbf{t}\|_1. \tag{6}$$

Neural Tensor Network (NTN): NTN [11] uses an expressive score function as follows

$$f_r(h,t) = \mathbf{u}_r^\top g(\mathbf{h}^\top \mathbf{M}_r \mathbf{t} + \mathbf{M}_{r,1}\mathbf{h} + \mathbf{M}_{r,2}\mathbf{t} + \mathbf{b}_r), \tag{7}$$

where $\mathbf{u}_r$ is a relation-specific linear layer, $g()$ is `tanh` operation, $\mathbf{M}_r \in \mathbb{R}^{d\times d\times k}$ is a 3-way tensor and $\mathbf{M}_{r,1}, \mathbf{M}_{r,2} \in \mathbb{R}^{k\times d}$ are weight matrices.

Single Layer Model (SLM): SLM was designed as a baseline for NTN [11] by using the following score function

$$f_r(h,t) = \mathbf{u}_r^\top f(\mathbf{M}_{r,h}\mathbf{h} + \mathbf{M}_{r,t}\mathbf{t}), \tag{8}$$

where $\mathbf{M}_{r,h}$ and $\mathbf{M}_{r,t}$ are parameter matrices and $f()$ is the `tanh` operation.

Semantic Matching Energy (SME): SME [13], [25] captures the correlations between entities and relations via matrix operations, and uses the same parameters for different relations. SME considers two kinds of semantic matching energy functions for the training process, one is the linear form

$$f_r(h,t) = (\mathbf{M}_1\mathbf{h} + \mathbf{M}_2\mathbf{r} + \mathbf{b}_1)^\top(\mathbf{M}_3\mathbf{t} + \mathbf{M}_4\mathbf{r} + \mathbf{b}_2), \tag{9}$$

and the other is the bilinear form

$$f_r(h,t) = ((\mathbf{M}_1\mathbf{h})\otimes(\mathbf{M}_2\mathbf{r})+\mathbf{b}_1)^\top((\mathbf{M}_3\mathbf{t})\otimes(\mathbf{M}_4\mathbf{r})+\mathbf{b}_2), \tag{10}$$

where $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_3$ and $\mathbf{M}_4$ are weight matrices, $\otimes$ is the Hadamard product, $\mathbf{b}_1$ and $\mathbf{b}_2$ are bias vectors. In [13], the matrices of bilinear form are redefined with 3-way tensors.

Latent Factor Model (LFM): LFM [26], [27] encodes each entity as a vector and sets a matrix $\mathbf{M}_r$ for each relation. Its score function is as follows

$$f_r(h,t) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t}. \tag{11}$$

RESCAL Is a collective matrix factorization model and we also report its results presented in [19], [28], and [29].

III. OUR METHOD

In the rest of the paper, we use S and S' to denote respectively the set of positive triples and the set of negative triples. Therefore, $(h,r,t) \in S$ means that "(h,r,t) is correct" and $(h,r,t) \in S'$ means that "(h,r,t) is not correct". The set of entities and relations is represented by E and R respectively.

A. MOTIVATION OF OUR APPROACH

Although existing translation-based methods achieved significant performance on link prediction and triple classification, they are still not good at dealing with large scale knowledge graphs containing diverse entities and relations. Here we investigate two phenomenons.

(1) Entities in knowledge graphs are diverse and complex. Firstly, for each entity acted as a head in triples, it is often connected to many tail entities. Correspondingly, for each entity acted as a tail, there are often many head entities connected to it. Secondly, for each pair of entities which acted as a head and a tail respectively, there are often many relations connected to the pair.

Fig. 6 illustrates this phenomenon on the FB15k data set. The X-axis in the figure is the index of entities which might act as heads, and the Y-axis is the index of entities which might act as tails. The Z-axis is the number of relations. Each dot in the figure indicates the number of relations connected to the pair that represented by the X-value and Y-value of the dot. From the figure, we can see that although majority of pairs in FB15k are connected by one or two relations,

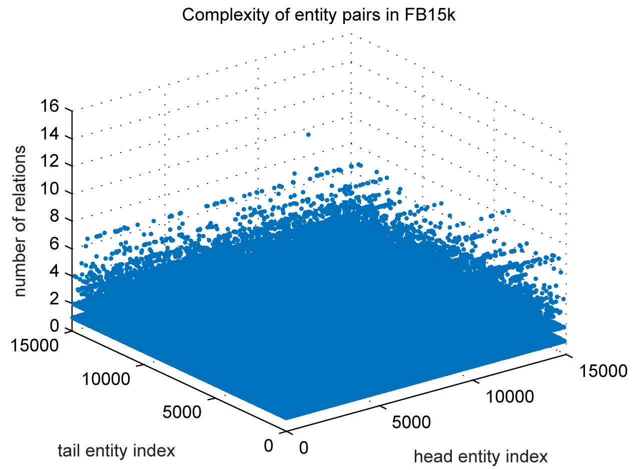


FIGURE 6. Entity pair statistics of FB15k which contains 14,951 entities.

there are still many pairs which are connected by eight or even ten relations. As an intuitive example, entities *William Jefferson Clinton* and *America* are connected by many relations as follows (*William Jefferson Clinton, PresidentOf, America*), (*William Jefferson Clinton, BornIn, America*), (*William Jefferson Clinton, Nationality, America*).

(2) Relations in a knowledge graph are very complex. As discussed by Bordes *et al.* [18], 73.8% of relationships in FB15k are complex relationships: 22.7% for 1-to-N, 28.3% for N-to-1, and 22.8% for N-to-N. Furthermore, there exist many special relations which are reflexive, symmetric, and transitive.

The principle $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ adopted by many translation-based models can not deal with the above phenomenons. More precisely, let's investigate the following four cases.

(1) Suppose there are many triples $(h, r_1, t) \in S, \ldots, (h, r_n, t) \in S$. Since all entities and relations are embedded in the same space, by the principle $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, we will get $\mathbf{r}_1 = \ldots = \mathbf{r}_n$.

(2) As shown by Fig. 7a, suppose r is a 1-to-N relation and there exist triples $(h, r, t_i) \in S$ with $i \in 1, 2, \ldots, n$, then we will also get $\mathbf{t}_1 = \ldots = \mathbf{t}_n$. Similarly, as shown by Fig. 7b, suppose r is a N-to-1 relation and there exist triples $(h_i, r, t) \in S$ with $i \in 1, 2, \ldots, n$, we will get $\mathbf{h}_1 = \ldots = \mathbf{h}_n$.

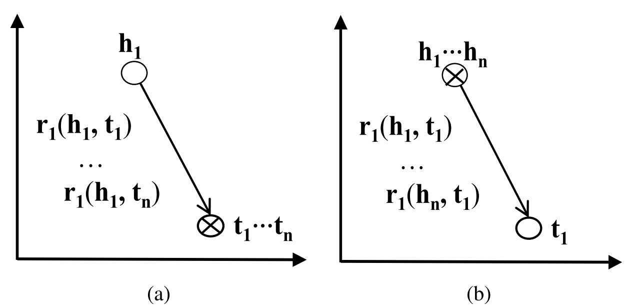


FIGURE 7. An example of entity vectors trained wrong by complex relations.

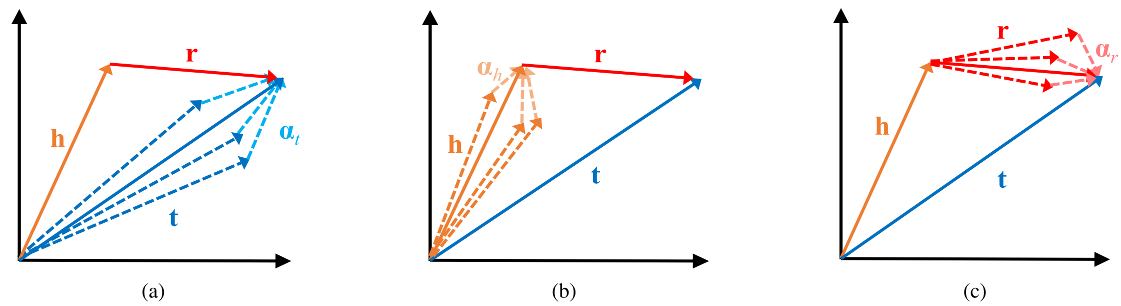


FIGURE 8. Simple illustration of DT.

(3) Suppose r is a reflexive relation and there are two triples $(h, r, t), (t, r, h) \in S$, then we will get a strange result of $\mathbf{h} = \mathbf{t}$ and $\mathbf{r} = 0$.

(4) Suppose r is a transitive relation and there exist some triples $(h_1, r, t_1), (t_1, r, t_2), (h_1, r, t_2) \in S$, then we will get $\mathbf{h}_1 + 2\mathbf{r} = \mathbf{t}_2$ and $\mathbf{h}_1 + \mathbf{r} = \mathbf{t}_2$, and consequently $\mathbf{r} = 0$.

Although the FT model illustrated in Fig. 5 is more flexible than TransE, it still can not solve the above problems since the direction of vector embeddings is constrained.

B. DYNAMIC TRANSLATION

In this paper, we propose a new translation principle named dynamic translation (DT for short) to solve the above problems. Basic idea of DT is shown in Fig. 8. For each triple (h, r, t), suppose the embeddings of h and r are given, then we permit $\mathbf{t}$ to be a range of plane, rather than be a fixed vector by the TransE model or a set of vectors in the same direction by the FT model. Similarly, suppose the embeddings of h and t are given, then the range of $\mathbf{r}$ is a plane; suppose the embeddings of r and t are given, then the range of $\mathbf{h}$ is also a plane.

By using the DT principle, the four problems investigated in the previous subsection can be solved as follows.

(1) In the case that there are many triples $(h, r_i, t) \in S$ with $i \in 1, 2, \ldots, n$, we will not reach a strange result since $\mathbf{r}_1 = \mathbf{t} - \mathbf{h} - \boldsymbol{\alpha}_{r_1}, \ldots, \mathbf{r}_n = \mathbf{t} - \mathbf{h} - \boldsymbol{\alpha}_{r_n}$.

(2) In the case that r is a 1-to-N relation with triples $(h, r, t_i) \in S$, $i \in 1, 2, \ldots, n$, we will get $\mathbf{t}_1 = \mathbf{h} + \mathbf{r} - \boldsymbol{\alpha}_{t_1}, \ldots, \mathbf{t}_n = \mathbf{h} + \mathbf{r} - \boldsymbol{\alpha}_{t_n}$.

(3) In the case that r is a reflexive relation and there are two triples $(h, r, t), (t, r, h) \in S$, we will get $\mathbf{r} = (\mathbf{t}+\boldsymbol{\alpha}_t)-(\mathbf{h}+\boldsymbol{\alpha}_h)$.

(4) In the case that r is a transitive relation and some triples $(h_1, r, t_1), (t_1, r, t_2), (h_1, r, t_2) \in S$, we will get $\mathbf{h}_1+\boldsymbol{\alpha}_{h_1}+\mathbf{r} = \mathbf{t}_1 + \boldsymbol{\alpha}_{t_1}, \mathbf{t}_1 + \boldsymbol{\alpha}_{h_1} + \mathbf{r} = \mathbf{t}_2 + \boldsymbol{\alpha}_{t_2}$ and $\mathbf{h}_1 + \boldsymbol{\alpha}_{h_1} + \mathbf{r} = \mathbf{t}_2 + \boldsymbol{\alpha}_{t_2}$.

It should be noted that, given the embeddings of h and r, all the possible embeddings of t are very similar, and the angle between any two possible embeddings of t is very small. In another word, the range of all the possible embeddings of t is a very small plane. Similarly, given the embeddings of h and t (or r and t), the range of all the possible embeddings of r (resp. h) is a very small plane. More precisely, in models such as TransE, the constraints of L_2-norm for the embeddings of entities and relations are 1. Here, in our DT model, we introduce $\boldsymbol{\alpha}_h$, $\boldsymbol{\alpha}_r$ and $\boldsymbol{\alpha}_t$, and the L_2-norm of them are values from the set $\{0.1, 0.2, 0.3\}$, so that the possible embeddings of h, r and t can be restricted in a very small plane. DT is a flexible principle that can be combined with many translation-based models. In the following subsections, we will combine DT with the models TransE, TransR and TranSparse respectively.

C. TRANSE-DT

We use TransE-DT to denote the model constructed by combining TransE with DT. In TransE, embeddings of entities and relations are in the same space and the translation principle is

$$\mathbf{h} + \mathbf{r} \approx \mathbf{t}. \tag{12}$$

In TransE-DT, we redefine the translation principle as

$$(\mathbf{h} + \boldsymbol{\alpha}_h) + (\mathbf{r} + \boldsymbol{\alpha}_r) \approx (\mathbf{t} + \boldsymbol{\alpha}_t). \tag{13}$$

Correspondingly, the score function is

$$f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{\ell_{1/2}}, \tag{14}$$

where $\mathbf{h}, \mathbf{r}, \mathbf{t}, \boldsymbol{\alpha}_h, \boldsymbol{\alpha}_r, \boldsymbol{\alpha}_t \in \mathbb{R}^n$.

D. TRANSR-DT

We use TransR-DT to denote the model constructed by combining TransR with DT. In TransR, entities are projected to the relation space by $\mathbf{M}_r$ and the translation principle is

$$\mathbf{M}_r\mathbf{h} + \mathbf{r} \approx \mathbf{M}_r\mathbf{t}. \tag{15}$$

In TransR-DT, the translation principle is redefined as

$$(\mathbf{M}_r\mathbf{h} + \boldsymbol{\alpha}_h) + (\mathbf{r} + \boldsymbol{\alpha}_r) \approx (\mathbf{M}_r\mathbf{t} + \boldsymbol{\alpha}_t). \tag{16}$$

Correspondingly, the score function is

$$f_r(h, t) = \|\mathbf{M}_r\mathbf{h} + \mathbf{r} - \mathbf{M}_r\mathbf{t}\|_{\ell_{1/2}}, \tag{17}$$

where $\mathbf{h}, \mathbf{t} \in \mathbb{R}^n, \mathbf{r}, \boldsymbol{\alpha}_h, \boldsymbol{\alpha}_r, \boldsymbol{\alpha}_t \in \mathbb{R}^m$, and $\mathbf{M}_r \in \mathbb{R}^{m\times n}$. Here, $\mathbf{M}_r$ is a projection matrix for relation r, by which entities are projected from entity space to the semantic space of relation r.

TABLE 1. Complexities (the number of parameters and the times of operations) of several embedding models. N_e and N_r represent the number of entities and relations, respectively. N_t represents the number of triples in a knowledge graph. m is the dimension of entity embedding space and n is the dimension of relation embedding space.

Model	#Parameters	#Operations (Time complexity)
SE	$O(N_em + 2N_rn^2)(m = n)$	$O(2m^2N_t)$
SME(linear)	$O(N_em + N_rn + 4mk + 4k)(m = n)$	$O(4mkN_t)$
SME(bilinear)	$O(N_em + N_rn + 4mks + 4k)(m = n)$	$O(4mksN_t)$
LFM	$O(N_em + N_rn^2)(m = n)$	$O((m^2 + m)N_t)$
SLM	$O(N_em + N_r(2k + 2nk))(m = n)$	$O((2mk + k)N_t)$
NTN	$O(N_em + N_r(n^2s + 2ns + 2s))(m = n)$	$O(((m^2 + m)s + 2mk + k)N_t)$
TransE	$O(N_em + N_rn)(m = n)$	$O(N_t)$
TransH	$O(N_em + 2N_rn)(m = n)$	$O(2mN_t)$
TransR	$O(N_em + N_r(m + 1)n)$	$O(2mnN_t)$
CTransR	$O(N_em + N_r(m + d)n)$	$O(2mnN_t)$
TransD	$O(2N_em + 2N_rn)(m = n)$	$O(2nN_t)$
TranSparse	$O(N_em + 2N_r(1 - \hat{\theta})(m + 1)n)(0 \ll \hat{\theta} \leq 1)$	$O(2(1 - \hat{\theta})mnN_t)(0 \ll \hat{\theta} \leq 1)$
TransE-DT	$O(N_em + N_rn)(m = n)$	$O(3mN_t)$
TransR-DT	$O(N_em + 2N_r(m + 1)n)$	$O((2n + 3)mN_t)$
TranSparse-DT	$O(N_em + 2N_r(1 - \hat{\theta})(m + 1)n)(0 \ll \hat{\theta} \leq 1)$	$O((2n - 2\hat{\theta}n + 3)mN_t)(0 \ll \hat{\theta} \leq 1)$

E. TRANSPARSE-DT

We use TranSparse-DT to denote the model constructed by combining TranSparse with DT. TranSparse is an extension of TransR. It replaces the dense matrices in TransR by sparse matrices, and makes use of two separate sparse matrices $\mathbf{M}_r^h(\theta_r^h)$ and $\mathbf{M}_r^t(\theta_r^t)$ to project the head entity and the tail entity respectively. Therefore, the translation principle used by TranSparse is as follows:

$$\mathbf{M}_r^h(\theta_r^h)\mathbf{h} + \mathbf{r} \approx \mathbf{M}_r^t(\theta_r^t)\mathbf{t}. \tag{18}$$

Here, the sparse degrees of transfer matrices are defined as

$$\theta_r^l = 1 - (1 - \theta_{min})N_r^l/N_r^l{*} \quad (l = h, t), \tag{19}$$

where N_r^l denotes the number of entities (head or tail) linked by relation r at location l and $N_r^l{*}$ denotes the maximum number of N_r^l. In TranSparse-DT, we redefine the translation principle as

$$(\mathbf{M}_r^h(\theta_r^h)\mathbf{h} + \boldsymbol{\alpha}_h) + (\mathbf{r} + \boldsymbol{\alpha}_r) \approx (\mathbf{M}_r^t(\theta_r^t)\mathbf{t} + \boldsymbol{\alpha}_t). \tag{20}$$

Correspondingly, we use the following score function:

$$f_r(h, t) = \|\mathbf{M}_r^h(\theta_r^h)\mathbf{h} + \mathbf{r} - \mathbf{M}_r^t(\theta_r^t)\mathbf{t}\|_{\ell_{1/2}}, \tag{21}$$

where $\mathbf{h}, \mathbf{t} \in \mathbb{R}^n$, and $\mathbf{r}, \boldsymbol{\alpha}_h, \boldsymbol{\alpha}_r, \boldsymbol{\alpha}_t \in \mathbb{R}^m$.

F. TRAINING OBJECTIVE

In all the above models, we use the following common margin-based score function for training process:

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} max(0, f_r(h, t) + \gamma - f_r(h', t')), \tag{22}$$

where $max(x, y)$ aims to get the maximum between x and y, and γ is the margin.

During the training process, we will make use of a set S' of negative triples. The set S' is generated by replacing the entities of each positive triple $(h, r, t) \in S$. For corrupting a triple $(h, r, t) \in S$, following the idea of Wang *et al.* [19], we take different probabilities to replace the head entity h and the tail entity t. In order to reduce the possibility of generating false negative-triples by such a replacement, for relations of 1-to-N,N-to-1 and N-to-N, it is better to take more probability to replace the "one" side. In the experiments of Section 4, we will make use of two sampling methods: the traditional sampling method "unif", and the method "bern" adopted in [19]. Furthermore, we will make use of the stochastic gradient descent (SGD) [30] to minimize the objective loss function.

G. COMPARISON OF COMPLEXITY

In table 1 we list the complexities of all the models introduced in the Section of Related Work, and compare them with the complexities of our models. It is shown that there is a slight increasement in time complexity after introducing the DT principle. However, the introducing of DT principle does not increase the number of parameters, i.e., TransE-DT (or TransR-DT, TranSparse-DT) has the same number of parameters with the original model TransE (resp., TransR-DT, TranSparse-DT). Therefore, we can say that our method are still efficient.

IV. EXPERIMENTS AND ANALYSIS

A. DATA SETS AND EXPERIMENT SETTING

We empirically evaluate our method on two tasks: link predication [18] and triple classification [11]. In order to compare our method with other works in the literature, we implement these tasks on two typical knowledge graphs, i.e., WordNet [5] and Freebase [7]. WordNet is a large-scale lexical knowledge graph which provides semantic knowledge of words. In WordNet, each entity is a synset which consists of several words and expresses different concepts. Relationships in WordNet are defined as conceptual-semantic and lexical relations. For example, the triple (*Robin, Color, Red*) builds an attribute *Color* between the noun *Robin* and the adjective *Red*. In the experiments, we use two subsets of WordNet, i.e., WN18 [13] and WN11 [11], which contains 18 relation types and 11 relation types respectively. Freebase is a large collaborative knowledge graph and represents general facts of the world. For example, the triple (*Steve Jobs, Nationality, America*) builds a relation of *Nationality* between the name entity *Steve Jobs* and the country entity *America*. We also use two subsets of Freebase, i.e., FB15K [13] and FB13 [11]. Table 2 lists statistics of these data sets.

TABLE 2. Statistics of the data sets.

Dataset	#Ent	#Rel	#Train	#Valid	#Test
WN18	40,943	18	141,442	5,000	5,000
FB15k	14,951	1,345	483,142	50,000	59,071
WN11	38,696	11	112,581	2,609	10,544
FB13	75,043	13	316,232	5,908	23,733

B. LINK PREDICTION

Link prediction aims to predict the missing entity (h or t) for a relation fact triple (h, r, t) [18], [24], [25]. Instead of finding the best suitable entity, this task will return a set of candidate entities for each position of missing entity. As the works in the literature [18], [24], we use the data sets WN18 and FB15K for the experiment.

For each test triple (h, r, t), we will first replace the head or tail entity by all of the entities in the knowledge graph, compute the similarity scores for these entities by the score function f_r, and then rank these entities in the descending order of similarity scores. We make use of two measures as our evaluation metric: (1) mean rank of correct entities (denoted by Mean Rank), and (2) proportion of correct entities ranked in top 10 (denoted by Hits@10). Obviously, a good model for link prediction should achieve a low value in Mean Rank and a high value in Hits@10.

We will report the evaluation results in two evaluation settings. Note that, for a triple (h, r, t), its corrupted triples may also exist in knowledge graphs and should be regard as correct triples. However, the above evaluation may rank these corrupted triples in front of the correct one and cause the underestimation of the performance. Therefore, we should filter out these triples from the training, validation and testing sets before the ranking process. If we do this, then we indicate the evaluation setting by "Filt", otherwise we indicate the evaluation setting by "Raw".

Table 3 shows the values of training parameters used in our experiments for TransE-DT, TransR-DT and TranSparse-DT. λ is the learning rate, γ is the margin, n and m are embedding dimensions for entities and relations, B is the mini-batch size, and D.S is the dissimilarity measure in score functions. The iteration number of SGD is 1000.

TABLE 3. Parameter values in link prediction.

Dataset	Model	λ	γ	n, m	B	D.S
WN18	TransE-DT	0.01	3.5	50	1,440	L_1
	TransR-DT	0.001	3.5	50	1,440	L_1
	TranSparse-DT	0.001	3.5	50	1,440	L_1
FB15k	TransE-DT	0.001	1.5	100	4,800	L_1
	TransR-DT	0.0001	1.5	100	4,800	L_1
	TranSparse-DT	0.0001	1.5	100	4,800	L_1

Results: Table 4 lists the experimental results of link prediction. The upper part of the results comes from the literature directly, since all methods discussed here use the same testing data sets. Because there is no result for FT on link prediction in the literature, we do not discuss it in the table.

From the table, we can see that our methods get the best results for the Hits@10 metric, and TranSparse-DT becomes the new state-of-the-art model. Furthermore, TransE-DT is better than TransE, and TransR-DT is better than TransR. For the Mean Rank metric, the superiority of our models is not so obvious. In the data set FB15k, our model TranSparse-DT (unif) (i.e., TranSparse-DT model with the traditional sampling method "unif") gets the best results in the "Filt" evaluation setting, and TranSparse-DT (bern) (i.e., TranSparse-DT model with the sampling method "bern") gets the best results in the "Raw" evaluation setting; furthermore TransE-DT is better than TransE and TransR-DT (unif) is better than TransR (unif). However, in the data set WN18, although TransE-DT is better than TransE, both TransR-DT and TranSparse-DT are not good as TransR and TranSparse. The reason is that, as discussed by Wang *et al.* [19], the number of relations in the data set WN18 is small and the advantage of our methods on dealing with complex relations is not reflected.

Table 5 exhibits Hits@10s according to the mapping property of relations on FB15k. Within the 1,345 relations contained in FB15k, 24% are 1-to-1, 23% are 1-to-N, 29% are N-to-1, and 24% are N-to-N. It is notable that TranSparse-DT outperforms all the other models in both N-to-N relations and N-to-1 relations. Furthermore, TransE-DT and TransR-DT achieve great improvement in all relation categories compared with TransE and TransR respectively. Since all transitive relations and some symmetric relations are in general N-to-N relations, we can say that our dynamic translation method has significant advantages in dealing with complex relations.

C. TRIPLE CLASSIFICATION

Triple classification is a binary classification task. It aims to judge whether a given triple (h, r, t) is correct or not. As other works in the literature, we use three data sets WN11, FB13 and FB15K for the experiment. We use the negative triples released by Socher *et al.* [11] on the data sets WN11 and FB13, and adopt the same setting of [11] to generate negative triples on the data set FB15K by corrupting positive

TABLE 4. Link prediction results.

Dataset	WN18				FB15k			
Metric	Mean Rank		Hits@10		Mean Rank		Hits@10	
	Raw	Filt	Raw	Filt	Raw	Filt	Raw	Filt
RESCAL	1180	1163	37.2	52.8	828	683	28.4	44.1
SE	1,011	985	68.5	80.5	273	162	28.8	39.8
SME (linear / bilinear)	542 / 526	533 / 509	65.1 / 54.7	74.1 / 61.3	274 / 284	154 / 158	30.7 / 31.3	40.8 / 41.3
LFM	469	456	71.4	81.6	283	164	26.0	33.1
TransE	263	251	75.4	89.2	243	125	34.9	47.1
TransH (unif / bern)	318 / 401	303 / 388	75.4 / 73.0	86.7 / 82.3	211 / 212	84 / 87	42.5 / 45.7	58.5 / 64.4
TransR (unif / bern)	232 / 238	219 / 225	78.3 / 79.8	91.7 / 92.0	226 / 198	78 / 77	43.8 / 48.2	65.5 / 68.7
CTransR (unif / bern)	243 / 231	230 / 218	78.9 / 79.4	92.3 / 92.3	233 / 199	82 / **75**	44.0 / 48.4	66.3 / 70.2
TransD (unif / bern)	242 / 224	229 / 212	79.2 / 79.6	92.5 / 92.2	211 / 194	67 / 91	49.4 / 53.4	74.2 / 77.3
TranSparse (unif / bern)	**233 / 223**	**221 / 211**	79.6 / 80.1	93.4 / 93.2	216 / 190	66 / 82	50.3 / 53.7	**78.4** / 79.9
TransE-DT (unif / bern)	234 / 228	220 / 216	78.2 / 76.2	91.6 / 88.4	**207** / 212	61 / 80	48.2 / 50.7	72.1 / 72.5
TransR-DT (unif / bern)	256 / 243	244 / 232	79.1 / 80.6	92.3 / 93.8	213 / 202	67 / 95	48.9 / 51.2	74.3 / 75.1
TranSparse-DT (unif / bern)	248 / 234	232 / 221	**80.0 / 81.4**	**93.6 / 94.3**	208 / **188**	**58** / 79	**51.2 / 53.9**	**78.4 / 80.2**

TABLE 5. Results on FB15k by relation category (%).

Tasks	Predicting Head (Hits@10)				Predicting Tail (Hits@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
SE	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME (linear)	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME (bilinear)	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH (unif / bern)	66.7 / 66.8	81.7 / 87.6	30.2 / 28.7	57.4 / 64.5	63.7 / 65.5	30.1 / 39.8	83.2 / 83.3	60.8 / 67.2
TransR (unif / bern)	76.9 / 78.8	77.9 / 89.2	38.1 / 34.1	66.9 / 69.2	76.2 / 79.2	38.4 / 37.4	76.2 / 90.4	69.1 / 72.1
CTransR (unif / bern)	78.6 / 81.5	77.8 / 89.0	36.4 / 34.7	68.0 / 71.2	77.4 / 80.8	37.8 / 38.6	78.0 / 90.1	70.3 / 73.8
TransD (unif / bern)	80.7 / 86.1	85.8 / 95.5	47.1 / 39.8	75.6 / 78.5	80.0 / 85.4	54.5 / 50.6	80.7 / 94.4	77.9 / 81.2
TranSparse (unif / bern)	**83.2** / 87.1	85.2 / **95.8**	51.8 / 44.4	80.3 / 81.2	82.6 / **87.5**	**60.0 / 57.0**	**85.5** / 94.5	82.5 / 83.7
TransE-DT (unif / bern)	78.9 / 84.7	80.6 / 94.5	46.5 / 34.1	72.4 / 72.1	77.6 / 83.1	50.3 / 41.7	81.8 / 93.8	74.6 / 74.6
TransR-DT (unif / bern)	80.3 / 80.6	81.9 / 92.3	47.8 / 46.5	73.5 / 74.8	78.9 / 81.6	51.6 / 50.6	80.2 / 90.7	74.9 / 76.7
TranSparse-DT (unif / bern)	83.0 / **87.4**	**85.7 / 95.8**	**51.9 / 47.7**	**80.5 / 81.6**	**82.8** / 86.7	59.9 / 56.3	**85.5 / 94.8**	**82.9 / 84.0**

triples. For the triple classification, we set a threshold δ_r, which is optimized by maximizing classification accuracies on the validation set. A triple (h, r, t) will be classified as positive if its dissimilarity score is lower than δ_r, and negative otherwise. Parameter values for training TransE-DT, TransR-DT and TranSparse-DT are list in Table 6.

TABLE 6. Parameter values in triple classification.

Dataset	Model	α	λ	n, m	B	D.S
WN11	TransE-DT	0.01	4.2	20	1440	L_1
	TransR-DT	0.001	4.2	20	1440	L_1
	TranSparse-DT	0.001	4.2	20	1440	L_1
FB13	TransE-DT	0.002	2	100	960	L_1
	TransR-DT	0.001	2	100	480	L_1
	TranSparse-DT	0.001	2	100	200	L_2
FB15k	TransE-DT	0.001	2	100	4800	L_1
	TransR-DT	0.0001	2	100	4800	L_1
	TranSparse-DT	0.0001	2	100	4800	L_1

Results: Table 7 lists the experimental results of triple classification. The upper part of the results comes from the literature directly. Therefore, a good model for triple classification should achieve a high value.

We can see that, on all data sets, TransE-DT is better than TransE and TransR-DT is better than TransR. On the data set WN11 with the sampling method ąřbernąś, TranSparse-DT is better than all the other models and becomes the new state-of-the-art model. On the data sets FB13 and FB15 with the sampling method "bern", TranSparse-DT is also better than TranSparse although its performance is a little lower than that of TransD and TransE-FT respectively. As pointed by Lin *et al.* [20], FB13 is a denser graph and there are many strong correlations between entities; such a situation is favourable to TransD which uses two project vectors of entity and relation to construct dynamic matrices. We think this is the reason that TranSparse-DT is not better than TransD in the data set FB13. On the data set FB15k, TransE-FT achieves the best result, and the reason may be that TransE-FT changes the

TABLE 7. Accuracies on triple classification (%).

Dataset	WN11	FB13	FB15k
SE	53.0	75.2	-
SME (bilinear)	70.0	63.7	-
SLM	69.9	85.3	-
LFM	73.8	84.3	-
NTN	70.4	87.1	68.2
TransE (unif / bern)	75.9 / 75.9	70.9 / 81.5	77.3 / 79.8
TransH (unif / bern)	77.7 / 78.8	76.5 / 83.3	74.2 / 79.9
TransR (unif / bern)	85.5 / 85.9	74.7 / 82.5	81.1 / 82.1
CTransR (bern)	85.7	-	84.3
TransD (unif / bern)	85.6 / 86.4	85.9 / **89.1**	86.4 / 88.0
TranSparse (unif / bern)	86.8 / 86.8	86.5 / 87.5	87.4 / 88.5
TransE-FT	86.4	82.1	**90.5**
TransH-FT	78.3	80.7	82.1
TransR-FT	86.6	82.9	88.9
TransE-DT (unif / bern)	86.4 / 86.6	81.6 / 85.3	83.5 / 83.3
TransR-DT (unif / bern)	86.3 / 86.4	82.0 / 84.7	84.7 / 85.2
TranSparse-DT (unif / bern)	86.6 / **87.1**	85.8 / 87.9	87.2 / 88.9

score function which can better distinguish between negative triples and correct triples.

V. CONCLUSION

In this paper, we proposed a new translation-based principle named Dynamic Translation. Basic idea of DT is to relax the constraints on embeddings of entities or relations by permitting the embeddings to be at some planes. We combined this principle with the classical models TransE, TransR and TranSparse respectively, and presented new models TransE-DT, TransR-DT and TranSparse-DT correspondingly. Experimental results show that our approach achieves significant improvement in the link prediction task and the triple classification task, since it can capture more semantic information on complex entities and diverse relations.

In this paper, we only combine our DT principle with classical translation-based models. One of our future works is to incorporate more information such as the relation paths [31], [32] and the textual descriptions on entities [33]. Moreover, in the last year, there are two remarkable works on knowledge graph embeddings. One is HOLE [34] which takes into account the circular correlation of vectors and can capture rich interactions in relational data. The other is ProjE [35] which views the link prediction task as ranking problem and can learn the joint embeddings of both entities and relations. In the future work, we will try to combine our DT principle with these works.

REFERENCES

[1] A. Bordes, J. Weston, and N. Usunier, "Open question answering with weakly supervised embedding models," in *Proc. ECML PKDD*, Nancy, France, 2014, pp. 165–180.

[2] W. Shen, J. Wang, P. Luo, and M. Wang, "Linking named entities in Tweets with knowledge base via user interest modeling," in *Proc. ACM SIGKDD*, Chicago, IL, USA, 2013, pp. 68–76.

[3] S. Szumlanski and F. Gomez, "Automatically acquiring a semantic network of related concepts," in *Proc. CIKM*, Toronto, ON, Canada, 2010, pp. 19–28.

[4] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, "Knowledge-based weak supervision for information extraction of overlapping relations," in *Proc. ACL*, Portland, OR, USA, 2011, pp. 541–550.

[5] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995.

[6] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proc. WWW*, Banff, AB, Canada, 2007, pp. 697–706.

[7] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. SIGMOD*, Vancouver, BC, Canada, 2008, pp. 1247–1250.

[8] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, Jr., and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *Proc. AAAI*, Atlanta, GA, USA, 2010, pp. 1306–1313.

[9] N. Lao, T. Mitchell, and W. W. Cohen, "Random walk inference and learning in a large scale knowledge base," in *Proc. EMNLP*, Edinburgh, U.K., 2011, pp. 529–539.

[10] W. Y. Wang, K. Mazaitis, N. Lao, and W. W. Cohen, "Efficient inference and learning in a large knowledge base," *Mach. Learn.*, vol. 100, no. 1, pp. 101–126, Apr. 2015.

[11] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Proc. NIPS*, Lake Tahoe, NV, USA, 2013, pp. 926–934.

[12] H. Gui *et al.*, "Embedding learning with events in heterogeneous information networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 11, pp. 2428–2441, Nov. 2017.

[13] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Mach. Learn.*, vol. 94, no. 2, pp. 233–259, Feb. 2014.

[14] Z. Huang, Y. Zheng, R. Cheng, Y. Sun, N. Mamoulis, and X. Li, "Meta structure: Computing relevance in large heterogeneous information networks," in *Proc. ACM SIGKDD*, San Francisco, CA, USA, 2016, pp. 1595–1604.

[15] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin, "Relation extraction with matrix factorization and universal schemas," in *Proc. HLT-NAACL*, Atlanta, GA, USA, 2013, pp. 74–84.

[16] J. Weston, A. Bordes, O. Yakhnenko, and N. Usunier, "Connecting language and knowledge bases with embedding models for relation," in *Proc. EMNLP*, Seattle, WA, USA, 2013, pp. 1366–1371.

[17] T. Rocktäschel, S. Singh, and S. Riedel, "Injecting logical background knowledge into embeddings for relation extraction," in *Proc. HLT-NAACL*, Denver, CO, USA, 2015, pp. 1119–1129.

[18] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. NIPS*, Lake Tahoe, NV, USA, 2013, pp. 2787–2795.

[19] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI*, Québec City, QC, Canada, 2014, pp. 1112–1119.

[20] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI*, Austin, TX, USA, 2015, pp. 2181–2187.

[21] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. ACL*, Beijing, China, 2015, pp. 687–696.

[22] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *Proc. AAAI*, Phienix, AZ, USA, 2016, pp. 985–991.

[23] J. Feng, M. Huang, M. Wang, M. Zhou, Y. Hao, and X. Zhu, "Knowledge graph embedding by flexible translation," in *Proc. KR*, Cape Town, South Africa, 2016, pp. 557–560.

[24] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Proc. AAAI*, San Francisco, CA, USA, 2011, pp. 301–306.

[25] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Joint learning of words and meaning representations for open-text semantic parsing," in *Proc. AISTATS*, La Palma, Spain, 2012, pp. 127–135.

[26] R. Jenatton, N. L. Roux, A. Bordes, and G. Obozinski, "A latent factor model for highly multi-relational data," in *Proc. NIPS*, Lake Tahoe, NV, USA, 2012, pp. 3167–3175.

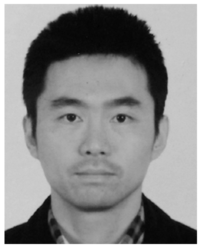
[27] I. Sutskever, R. Salakhutdinov, and J. B. Tenenbaum, "Modelling relational data using Bayesian clustered tensor factorization," in *Proc. NIPS*, Vancouver, BC, Canada, 2009, pp. 1821–1828.

[28] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proc. ICML*, Bellevue, WA, USA, 2011, pp. 809–816.
[29] M. Nickel, V. Tresp, and H.-P. Kriegel, "Factorizing YAGO: Scalable machine learning for linked data," in *Proc. WWW*, Lyon, France, 2012, pp. 271–280
[30] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.
[31] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases," in *Proc. EMNLP*, Lisbon, Portugal, 2015, pp. 705–714.
[32] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: Meta path-based top-k similarity search in heterogeneous information networks," *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, Aug. 2011.
[33] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proc. AAAI*, Phoenix, AZ, USA, 2016, pp. 2659–2665.
[34] M. Nickel, L. Rosasco, and T. A. Poggio, "Holographic embeddings of knowledge graphs," in *Proc. AAAI*, Phoenix, AZ, USA, 2016, pp. 1955–1961.
[35] B. Shi and T. Weninger, "ProjE: Embedding projection for knowledge graph completion," in *Proc. AAAI*, San Francisco, CA, USA, 2017, pp. 1236–1242.

LIANG CHANG received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, in 2008. He is currently a Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, China. His research interests include data and knowledge engineering, formal methods, and intelligent planning.

MANLI ZHU received the bachelor's degree in software engineering from the Institute of Zhoukou Normal University, in 2013. She is currently a Graduate Student with the School of Computer Science and Information Security, Guilin University of Electronic Technology, China. Her research interest is knowledge graph.

TIANLONG GU received the Ph.D. degree from Zhejiang University, China, in 1996. From 1998 to 2002, he was a Research Fellow with the School of Electrical and Computer Engineering, Curtin University of Technology, Australia, and a Post-Doctoral Fellow with the School of Engineering, Murdoch University, Australia. He is currently a Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, China. His research interests include formal methods, data and knowledge engineering, and software engineering.



CHENZHONG BIN received the M.S. degree from North West Polytechnical University, Xian, China, in 2005. He is currently a Lecturer with the School of Computer and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include data mining, machine learning, and IOT applications.

JUNYAN QIAN received the Ph.D. degree from the Southeast University of China in 2008. He is currently a Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, China. His research interests include formal verification and optimization algorithm.

JI ZHANG received the Ph.D. degree in computer science from Dalhousie University, Canada, in 2008. He is currently an Associate Professor (Reader) in computing with the University of Southern Queensland, Australia. He is an Australian Endeavour Fellow, a Queensland Fellow, and an Izaak Walton Killam Fellow, Canada. He served as the Principal Advisor for Research in the Division of ICT Services, USQ, from 2010 to 2013. His research interests are knowledge discovery and data mining, health informatics, information privacy, and security.