

Research Article

Multiview Translation Learning for Knowledge Graph Embedding

Chenzhong Bin ^{1,2}, Saige Qin,² Guanjun Rao,² Tianlong Gu ², and Liang Chang²

¹School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China

²Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

Correspondence should be addressed to Chenzhong Bin; binchenzhong@163.com and Tianlong Gu; cctlgu@guet.edu.cn

Received 14 October 2019; Accepted 9 July 2020; Published 25 August 2020

Academic Editor: Zhiang Wu

Copyright © 2020 Chenzhong Bin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, knowledge graph embedding methods have attracted numerous researchers' interest due to their outstanding effectiveness and robustness in knowledge representation. However, there are still some limitations in the existing methods. On the one hand, translation-based representation models focus on conceiving translation principles to represent knowledge from a global perspective, while they fail to learn various types of relational facts discriminatively. It is prone to make the entity congestion of complex relational facts in the embedding space reducing the precision of representation vectors associating with entities. On the other hand, parallel subgraphs extracted from the original graph are used to learn local relational facts discriminatively. However, it probably causes the relational fact damage of the original knowledge graph to some degree during the subgraph extraction. Thus, previous methods are unable to learn local and global knowledge representation uniformly. To that end, we propose a multiview translation learning model, named MvTransE, which learns relational facts from global-view and local-view perspectives, respectively. Specifically, we first construct multiple parallel subgraphs from an original knowledge graph by considering entity semantic and structural features simultaneously. Then, we embed the original graph and construct subgraphs into the corresponding global and local feature spaces. Finally, we propose a multiview fusion strategy to integrate multiview representations of relational facts. Extensive experiments on four public datasets demonstrate the superiority of our model in knowledge graph representation tasks compared to state-of-the-art methods.

1. Introduction

Knowledge graphs [1] are a sort of directed graphs consisting of entities as nodes and relations between entities as edges. And each relational fact of knowledge graphs is stored as a triplet (*head*, *relation*, *tail*), abbr. (h , r , t), where h and t represent head and tail entities, respectively, and r is a relationship from h to t . With the advent of the big data era, the scale of knowledge graph continues to grow; diverse large-scale knowledge graphs (e.g., WordNet [2] and Freebase [3]) have appeared. Despite the large scale of current knowledge graphs, they are still far from the knowledge completeness. For example, 75% of people in Freebase lack nationality information and 71% lack birthplace [4]. Therefore, it is necessary to design approaches to automatically complete or infer missing relational facts of the existing knowledge graph.

Recently, embedding-based approaches present strong feasibility and robustness in terms of knowledge graph completion, which project entities and relations of knowledge graphs into a dense, continuous, and low-dimensional vector space. Among the existing approaches, translation-based approaches have attracted numerous researchers' interests due to the outstanding effectiveness and robustness in the knowledge representation. The first translation-based method was proposed by Bordes et al., named TransE [5]. For each triplet (h , r , t), TransE treats a relation r as a translation operation from h to t in a vector space. If (h , r , t) holds, the translation principle $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ should be satisfied in the vector space, where \mathbf{h} , \mathbf{r} , and \mathbf{t} are vector representations of h , r , and t . TransE is a simple yet effective translation model in processing 1-to-1 simple relational facts, which stands for each single head entity connecting only one tail entity via a specific relation. It achieves state-of-

the-art performance on link prediction. However, the translation principle is too rigid to deal with complex relational facts, including 1-to- N , N -to-1, and N -to- N facts. Technically, it may cause spatial congestion of entities when many head entities (or tail entities) are projecting at only one point of the vector space.

To eliminate the weakness of TransE in representing complex relational facts, a series of improved models were proposed, such as TransH [6], TransR [7], TransD [8], and TransSparse [9]. Essentially, the above methods focused on designing various translation principles to learn complex relational facts more precisely. However, they still embed a complete knowledge graph into a single vector space from a global perspective while they fail to learn the various type of relational facts discriminatively. That is, each entity and relation of knowledge graphs are learned as corresponding unique representing vectors in their spaces. Hence, there are still vector congestions in the translation spaces. For a real world example, the head entities of the triples (*Obama, President, the United States*) and (*Trump, President, the United States*), i.e., Obama and Trump, are projected closely in the vector space due to the same social status. But, they are quite different in other perspectives.

To solve this problem, puTransE [10] splits knowledge graphs into multiple parallel spaces in the form of subgraphs from a local perspective and achieves the spatial sparsification of complex relational facts. Specifically in puTransE, entities and relations have respective feature representations in different parallel spaces. This approach improves the ability to learn complex relational facts due to avoiding the spatial congestion of entities in complex relational facts. However, puTransE still has two shortcomings. First, puTransE poses excessive sparseness to simple relational facts during the parallel space generation. Consequently, it hardly learns the complete vector representations of simple relational facts within a single subspace. This is because puTransE performs the spatial sparsification not only to complex relational facts but also to simple ones. Second, puTransE randomly selects local knowledge to construct multiple parallel spaces, which is prone to impair relational facts of the original knowledge graph. For example, there is a golden triplet in the original graph that cannot be consisted by entities and relations in any parallel spaces.

In summary, all of the above methods embed knowledge graph from a single perspective, i.e., from a local-view or global-view. Thus, they fail to learn local and global knowledge representation uniformly. To that end, we borrow the idea of multiview learning methods [11, 12] to propose a multiview translation learning model, named MvTransE, which embeds relational facts from global-view and local-view, concurrently. In detail, we first generate multiple parallel subgraphs through semantic and structural perspectives of entities to accurately capture local-view knowledge of the knowledge graph. Then, the original knowledge graph and generated parallel subgraphs are embedded into global-view and local-view spaces, respectively. Finally, we propose a multiview fusion strategy to integrate multiview representations of relational facts. We outline the main contributions of this paper as follows:

- (1) We incorporate the idea of multiview learning into our model MvTransE, which can precisely learn relational facts from both global and local views
- (2) Our model extracts local knowledge from semantic and structural perspectives to construct multiple parallel subgraphs so as to solve the entity spatial congestion problem by learning local-view representations of relational facts
- (3) MvTransE applies a multiview fusion strategy to combine global-view and local-view representations of the knowledge graph, which effectively overcomes the missing of relational facts in parallel spaces
- (4) Extensive experiment results demonstrate our method outperforms state-of-the-art models in two knowledge graph completion tasks

2. Related Work

Since the appearance of knowledge graphs, numerous researchers have studied various methods to represent relational facts of the graphs. Initially, some embedding-based models such as Structured Embedding (SE) [13], Semantic Matching Energy (SME) [14, 15], Latent Factor Model (LFM) [16], and Neural Tensor Network (NTN) [17] achieved considerable performance in knowledge representation, while fail to cope with large-scale graphs due to the computing complexity. Recently, translation-based models have attracted lots of attention due to their effective and robust representation abilities.

TransE [5] is the first translation-based method, which treats a relation r as a translation from h to t for a triplet (h, r, t) . Hence, TransE defines the scoring function as $f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/1/2}$, where $\|\cdot\|_{1/1/2}$ stands for norm-1 or norm-2 computation. During the model training, if a triplet (h, r, t) holds, the translation principle $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ should be satisfied in the vector space, of which process is illustrated in Figure 1. That is, TransE keeps translation vectors $(\mathbf{h} + \mathbf{r})$ approximate to tail vector \mathbf{t} . TransE achieves remarkable performance in representing simple relational facts, i.e., 1-to-1 triplets. However, it has limitations in dealing with complex relationship facts including 1-to- N , N -to-1, and N -to- N triplets due to the rigid translation principle.

TransH [6] tries to solve the problem of TransE by implementing an entity to have unique representations when the entity is involved in different relations. Specifically, for each triplet (h, r, t) , TransH projects h and t to a relation r specific hyperplane to obtain projected vectors $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^T \mathbf{h} \mathbf{w}_r$ and $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^T \mathbf{t} \mathbf{w}_r$. The scoring function is defined as $f_r(h, t) = \|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_{1/1/2}$.

TransR/CTransR [7] models entities and relations in different vector spaces, respectively, i.e., the entity vector space and the relation vector space. For each relation r , it set a projection matrix \mathbf{M}_r to map the vector of entities from the entity vector space to the relation vector space, i.e., $\mathbf{h}_r = \mathbf{h} \mathbf{M}_r$ and $\mathbf{t}_r = \mathbf{t} \mathbf{M}_r$. Its scoring function is $f_r(h, t) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_{1/1/2}$.

TransD [8] considers the diversity of entities and relations simultaneously. It uses the product of two vectors of an

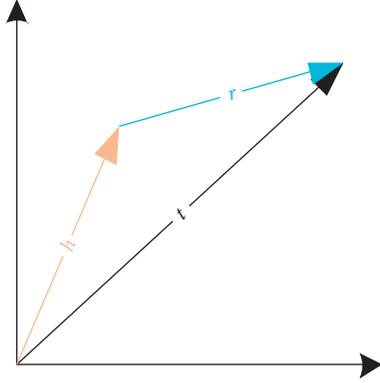


FIGURE 1: A relational fact translating example of TransE.

entity-relation pair to replace a projection matrix, i.e., $\mathbf{M}_{rh} = \mathbf{r}_p \mathbf{h}_p$ and $\mathbf{M}_{rt} = \mathbf{r}_p \mathbf{t}_p$. TransD is more extensible and can be applied to the large-scale knowledge graphs. Its scoring function is $f_r(h, t) = \|\mathbf{h}\mathbf{M}_{rh} + \mathbf{r} - \mathbf{t}\mathbf{M}_{rt}\|_{l1/l2}$.

TranSparse [9] considers the heterogeneity and imbalance of entities and relations in a knowledge graph, which are generally ignored by previous works. TranSparse constructs adaptive sparse matrices $\mathbf{M}_r^h(\theta_r^h)$ and $\mathbf{M}_r^t(\theta_r^t)$, instead of projection matrices, to concurrently prevent the overfitting of simple relational facts and the underfitting of complex relational facts. Its scoring function is $f_r(h, t) = \|\mathbf{M}_r^h(\theta_r^h)\mathbf{h} + \mathbf{r} - \mathbf{M}_r^t(\theta_r^t)\mathbf{t}\|_{l1/l2}$.

FT [18] and DT [19] design the flexible translation principles and the dynamic translation principles, respectively. To some extent, they improve the ability to handle complex relational facts. Essentially, they focus on elaborating various translation principles to learn complex relational facts more accurately. TransAt [17] and GAN-based framework [20] use attention mechanisms and generate adversarial networks to improve the model performance, respectively. However, all of the above methods naturally embed the complete knowledge graph into a uniform vector space from a single perspective, failing to solve the space congestion problem thoroughly.

PuTransE [21] is an online and robust improvement of TransE solving the hyperparameters sensitivity problem and the spatial congestion of entity and relation, as well as the processing of dynamic knowledge graphs. It adopts multiple parallel spaces to learn the vectors of entities and relations, thus avoiding spatial congestion in complex relational facts. Therefore, puTransE achieves state-of-the-art performance on the link prediction task. However, puTransE still has two weaknesses resulting in the performance limitation. First, puTransE causes the excessive sparseness of simple relational facts during the random parallel space generation. It performs knowledge extraction including not only complex relational facts but also sparse simple relational facts. Thus, it probably cannot learn complete vector representations of simple relational facts via a single subgraph. Second, puTransE randomly selects local knowledge to construct multiple parallel spaces, which is prone to impair original facts of knowledge graphs. For example, there is a golden triplet, i.e., a positive sample, in the original graph which

cannot be consisted by entities and relation in any parallel spaces. This situation decreases the relational fact prediction accuracy of puTransE.

3. Our Method

In this section, we introduce the details of MvTransE, which embeds relational facts from global-view and local-view, respectively. The workflow of MvTransE mainly consists of three steps. The first step (Section 3.1) is to generate multiple parallel subgraphs so as to extract particular local relational facts of the knowledge graph accurately. The second step (Section 3.2) aims at discriminatively embedding the knowledge graph into multiple parallel spaces to acquire multiview representations of relational facts. The last step (Section 3.3) integrates multiple versions of knowledge representations, i.e., fuses local-view and global-view representations of entities and relations. Figure 2 presents a multiview knowledge learning process of MvTransE.

3.1. Subgraph Generation. The subgraph generation aims to extract local relational facts from different perspectives, so as to solve spatial congestion of entities by sparsely embedding entities and relations into different parallel vector spaces in the following graph embedding step. Therefore, we construct multiple parallel subgraphs based on different relations of the knowledge graph. Each subgraph mainly contains the local relational facts selected from a specific relation.

Initially, we give the definition of some related symbols in the subgraph generation process. We define a knowledge graph as $G = (E, R, T)$, where E and R denote an entity set and a relation set of graph G , respectively; $T \subseteq E \times R \times E$ represents a triplet set of G . And \mathbb{G}_{sub} is the final generated subgraph set, \mathcal{G}_i is a subgraph that $\mathcal{G}_i \in \mathbb{G}_{sub}$, and \mathcal{E}_i and \mathcal{R}_i represent an entity set and a relation set of \mathcal{G}_i , respectively. Algorithm 1 demonstrates the details of the subgraph generation, which mainly consists of two steps given in the following.

3.1.1. Semantics-Related Entity Selection. To accurately learn local relational facts, we first select relation relevant entities for a subgraph to ensure the semantic consistency of its knowledge as much as possible. That is, entities in a subgraph should be semantically related to each other based on a specific relation. We randomly select a relation r from relation set R then to generate entity set E_r , which consists of extracted entities interconnecting with r . Since E_r is generated via relation r ; therefore, r is deemed as the semantic center of the current subgraph \mathcal{G}_i .

3.1.2. Structure-Related Subgraph Expansion. In order to learn latent knowledge associating with r more comprehensively, we need to expand each subgraph according to the local graph structure of entities in the set E_r . This step ensures a generated subgraph containing semantic and structural features of local relational facts simultaneously. Specifically, we first randomly select an entity e_i from E_r as a

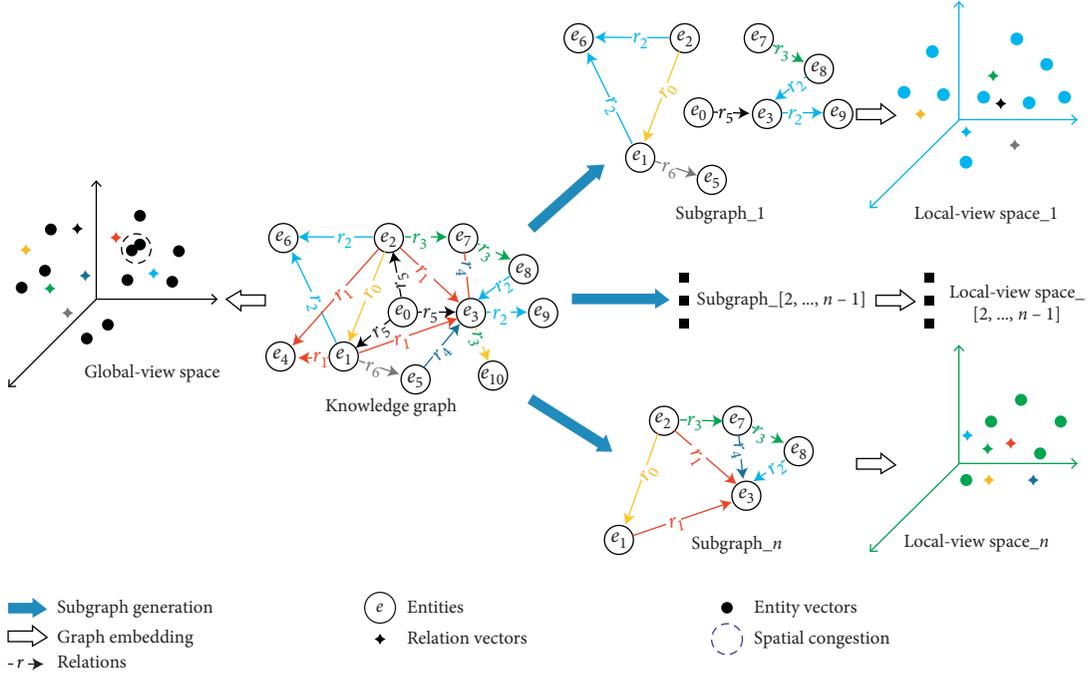


FIGURE 2: A multiview knowledge learning process of the MvTransE model. The blue arrow indicates a subgraph generation operation for extracting local relational facts from the knowledge graph; and the white arrow indicates a graph embedding process projecting the original knowledge graph and generated subgraphs into a global-view vector space and several local-view vector spaces, respectively. In each vector space, the colored coordinates represent different vector spaces. The solid circles represent entity vectors in a vector space whose color is the same as the color of the space coordinate. The four-angle stars represent relation vectors, and its color corresponds to the color of relations in the knowledge graph.

starting entity to expand the subgraph \mathcal{G}_i . And then, we randomly select a triplet whose head or tail entity is the starting entity e_i and add the head or tail entity to the subgraph \mathcal{G}_i . Consequently, we can get the local structure information of E_r regarding G by repeating the above two operations for n_s time. Due to the randomly selected entities and triplets in the subgraph expansion, each subgraph expanded from E_r may include different relational facts, which makes generated subgraphs slightly different from each other in terms of semantics and structure. Thereafter, with respect to learning local relational facts discriminately, these random operations ensure MvTransE can learn local knowledge representations from multiple perspectives.

Besides, to make each subgraph focusing on particular relational facts, we need to control the scale of subgraphs for avoiding extracting excessive irrelevant facts. We set hyperparameter n_t to control the number of triplets in subgraphs, set hyperparameter n_s to control the expansion speed of each subgraph, set hyperparameter τ to control the maximum iterations of the triplet selection, and set hyperparameter n to control the number of generated subgraphs.

3.2. Graph Embedding. The goal of this step is to obtain global-view representation and local-view representation of entities and relations. Hence, we perform original knowledge graph G embedding and subgraphs \mathcal{G} embedding to learn global knowledge and local knowledge, respectively. In each vector space, we define the following equation as the scoring function $f_r(h, t)$ to translate each triplet (h, r, t) :

$$f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{l_1/l_2}, \quad (1)$$

where \mathbf{h} , \mathbf{r} , and \mathbf{t} are vector representations of h , r , and t , and l_1/l_2 is the l_1 -norm or l_2 -norm distance.

In MvTransE, we use the margin-based loss function as the optimization target in each vector space, which is defined as follows:

$$L = \sum_{\Psi_i \in \Psi(h,r,t) \in T} \sum_{(h',r,t') \in T'} [f_r(h, t) + \gamma - f_r(h', t')]_+, \quad (2)$$

where Ψ is a set of embedded vector spaces, T is a set of positive triplets in a graph, and T' is a set of negative triplets generated by randomly replacing the head (or tail) of each positive triplet $(h, r, t) \in T$, and γ is a fixed margin distance for distinguishing positive and negative triplets. We use the stochastic gradient descent (SGD) [22] to minimize the loss function.

Algorithm 2 presents the multiview graph embedding process, which aims at respectively embedding each knowledge graph and subgraph into single vector spaces. Thus, we will get $n + 1$ vector spaces, including one global-view vector space and n local-view vector spaces. The global-view vector space obtains the global-view representation of all entities and relations regarding the original knowledge graph; the local-view vector spaces differentially learn local-view representation of entities and relations regarding complex relational facts from different semantic and structural perspectives.

Input: Knowledge Graph $G = (E, R, T)$, hyperparameters of subgraph n, τ
Output: A subgraphs set \mathbb{G}_{sub} .

- (1) $\mathbb{G}_{\text{sub}} \leftarrow$ Initialize empty set for subgraph
- (2) **while** $n > 0$ **do**
- (3) $\tau \leftarrow 128$ //Set the maximum iterations
- (4) $r \leftarrow$ Random sample a relation $r \in R$
- (5) $E_r \leftarrow$ Select semantics related entities
- (6) $n_s, n_t \leftarrow$ Generate subgraph scale hyperparameters
- (7) $\mathcal{E} \leftarrow \emptyset$ //Initialize empty set for selected triplets
- (8) **while** $|\mathcal{E}| < n_t$ and $\tau > 0$ **do**
- (9) $\text{count} \leftarrow 0$
- (10) **while** $\text{count} < n_s$ **do** //Control the scale of subgraphs
- (11) $e \leftarrow$ Randomly sample an entity $e \in E_r$
- (12) $t \leftarrow$ Randomly select a relevant triplet $t(e)$ from G
- (13) $\mathcal{E} \leftarrow \mathcal{E} \cup t$ //Add a selected triplet
- (14) $\text{count} \leftarrow \text{count} + 1$
- (15) **end while**
- (16) $E_r \leftarrow$ All entities collected in \mathcal{E}
- (17) $\tau \leftarrow \tau - 1$
- (18) **end while**
- (19) $\mathbb{G}_{\text{sub}} \leftarrow \mathbb{G}_{\text{sub}} \cup \mathcal{E}$ //Add a subgraph
- (20) $n \leftarrow n - 1$
- (21) **end while**

ALGORITHM 1: Generating subgraph.

Input: A set of Knowledge Graph and Subgraphs $\mathbb{G} = \{G, \mathcal{E}_1, \dots, \mathcal{E}_n\}$, vector dimension k , global-view margin γ_0 , global-view learning rate α_0 .
Output: A set of generated vector spaces ψ .

- (1) $\psi \leftarrow$ Initialize empty set for vector spaces
- (2) **for** $\mathcal{E} \in \mathbb{G}$ **do**
- (3) **if** $\mathcal{E} = G$ **then**
- (4) $\gamma, \alpha \leftarrow$ global-view margin γ_0 , global-view learning rate α_0
- (5) **else**
- (6) $\gamma, \alpha \leftarrow$ Randomly initialize local-view margin and learning rate
- (7) **end if**
- (8) $\mathcal{E}, \mathcal{R} \leftarrow$ All entities and relations in \mathcal{E} respectively
- (9) $\mathbf{e}, \mathbf{r} \leftarrow$ Initialize uniform $(-6\sqrt{k}, 6\sqrt{k})$ for each $e \in \mathcal{E}, r \in \mathcal{R}$
- (10) **loop**
- (11) $S_{\text{batch}} \leftarrow$ sample (S, b) //Sample a minibatch of size b
- (12) $T_{\text{batch}} \leftarrow \emptyset$ //Initialize the set of pairs of triplets
- (13) **for** $(h, r, t) \in S_{\text{batch}}$ **do**
- (14) $(h, r, t) \leftarrow$ sample $(S_{(h', r, t')})$ //Sample corrupted triplet
- (15) $T_{\text{batch}} \leftarrow T_{\text{batch}} \cup \{(h, r, t), (h', r, t')\}$
- (16) **end for**
- (17) Update vectors w.r.t

$$\sum_{((h, r, t), (h', r, t')) \in T_{\text{batch}}} \nabla [f_r(h, t) + \gamma - f_r(h', t')]_+$$
- (18) **end loop**
- (19) $\Delta \leftarrow (\mathcal{E}, \mathbf{e}, \mathbf{r})$ //Trained parameters are saved as one vector space
- (20) $\psi \leftarrow \psi \cup \Delta$ //Add vector space to sets
- (21) **end for**

ALGORITHM 2: Learning multiview graph embedding.

3.3. Multiview Fusion Strategy. In this section, we propose a multiview fusion strategy that adopts an adaptive selection principle to integrate knowledge representations of global-

view vector space and local-view vector space. For each testing triplet (h, r, t) , we define a scoring estimation function $S_r(h, t)$ to calculate the distance score of a vector

space, and then dynamically select the final representation of the triplet according to the minimum score. The scoring estimation function is defined as follows:

$$S_r(h, t) = \min_{\Delta \in \psi} \|\mathbf{h}\Delta + \mathbf{r}\Delta + \mathbf{t}\Delta\|_{l_1/l_2}, \quad (3)$$

where Δ is a vector space in ψ which contains h , r , and t ; $\mathbf{h}\Delta$, $\mathbf{r}\Delta$ and $\mathbf{t}\Delta$ are vectors of h , r and in a vector space Δ .

Since each parallel vector space generally contains local relational facts related to a particular relation, our model can subtly solve the spatial congestion problem. Additionally, MvTransE constructs a global-view vector space containing complete knowledge representations in a knowledge graph, which makes any testing triplet able to find a knowledge representation at least. Thus, our model significantly improves the performance of learning simple relational knowledge.

4. Experiments

In this section, we study the performance of our model in link prediction and triplet classification tasks under four public datasets, i.e., WN18, WN18RR, WN11, and FB15K-237.

4.1. Datasets. WordNet [2] is a large knowledge graph of English vocabulary which is widely used in graph embedding works. In WordNet, a set of synonyms representing a basic vocabulary concept is taken as an entity, and various semantic relations are established between these synonym sets. In the following experiments, we use three public subsets of WordNet, i.e., WN18, WN18RR, and WN11. WN18 contains 18 relations and 40943 entities. WN18RR is a modified version of WN18 introduced by Dettmers et al. [23], which removes the reversing relational facts avoiding information leakage problem in representation tasks. WN11 consists of 11 relations and 38696 entities. Freebase is a large collaborative knowledge graph storing the general facts of the real world. We use a subset of Freebase, i.e., FB15k-237 [21], which consists of 237 relations and 14541 entities in total. Table 1 presents the statistics of the above datasets.

4.2. Link Prediction. Link prediction aims to predict the missing head entity h or tail entity t of a test triplet (h, r, t) . In this experiment, we take the entity h (or t) missed in test triplets as the correct entity, and all other entities are considered as candidate entities. Firstly, we construct candidate triplets by replacing h (or t) of the test triplet. Then, the link prediction score of each triplet is calculated by the scoring function of our model. Finally, candidate entities and the correct entity are sorted in ascending order based on their prediction scores. We adopt two metrics used in [5] to evaluate our model: the average rank of each correct entity (i.e., Mean Rank) and the average number of correct entities ranked in the top 10 (i.e., Hits@10). Obviously, a good prediction performance should achieve a high Hits@10 and a low Mean Rank.

TABLE 1: Dataset information.

Dataset	#Ent	#Rel	#Train	#Valid	#Test
WN18	40943	18	141442	5000	5000
WN11	38696	11	112581	2609	10544
WN18RR	40943	11	86835	3034	3134
FB15K-237	14541	237	272115	17535	20466

Note that the candidate triplets may already exist in the knowledge graph, so these candidate triplets should be considered as the correct triplets. The scores of these candidate triplets are likely to be lower than the correct triplets. Therefore, we should filter out these candidate triplets that have already appeared in the train, validation, and test sets. We denote an evaluation setting by ‘‘Filt’’ if we filter out these candidate triplets before the test, otherwise denote it by ‘‘Raw.’’

We compare MvTransE with a few state-of-the-art methods in the link prediction task on WN18, WN18RR, and FB15K-237 datasets. On WN18, MvTransE is compared to RESCAL [24], SE, SME, LFM, TransE, TransH, TransR/CTransR, puTransE, and TransAt in Table 2. In Table 3, we compare MvTransE with three competitive methods DistMult [25], ComplEx [26], and ConvE [23] on WN18RR and FB15K-237 datasets both of which do not have information leakage problem found on WN18 dataset. We directly use the results reported in their published papers or in [7] due to the same experimental settings. For MvTransE, our experimental settings are as follows.

4.2.1. Subgraph Generation Setup. On WN18, we set the number of subgraphs $n = 5000$, the expanding speed of subgraphs $n_s \in [50, 450]$, and the size of subgraphs $n_t \in [200, 1500]$. On WN18RR, we set the number of subgraphs $n = 5000$, the expanding speed of subgraphs $n_s \in [50, 300]$, and the size of subgraphs $n_t \in [200, 1000]$. On FB15K-237, we set the number of subgraphs $n = 5000$, the expanding speed of subgraphs $n_s \in [100, 350]$, and the size of subgraphs $n_t \in [1000, 2200]$.

4.2.2. Graph Embedding Setup. The dimension of vectors k is set among $\{25, 30, 35, 40, 50\}$, the training phase *epoch* among $\{200, 500, 1000\}$. In global-view vector space, we set the learning rate α_0 among $\{0.01, 0.005, 0.001, 0.0008\}$, the margin value γ_0 among $\{2, 3, 3.5, 4, 4.5, 5, 5.5\}$, the minibatch size B_0 among $\{100, 200, 500\}$. For each dataset, we choose the best parameter configuration through validation sets and they are as follows: $k = 35$, $\text{epoch} = 1000$, $\alpha_0 = 0.005$, $\gamma_0 = 3.5$, $B_0 = 100$ on WN18; $k = 35$, $\text{epoch} = 1000$, $\alpha_0 = 0.0008$, $\gamma_0 = 5.5$, $B_0 = 100$ on WN18RR; $k = 35$, $\text{epoch} = 1000$, $\alpha_0 = 0.001$, $\gamma_0 = 3$, $B_0 = 100$ on FB15K-237. In local-view spaces, we randomly initialize the learning rate α_i and the margin γ_i within a certain range, and obtain the best parameter settings as follows: $\alpha_i \in [0.001, 0.005]$, $\gamma_i \in [1.5, 5.5]$ on WN18; $\alpha_i \in [0.0008, 0.005]$, $\gamma_i \in [3.5, 5.5]$ on WN18RR; $\alpha_i \in [0.005, 0.015]$, $\gamma_i \in [3.5, 5.5]$ on FB15K-237.

Table 2 presents the corresponding experimental results of three model settings: (1) MvTransE (Global-view) denotes

TABLE 2: Experimental results of link prediction on WN18.

Dataset	WN18			
	Mean rank		Hits@10 (%)	
Metric	Raw	Filt	Raw	Filt
RESCAL	1180	1163	37.2	52.8
SE	1011	985	68.5	80.5
SME (linear/bilinear)	545/526	533/509	65.1/54.7	74.1/61.3
LFM	469	456	71.4	81.6
TransE	263	251	75.4	89.2
TransH (unif/bern)	318/401	303/388	75.4/73.0	86.7/82.3
TransR (unif/bern)	232/238	219/225	78.3/79.8	91.7/92.0
CTransR (unif/bern)	243/231	230/218	78.9/79.4	92.3/92.3
puTransE	39	29	88.1	94.9
TransAt	169	157	81.4	95.1
MvTransE (global-view)	198	186	79.7	92.2
MvTransE (local-view)	46	42	84.1	93.9
MvTransE (Multiview)	29	24	88.3	95.1

TABLE 3: Results of link prediction on WN18RR and FB15K-237 under the Filt setting.

Dataset	WN18RR		FB15K-237	
	Mean Rank(Filt)	Hits@10 (filt)	Mean Rank(Filt)	Hits@10 (filt)
DisMult	5110	0.49	254	0.41
ComplEx	5261	0.51	339	0.42
ConvE	5277	0.46	246	0.49
MvTransE	1323	0.52	139	0.49

the prediction results by using the global-view representation. (2) MvTransE (Local-view) denotes the prediction results by using all parallel local-view representation. (3) MvTransE (Multiview) denotes the prediction results of the integrated multiview representation derived from a multiview fusion strategy. It can be seen from Table 2, the multiview representation yields better results than global-view and local-view representations. The results prove that our idea of representing knowledge from multiple perspectives is effective. In detail, our method substantially outperforms state-of-the-art methods in the Mean Rank metric. In Hits@10, our method is also superior to all baseline methods and achieves the best performance under “Raw,” and achieves the same best performance as TransAt under “Filt” settings.

Table 3 presents the results of MvTransE (Multiview) on WN18RR and FB15K-237 to further illustrate the merits of our method. Obviously, MvTransE (Multiview) markedly outperforms all methods on WN18RR, achieving state-of-the-art performance in two metrics. On FB15K-237, MvTransE (Multiview) has achieved the best performance over all of the methods on the Mean Rank metric, performing the same as ConvE on Hits@10 metric. Particularly, our model achieves excellent performance on all three different datasets in the Mean Rank metric, which evaluates the overall quality of the learned knowledge representations. This is because that MvTransE aims at learning knowledge representations from multiple perspectives and dynamically fusing these representations as an optimal combination.

To further explain the above observation, we present the prediction results of our method regarding all types of

TABLE 4: Type distribution of triplet.

Relation category	1-to-1	1-to- N	N -to-1	N -to- N
#Triplet (train)	1281	54655	54555	31014
#Triplet (valid)	44	1938	1897	1119
#Triplet (test)	42	1847	1981	1130

relational facts of WN18. Table 4 lists the type distribution of triplets in WN18 based on four relation categories. Table 5 presents the experimental results of three model settings on each relation category. Specifically, the global-view setting outperforms the local-view setting in predicting 1-to-1, 1-to- N head, and N -to-1 tail on both metrics, which actually fall into the simple relational facts learning category. On the contrary, the local-view setting exhibits superior performance in predicting complex relational facts, i.e., including N -to- N , N -to-1 head, and 1-to- N tail facts learning category. Clearly, by combining the advantages of global-view and local-view settings, the multiview method achieves the best performance in learning simple and complex relational facts.

4.3. Triplet Classification. Triple classification task aims to determine whether a given triple (h, r, t) is correct or not. In this experiment, we adopt WN11 to verify the effectiveness of our method. Following the experiment setting of previous work [5], we set a classifying threshold δ_r for each relation r . To maximize the classification accuracy, we optimize δ_r on the validation set. Giving a test triple (h, r, t) , if its score is lower than δ_r , it will be classified as a positive sample, otherwise a negative sample.

TABLE 5: Experimental results of WN18 by mapping properties of relations.

Tasks ($k = 35$)		Predicting head				Predicting tail			
		1-to-1	1-to- N	N -to-1	N -to- N	1-to-1	1-to- N	N -to-1	N -to- N
Global-view	Hits@10 (%)	90.5/90.5	91.9/92.7	56.0/87.6	85.7/89.7	92.9/93.2	54.5/86.9	94.1/94.2	84.6/88.6
(Raw/Filt)	Mean rank	12/12	158/158	228/198	217/216	35/35	258/228	132/132	227/225
Local-view	Hits@10 (%)	71.4/71.4	91.8/92.0	70.3/86.8	94.6/95.0	71.4/71.4	78.4/91.7	84.4/84.4	94.3/94.8
(Raw/Filt)	Mean rank	97/97	23/23	72/61	22/21	177/177	44/33	67/67	23/23
Multiview	Hits@10 (%)	90.5/90.5	94.2/94.3	76.1/94.2	95.0/95.4	92.9/93.9	79.8/93.9	94.1/94.2	94.8/95.2
(Raw/Filt)	Mean rank	8/8	12/11	47/34	39/38	20/19	44/31	14/14	23/22

The bold values stand for the best performance in the corresponding metric.

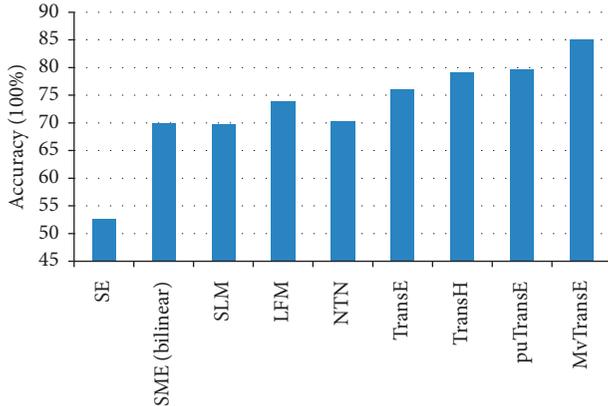


FIGURE 3: The experimental results of triplet classification.

We choose SE, SME, Single Layer Model (SLM) [17], LFM, NTN, TransE, TransH, and puTransE as baselines. We use the results reported in [7] directly since the data set is the same. For MvTransE, our experimental settings are as follows.

4.3.1. Subgraph Generation Setup. We set the number of subgraphs $n = 5000$, the length of random walk $n_s \in [50, 300]$, and the size of subgraphs $n_t \in [200, 1000]$.

4.3.2. Graph Embedding Setup. The dimension of vectors k is set among $\{15, 20, 25\}$, the training phase *epoch* among $\{200, 500, 1000\}$. In global-view vector space, we set the learning rate α_0 among $\{0.02, 0.01, 0.05\}$, the margin value γ_0 among $\{3, 4, 4.5, 5\}$, and the minibatch size B_0 among $\{100, 200, 500\}$. We choose the best configuration by the validation set, which is as follows: $k = 20$, *epoch* = 1000, $\alpha_0 = 0.01$, $\gamma_0 = 4.5$, $B_0 = 100$. In local-view vector spaces, we randomly initialize the learning rate $\alpha_i \in [0.005, 0.015]$, the margin value $\gamma_i \in [3.5, 5.5]$.

The experimental results of the triplet classification are shown in Figure 3. Clearly, MvTransE obtains the best performance among all of the baseline methods. Compared with the translation-based methods, i.e., TransE and TransH, our method is able to learn complex relational facts more subtly by constructing subgraphs from the original knowledge graph. On the other side, our method outperforms the recent competitor puTransE due to our knowledge fusion strategy which leverages an adaptive selection

principle to integrate the global-view and the local-view knowledge representations reasonably. Therefore, MvTransE is more suitable for embedding large and complex knowledge graphs. MvTransE has great advantages in knowledge graph completion tasks due to its multiview knowledge learning and fusing methods.

5. Conclusion and Future Work

In this work, we propose a multiview translation learning model, named MvTransE, which aims at presenting graph relational facts from global-view and local-view, respectively. MvTransE achieves state-of-the-art performance by solving the entity spatial congestion problem and the relational fact impairment problem. Extensive experiments demonstrate that MvTransE outperforms state-of-the-art models on the link prediction task and triplet classification task. In the future, we will focus on the subgraph construction scheme to learn the local relational facts more efficiently.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Nos. U1811264, 61966009, and U1711263), the Natural Science Foundation of Guangxi Province (Nos. 2020GXNSFAA159055, 2019GXNSFBA245059, and 2019GXNSFBA245049), the Guangxi Innovation-Driven Development Project (No. AA17202024), and the Beihai Science and Technology Project Contract (No. 202082001).

References

- [1] R. Ronald Bakker, “Knowledge graphs: representation and structuring of scientific knowledge,” 1987.
- [2] G. A. Miller, “WordNet: a lexical database for English,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [3] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: a collaboratively created graph database for

- structuring human knowledge,” in *Proceedings of the Special Interest Group on Management of Data*, pp. 1247–1250, Vancouver, Canada, June 2008.
- [4] X. Dong, E. Gabrilovich, G. Heitz et al., “Knowledge vault: a web-scale approach to probabilistic knowledge fusion,” in *Proceedings of the Special Interest Group on Knowledge Discover and Data Mining*, pp. 601–610, New York, NY, USA, August 2014.
- [5] B. Antoine, N. Usunier, A. Garc’ia-Dur’an, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Proceedings of Neural Information Processing Systems*, pp. 2787–2795, Lake Tahoe, NV, USA, December 2013.
- [6] Z. Wang, J. Zhang, J. Feng, and Z. Chen, “Knowledge graph embedding by translating on hyperplanes,” in *Proceedings of Association for the Advances of Artificial Intelligence*, pp. 1112–1119, Québec City, Canada, July 2014.
- [7] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, “Learning entity and relation embeddings for knowledge graph completion,” in *Proceedings of Association for the Advances of Artificial Intelligence*, pp. 2181–2187, Austin, TX, USA, January 2015.
- [8] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, “Knowledge graph embedding via dynamic mapping matrix,” in *Proceedings of the Association for Computer Linguistics*, pp. 687–696, Beijing, China, July 2015.
- [9] G. Ji, K. Liu, S. He, and J. Zhao, “Knowledge graph completion with adaptive sparse transfer matrix,” in *Proceedings of the Association for the Advances of Artificial Intelligence*, pp. 985–991, Phoenix, AZ, USA, February 2016.
- [10] Yi Tay, A. T. Luu, and S. C. Hui, “Non-parametric estimation of multiple embedding for link prediction on dynamic knowledge graphs,” in *Proceedings of the Association for the Advances of Artificial Intelligence*, pp. 1243–1249, San Francisco, CA, USA, February 2017.
- [11] X. Tian Xia, D. Dacheng Tao, T. Tao Mei, and Y. Yongdong Zhang, “Multiview spectral embedding,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 6, pp. 1438–1446, 2010.
- [12] D. Zhai, H. Chang, S. Shan, X. Chen, and W. Gao, “Multiview metric learning with global consistency and local smoothness,” *ACM Transactions on Intelligent Systems and Technology*, vol. 3, no. 3, pp. 1–22, 2012.
- [13] B. Antoine, J. Weston, R. Collobert, and Y. Bengio, “Learning structured embeddings of knowledge bases,” in *Proceedings of the Association for the Advances of Artificial Intelligence*, pp. 301–306, San Francisco, CA, USA, February 2011.
- [14] B. Antoine, X. Glorot, J. Weston, and Y. Bengio, “Joint learning of words and meaning representations for open-text semantic parsing,” in *Proceedings of the Artificial Intelligence and Statistics*, pp. 127–135, Canary Islands, Spain, April 2012.
- [15] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, “A semantic matching energy function for learning with multi-relational data,” *Machine Learning*, vol. 94, no. 2, pp. 233–259, 2014.
- [16] R. Jenatton, N. Le Roux, B. Antoine, and G. Obozinski, “A latent factor model for highly multi-relational data,” in *Proceedings of the Neural Information Processing Systems*, pp. 3176–3184, Lake Tahoe, NV, USA, December 2012.
- [17] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, “Reasoning with neural tensor networks for knowledge base completion,” 2013.
- [18] J. Feng, M. Huang, M. Wang, M. Zhou, H. Yu, and X. Zhu, “Knowledge graph embedding by flexible translation,” in *Proceedings of the Knowledge Representation and Reasoning*, pp. 557–560, Cape Town, South Africa, April 2016.
- [19] L. Chang, M. Zhu, T. Gu, C. Bin, J. Qian, and J. Zhang, “Knowledge graph embedding by dynamic translation,” *IEEE Access*, vol. 5, pp. 20898–20907, 2017.
- [20] P.F. Wang, S. Li, and R. Pan, “Incorporating gan for negative sampling in knowledge representation learning,” in *Proceedings of Association for the Advances of Artificial Intelligence*, pp. 2005–2012, New Orleans, LO, USA, February 2018.
- [21] K. Toutanova and D. Chen, “Observed versus latent features for knowledge base and text inference,” in *Workshop on Continuous Vector Space and their Compositionality*, pp. 57–66, Beijing, China, July 2015.
- [22] J. C. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.
- [23] Tim Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, “Convolutional 2D knowledge graph embeddings,” in *Proceedings of Association for the Advances of Artificial Intelligence*, pp. 1811–1818, New Orleans, LO, USA, February 2018.
- [24] M. Nickel, V. Tresp, and H. P. Kriegel, “A three-way model for collective learning on multi-relational data,” in *Proceedings of International Conference on Machine Learning*, pp. 809–816, Bellevue, WA, USA, February 2011.
- [25] B. Yang, W.-T. Yih, X. He, J. Gao, and Li Deng, “Embedding entities and relations for learning and inference in knowledge bases,” in *Proceedings of International Conference on Learning Representations*, pp. 1–13, San Diego, CA, USA, December 2015.
- [26] Th’eo Trouillon, J. Welbl, S. Riedel, E. ’ric Gaussier, and G. Bouchard, “Complex embeddings for simple link prediction,” in *Proceedings of International Conference on Machine Learning*, pp. 2071–2080, New York, NY, USA, June 2016.