TransMS: Knowledge Graph Embedding for Complex Relations by Multidirectional Semantics

Shihui Yang[1,2] , **Jidong Tian**[1,2] , **Honglun Zhang**[1,2] , **Junchi Yan**[2,3] , **Hao He**[1,2] and **Yaohui Jin**[1,2,*]
[1]State Key Lab of Advanced Optical Communication System and Network, Shanghai Jiao Tong University
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[3]Department of Computer Science and Engineering, Shanghai Jiao Tong University
{ysh_sjtu,frank92,zhanghonglun,yanjunchi,hehao,jinyh}@sjtu.edu.cn

Abstract

Knowledge graph embedding, which projects the symbolic relations and entities onto low-dimension continuous spaces, is essential to knowledge graph completion. Recently, translation-based embedding models (e.g. TransE) have aroused increasing attention for their simplicity and effectiveness. These models attempt to translate semantics from head entities to tail entities with the relations and infer richer facts outside the knowledge graph. In this paper, we propose a novel knowledge graph embedding method named **TransMS**, which *translates and transmits multidirectional semantics*: i) the semantics of head/tail entities and relations to tail/head entities with *nonlinear functions* and ii) the semantics from entities to relations with *linear bias vectors*. Our model has merely one additional parameter α than TransE for each triplet, which results in its better scalability in large-scale knowledge graph. Experiments show that TransMS achieves substantial improvements against state-of-the-art baselines, especially the Hit@10s of head entity prediction for N-1 relations and tail entity prediction for 1-N relations improved by about 27.1% and 24.8% on FB15K database respectively.

1 Introduction

Knowledge graph, which is a semantic graph composed of triplets, has become a hot topic for its wide applications in information retrieval. As the basic unit for knowledge graph, the triplet in the form of $(head, relation, tail)$ – also abbreviated as (h, r, t), consists of two entities (i.e., $head$ and $tail$) as nodes and one relation $(relation)$ as edge from $head$ to $tail$. Although knowledge graphs can consist of millions of entities and billions of relations for the facts of the world, they are still far from completeness, which results in that the prediction for new relations between entities according to the existing triplets in knowledge graph especially for the complex relations is an indispensable task of knowledge completion. Knowledge graph embedding has recently been regarded as an effective way of knowledge graph completion. In particular, the seminal model TransE [Bordes *et al.*, 2013] has triggered increasing research in translation-based models for their simplicity and effectiveness. TransE first projects entities and relations onto low-dimension continuous spaces, formally as $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$ and then translates semantics from head to tail entities by relations, which requires $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when triplet (h, r, t) holds. However, such a constraint is too strict to be practical in dealing with the complex relations of 1-N, N-1 and N-N, and followup methods based on translation have been devised as improvements to overcome the shortcoming of TransE.

Specifically, TransH [Wang *et al.*, 2014] interprets each relation as a translating operation on a hyperplane with a translation vector and a norm vector, and projects the entity vectors onto the hyperplane by the norm vector. However, TransR [Lin *et al.*, 2015] supports that various relations focus on different aspects of entities which have multiple aspects. TransR first projects entities and relations onto entity spaces and relation spaces respectively, and then translates entities to relation spaces by a relational projection matrix. The method TransD [Ji *et al.*, 2015] is based on the assumption that projection matrices of TransR should not be only related to relations but also to entities. While TranSparse [Ji *et al.*, 2016] focuses on the *heterogenenous* which means some relations only link a few entity pairs but others not, and the *unbalanced* which means relations may link many head/tail entities and fewer tail/head entities, so that TranSparse sets parameters related the number of entities to the projection matrix. Gtrans [Tan *et al.*, 2018], which does not believe that Trans (E, H, R and D) fully consider the complicacy of the entities, represents different semantic aspects of entities and relation respectively with two vectors.

Issues: Although these methods show improvements compared with previous methods, they have difficulty in dealing with the link prediction for complex relations, which we attribute to the following reasons:

i) They transform the entities embedding only by transmitting the semantics from relations to entities but not transmitting the semantics from head/tail entities to tail/head entities at the same time. Take Figure 1 for example, there are three right triplets: $(tiger, eat, meat)$, $(people, eat, meat)$ and $(people, eat, fruit)$. These methods just transmit semantics from eat to $people$, $tiger$, $fruit$ and $meat$ while from

[*] Yaohui Jin is corresponding author.

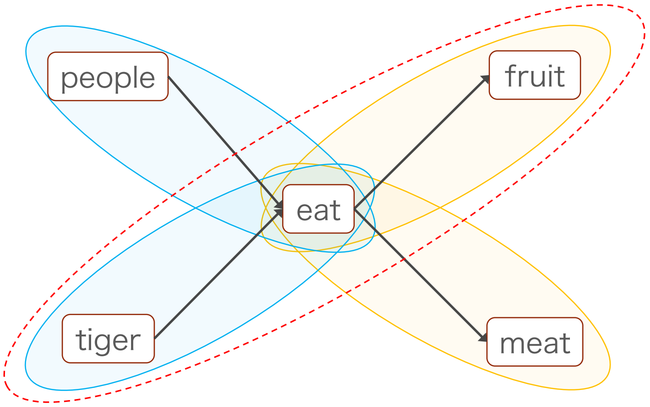

Figure 1: Sketch illustration of previous translation-based models. There are three right triplets: (*people, eat, fruit*), (*people, eat, meat*) and (*tiger, eat, meat*), and a wrong triplet: (*tiger , eat, fruit*). The semantic information is transmitted between entities and relation, i.e. *people* and *eat*, *tiger* and *eat*, *eat* and *fruit*, *eat* and *meat*, but not between head and tail entities nor from entities to relations.

the human cognition perspective, there are semantic transmissions between $people$ and $fruit$, $people$ and $meat$, $tiger$ and $fruit$, $tiger$ and $meat$, which may result in a wrong inference: $(tiger, eat, fruit)$.

ii) They do not transform the relation embedding by transmitting the semantics from entities to relations neither. As in some academic, the triplets (h, r, t) could be regarded as the grammar structure of (subject, predicate, object), i.e. relation is the predicate owning an executor h and an enforcer t for each triplet, which means there are semantic transmissions from subject (h) and object (t) to predicate (r).

iii) They show poor scalability to the large-scale knowledge graph for having much more parameters than TransE. They usually have some additional vectors or matrices than TranE for each triplet, such as TransR having a additional projection matrix for each relation.

iv) They transmit the semantics by linear transformation, which might be limited and shall be replaced by nonlinear transformation for better semantics translation.

v) From the mathematical point of view, when head/tail entity vector and relation vector are fixed, the left/right side of $\mathbf{h}_\perp + \mathbf{r} = \mathbf{t}_\perp$ is fixed, i.e. no matter how linearly transform to get tail/head entity vector, the final tail/head entity vectors are distributing around one center, resulting in that tail/head entity vectors are close to each other when head/tail entity vector and relation vector are fixed.

To address these issues, we propose a novel method named TransMS which has merely one additional parameter α than TransE for each triplet as illustrated in Figure 2. First, entities are projected as k_e-dimension vectors $\mathbf{h}, \mathbf{t} \in \mathbb{R}^{k_e}$, relation as k_r-dimension vectors $\mathbf{r} \in \mathbb{R}^{k_r}$ and a one-dimension variable α where $k_e = k_r$ for each triplet (h, r, t) in our model. Then, we transform the tail entity embedding by transmitting the semantics from head entity h and relation r to tail entity t by a nonlinear function $p(\mathbf{h}, \mathbf{r})$, while transform the head entity embedding by transmitting the semantics from tail entity t and relation r to head entity h by $p(-\mathbf{t}, \mathbf{r})$ because it is in the opposite direction of relation. Due to the nonlinear transmissions from relation to entities, we just add a bias vector to the relation embedding $\mathbf{r}$ to prevent the nonlinear transmissions from canceling each other out, where the bias vector is $\alpha \cdot g(\mathbf{h}, \mathbf{t})$ related to both $\mathbf{h}$ and $\mathbf{t}$. Therefore, our model not only conforms to human cognition in semantics, but also overcomes the shortcoming of one entity's vector learning around a center when the other entity and relation of a triplet are fixed in previous models.

Contribution: In general, the highlights of the paper are:

i) We propose a novel model TransMS that considers the semantic information not only from relations to entities, but also from the entities to relations and between entities;

ii) Our model has merely one additional parameter α than the simplest pioneered model (i.e. TransE) in contrast to peer methods having much more parameters for each triplet, such as TransR which has $k_e \times k_r$ more parameters than TransE, so our model has better scalability than the others;

iii) Our model has notable improvement on link prediction for complex relations against the baselines model, especially the Hit@10 for the head entity prediction for N-1 relations and the tail entity prediction for 1-N relations, which are improved by about 27.1% and 24.8% on FB15k [Bollacker *et al.*, 2008], respectively.

2 Related Work

To facilitate the introduction of relate works, we briefly define notations used in this paper. We denote a triplet by (h, r, t), which belongs to knowledge graph, and their embedding vectors by the bold lower case letters $(\mathbf{h}, \mathbf{t}, \mathbf{r})$; matrices by the bold upper case letters, such as $\mathbf{M}$, and tensor whose dimension is no less than three by underlining the bold upper case letters, such as $\underline{\mathbf{W}}$. Score function is denoted by $f_r(\mathbf{h}, \mathbf{t})$.

2.1 Translation-Based Methods

TransE [Bordes *et al.*, 2013]. As the seminal work for translation-based model, TransE first projects the triplets onto the low-dimension continuous space as $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$, and then translates the semantics from head entities to tail entities by relations, formally as $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. Hence, the score function is $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{\ell_{1/2}}$. TransE is effective for simple 1-1 relations, but its strict constraint makes it less capable on complex relations (i.e. 1-N, N-1 and N-N).

TransH [Wang *et al.*, 2014]. In order to overcome the shortcoming of TransE, TransH projects the entity embeddings $\mathbf{h}$ and $\mathbf{t}$ onto a relation-specific hyperplane to get $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_\mathbf{r}^\top \mathbf{h}\mathbf{w}$ and $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_\mathbf{r}^\top \mathbf{t}\mathbf{w}$ where $\mathbf{w}$ is a hyperplane normal vector, while the relation embedding is denoted as a translation vector $\mathbf{r}$ and a norm vector $\mathbf{w}$. In this way, TransH enables every entity to have distributed embedding vectors corresponding to different relations. Finally, the score function of TransH turns into $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_{\ell_{1/2}}$.

TransR/CTransR [Lin *et al.*, 2015]. TransE and TransH just project the entities and relations onto the same semantic space but not consider that every entity may have multiple attributes and different relations may focus on different attributes of entities. In this connection, TransH first

Dataset	#Rel	#Ent	#Train	#Valid	#Test
FB15K	1,345	14,951	483,142	50,000	59,071
FB15K-237	237	14,541	272,115	17,535	20,466
WN18	18	40,943	141,442	5,000	5,000
WN18RR	11	40,943	86,835	3,034	3,134

Table 1: Statistics of datasets used in experiments.

projects the entities onto the entity embedding space different from the relation space onto which the relations are projected, and then the entity embedding vectors are projected onto the relation space by projection matrix $\mathbf{M}_r$. Finally, the score function of TransR/CTransR turns into $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{M_r h} + \mathbf{r} - \mathbf{M_r t}\|_{\ell_{1/2}}$.

TransD [Ji *et al.*, 2015]. Considering the diversity of both entities and relations at the same time, TransD sets vector-vector multiplications to replace the projection matrix of TransR formally as $\mathbf{M}_h = \mathbf{r}_p \mathbf{h}_p^\top + \mathbf{I}^{k_r \times k_e}$ for the head projection matrix where k_r and k_e are the dimension of relation and entity embedding vectors while the $\mathbf{r}_p$ is related to relation r and $\mathbf{h}_p$ is related to the head entity h, so that TranD has fewer parameters and better scalability for large-scale knowledge graph.

TranSparse [Ji *et al.*, 2016]. In consideration of the heterogeneity of different relations and the unbalanced quantities of head and tail entities for each relation, TranSparse replaces the projection matrices with adaptive sparse matrices whose sparse degrees are determined by the number of entity pairs or entities linked by relations. Its score function turns into $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{M}_r^h(\theta_r^h)\mathbf{h} + \mathbf{r} - \mathbf{M}_r^t(\theta_r^t)\mathbf{t}\|_{\ell_{1/2}}$, where $\mathbf{M}_r^h(\theta_r^h)$ and $\mathbf{M}_r^t(\theta_r^t)$ are adaptive sparse matrices for head entity h and tail entity t going with relation r in each triplet (h, r, t) respectively. Here θ_r^h and θ_r^h are their sparse degrees.

GTrans [Tan *et al.*, 2018]. GTrans supports that Trans (E, H, R and D) underestimate the complicacy of entities and do not fully describe semantics of entities and that relation is an abstraction of entity facts (h, t) as these models following the similar score function $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_{\ell_{1/2}}$. GTrans projects each entity onto two states - *eigenstate* and *minesis* which reflect its *intrinsic* and *exogenous* attributes respectively, formally as $\mathbf{h_a}, \mathbf{h_e}, \mathbf{t_a}, \mathbf{t_e}, \mathbf{r_a}, \mathbf{r_e} \in \mathbb{R}^k$.

2.2 Other Methods

Structured Embedding (SE) [Bordes *et al.*, 2011]. For each relation r in triplet (h, r, t), SE projects head entities and tail entities by two different matrices $\mathbf{M}_r^h$ and $\mathbf{M}_r^t$ respectively. Its score function is $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{M}_r^h\mathbf{h} - \mathbf{M}_r^t\mathbf{t}\|_{\ell_{1/2}}$.

Semantic Matching Energy (SME) [Bordes *et al.*, 2012; Bordes *et al.*, 2014]. SME first projects entities and relations onto low-dimension spaces, and then constructs a neural network which acquires correlations between entities and relations by matrix operations including the linear form whose score function is $f_r(\mathbf{h}, \mathbf{t}) = -(\mathbf{M}_{h_1}\mathbf{e}_h + \mathbf{M}_{h_2}\mathbf{r} + \mathbf{b}_h)^\top(\mathbf{M}_{t_1}\mathbf{e}_t + \mathbf{M}_{r_2}\mathbf{r} + \mathbf{b}_t)$ and the bilinear form whose score function is $f_r(\mathbf{h}, \mathbf{t}) = -[(\mathbf{M}_{h_1}\mathbf{e}_h) \otimes (\mathbf{M}_{h_2}\mathbf{r}) + \mathbf{b}_h]^\top[(\mathbf{M}_{t_1}\mathbf{e}_t) \otimes (\mathbf{M}_{t_2}r) + \mathbf{b}_t]$ and $\otimes$ is Hadamard product.

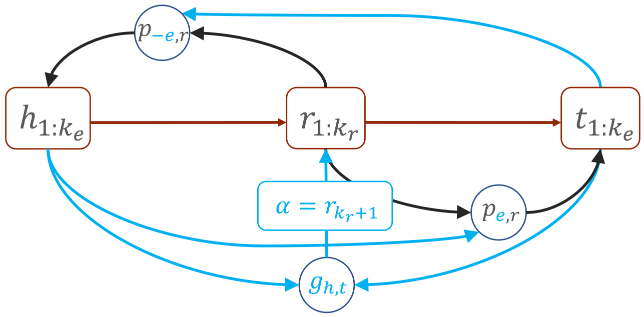

Figure 2: The illustration of our model. $p_{e,r}$ is the semantic translation function where r represents the semantics of relations, and e is inexistent or represents the semantic information of head (or tail) entities when the semantics of relations are transmitted to head (or tail) entities in previous models, while e represents the semantics of head (or tail) entities transmitted to tail (or head) entities in our model. The operation between $g_{h,t}$ and α is the bias vectors to transmit the semantics from head and tail entities to relations.

Latent Factor Model (LFM) [Jenatton *et al.*, 2012]. LFM projects entities as column vectors $(\mathbf{h}, \mathbf{t})$ and the relations into matrices $\mathbf{M}_r$. Its score function is $f_r(\mathbf{h}, \mathbf{t}) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t}$.

Single Layer Model (SLM) [Socher *et al.*, 2013]. As a base model of Neural Tensor Network, the model SLM designs its score function as $f_r(\mathbf{h}, \mathbf{t}) = \mathbf{u}_r^\top \tanh(\mathbf{M}_{r1}\mathbf{h} + \mathbf{M}_{r2}\mathbf{t} + \mathbf{b}_r$ with a nonlinear neural network, where $\mathbf{M}_{r1}, \mathbf{M}_{r2}$, and $\mathbf{b}_r$ are parameters related to relation r.

Neural Tensor Network (NTN) [Socher *et al.*, 2013]. To account for the second-order correlations, NTN extends the SLM by using nonlinear transformation neural networks. Its score function is $f_r(\mathbf{h}, \mathbf{t}) = \mathbf{u}_r^\top \tanh(\mathbf{h}^\top \underline{\mathbf{W}}_r \mathbf{t} + \mathbf{M}_{r1}\mathbf{h} + \mathbf{M}_{r2}\mathbf{t} + \mathbf{b}_r$, where $\mathbf{M}_{r1}, \mathbf{M}_{r2}$, and $\mathbf{b}_r$ are parameters related to relation r besides $\underline{\mathbf{W}}_r$ which is the 3-way tensor.

3 The Proposed Method

3.1 Motivation

As mentioned in the introduction, the head entity of the triplet can be interpreted as subject, the relation as predicate and the tail entity as object. Hence we can easily get that what the head entity, tail entity and relation are to each triplet as what subject, object and predicate are to a subject-predicate sentence. Intuitively, there is not only semantic information from the relation r to the entities h (or t), but also between h and t for each triplet (h, r, t). Previous methods mainly consider the former semantic information, which results in their poor capability on link prediction for complex relations. To overcome this shortcoming, our method transmits the semantic information from both entities h (or t) and relations r to entities t (or h). In addition, we believe that nonlinear functions are more efficient than linear functions to translate semantics, so we use nonlinear functions to replace the linear functions $p(\mathbf{r})$, formally as $p(\mathbf{e}, \mathbf{r})$ for the semantic transmission from both h and r to t and get final tail entity embedding vectors formally as $\mathbf{t}_\perp = \mathbb{P}(p(\mathbf{h}, \mathbf{r}), \mathbf{t})$. Given that the semantic transmission from t to h is in the opposite direction of r, we define the semantic transmission function from t and r

Model	Score function	#Parameters
SE [Bordes *et al.*, 2011]	$\Vert\mathbf{M_r^h h} - \mathbf{M_r^t t}\Vert_{\ell_{1/2}}$ $\mathbf{h}, \mathbf{t} \in \mathbb{R}^{k_e}, \mathbf{M_r^h}, \mathbf{M_r^t} \in \mathbb{R}^{k_e \times k_e}$	$2k_e n_e + 2k_e^2 n_r$
SME(linear) [Bordes *et al.*, 2012]	$-(\mathbf{M}_{h_1}\mathbf{e}_h + \mathbf{M}_{h_2}\mathbf{r} + \mathbf{b}_h)^\top(\mathbf{M}_{t_1}\mathbf{e}_t + \mathbf{M}_{r_2}\mathbf{r} + \mathbf{b}_t)$ $\mathbf{h}, \mathbf{t} \in \mathbb{R}^{k_e}, \mathbf{M_r^1}, \mathbf{M_r^2} \in \mathbb{R}^{k_e \times k_e}$	$2k_e n_e + 2k_e^2 n_r$
SME(bilinear) [Bordes *et al.*, 2012]	$-[(\mathbf{M}_{h_1}\mathbf{e}_h) \otimes (\mathbf{M}_{h_2}\mathbf{r}) + \mathbf{b}_h]^\top[(\mathbf{M}_{t_1}\mathbf{e}_t) \otimes (\mathbf{M}_{t_2}r) + \mathbf{b}_t]$ $\mathbf{h}, \mathbf{t} \in \mathbb{R}^{k_e}, \mathbf{M_r^1}, \mathbf{M_r^2} \in \mathbb{R}^{k_e \times k_e}$	$2k_e n_e + 2k_e^2 n_r$
TransE [Bordes *et al.*, 2013]	$\Vert\mathbf{h} + \mathbf{r} - \mathbf{t}\Vert_{\ell_{1/2}}$, $\mathbf{h}, \mathbf{t} \in \mathbb{R}^{k_e}, \mathbf{r} \in \mathbb{R}^{k_r}$	$k_e n_e + k_r n_r (*k_e = k_r)$
TransH [Wang *et al.*, 2014]	$\Vert\mathbf{h}(\mathbf{I} - \mathbf{w_r^\top w_t}) + \mathbf{r} - \mathbf{t}(\mathbf{I} - \mathbf{w_r^\top w_t})\Vert_{\ell_{1/2}}$, $\mathbf{h}, \mathbf{t} \in \mathbb{R}^{k}, \mathbf{w_r}, \mathbf{r} \in \mathbb{R}^{k}$	$k_e n_e + 2k_r n_r$
TransR [Lin *et al.*, 2015]	$\Vert\mathbf{hM_r} + \mathbf{r} - \mathbf{tM_r}\Vert_{\ell_{1/2}}$, $\mathbf{h}, \mathbf{t} \in \mathbb{R}^{k_e}, \mathbf{r} \in \mathbb{R}^{k_r}, \mathbf{M_r} \in \mathbb{R}^{k_e \times k_r}$	$k_e n_e + (k_r + k_r^2) n_r$
TransD [Ji *et al.*, 2015]	$\Vert(\mathbf{r}_p\mathbf{h}_p^\top + \mathbf{I}^{k_r \times k_e})\mathbf{h} + \mathbf{r} - (\mathbf{r}_p\mathbf{t}_p^\top + \mathbf{I}^{k_r \times k_e})\mathbf{t}\Vert_{\ell_{1/2}}$, $\mathbf{h}, \mathbf{t}, \mathbf{h_p}, \mathbf{t_p} \in \mathbb{R}^{k_e}, \mathbf{r}, \mathbf{r_p} \in \mathbb{R}^{k_r}$	$4k_e n_e + 2k_r n_r$
TranSparse(*separate) [Ji* et al., *2016]*	$\Vert\mathbf{M_r^1}(\theta_r^1)\mathbf{h} + \mathbf{r} - \mathbf{M_r^2}(\theta_r^2)\mathbf{t}\Vert_{\ell_{1/2}}$ $\mathbf{h}, \mathbf{t} \in \mathbb{R}^{k_e}, r \in \mathbb{R}^{k_r}, \mathbf{M_r^1}(\theta_r^1), \mathbf{M_r^2}(\theta_r^2) \in \mathbb{R}^{*k_r \times k_e}$	$2k_e n_e + 2(1-\theta)(k_e+1)k_r n_r$ $0 \ll \theta_r^1, \theta_r^2 \leq 1$
GTrans-DW [Tan *et al.*, 2018]	$\Vert 1/\sigma \odot [(\alpha\mathbf{h}_e + \beta\mathbf{r}_a\mathbf{h}_a^\top\mathbf{h}_e) + \mathbf{r}_e - (\alpha\mathbf{t}_e + \beta\mathbf{r}_a\mathbf{t}_a^\top\mathbf{t}_e)]\Vert_{\ell_{1/2}}$ $\mathbf{h}_a, \mathbf{h}_e, \mathbf{t}_a, \mathbf{t}_e \in \mathbb{R}^{k_e}, \mathbf{r}_a, \mathbf{r}_e \in \mathbb{R}^{k_r}, \sigma, \alpha, \beta \in \mathbb{R}$	$k_e n_e + 3k_r n_r (*k_e = k_r)$
GTrans-SW [Tan *et al.*, 2018]	$\Vert 1/\sigma \odot [(1-\alpha_{h,r})\mathbf{h}_e + \alpha_{h,r}\mathbf{r}_a\mathbf{h}_a^\top\mathbf{h}_e + \mathbf{r}_e - (1-\alpha_{t,r})\mathbf{t}_e$ $-\alpha_{t,r}\mathbf{r}_a\mathbf{t}_a^\top\mathbf{t}_e]\Vert_{\ell_{1/2}}$ $\mathbf{h}_a, \mathbf{h}_e, \mathbf{t}_a, \mathbf{t}_e \in \mathbb{R}^{k_e}, \mathbf{r}_a, \mathbf{r}_e \in \mathbb{R}^{k_r}, \sigma, \alpha, \beta \in \mathbb{R}$	$k_e n_e + 3k_r n_r (*k_e = k_r)$
TransMS	$\Vert - \tanh(\mathbf{t} \otimes \mathbf{r}) \otimes \mathbf{h} + \mathbf{r} - \tanh(\mathbf{h} \otimes \mathbf{r}) \otimes \mathbf{t} + \alpha \cdot g(\mathbf{h} \otimes \mathbf{t})\Vert_{\ell_{1/2}}$, $\mathbf{h}, \mathbf{t} \in \mathbb{R}^{k_e}, \mathbf{r} \in \mathbb{R}^{k_r}, \alpha = \mathbf{r}_{k_r+1} \in \mathbb{R}^1$	$k_e n_e + (k_r + 1) n_r (*k_e = k_r)$

Table 2: Statistics of datasets used in experiments. We mainly compare the models' score functions and their numbers of parameters. n_e and n_r represent the number of entities and relations in knowledge graph respectively. k_e and k_r represent the dimension of entity and relation in the low-dimensional space, $\mathbf{h}, \mathbf{t} \in \mathbb{R}^{k_e}$, $\mathbf{r} \in \mathbb{R}^{k_r}$. θ_r^1 and θ_r^1 denote the sparse degrees of transfer matrices in TranSparse. α represents one additional dimension for each relation vectors in our model TransMS. For TransE, GTrans and TransMS, ($*k_e = k_r$) means that k_e is equal to k_r.

to h as $p(-\mathbf{e}, \mathbf{r})$ and get final head entity embedding vectors formally as $\mathbf{h}_\perp = \mathbb{P}(p(-\mathbf{t}, \mathbf{r}), \mathbf{h})$.

In addition, it is common that every predicate r always owns its executors and enforcers, and subjects h are the executors of predicates while objects t are enforcers of predicates, so it is significant to transmit the semantic information from subjects h and objects t to predicates r. Although the semantic information of r is transmitted to h and t for each triplet respectively by the nonlinear function $p(\mathbf{e},\ \mathbf{r})$, the semantic information of h and t is not transmitted to r. To carry out this transmission, we structure bias vectors which is defined as $\alpha \cdot g(\mathbf{h}, \mathbf{t})$ to $\mathbf{r}$, where α is the one added dimension for relation and $g(\mathbf{h}, \mathbf{t})$ is the function related to both $\mathbf{h}$ and $\mathbf{t}$ and get final relation embedding vectors formally as $\mathbf{r}_\perp = \mathbb{G}(\mathbf{r}, \alpha \cdot g(\mathbf{h}, \mathbf{t}))$. Figure 2 illustrates our approach.

In line with TransE, our model follows the similar translation-based constraint for triplet embedding vectors $(\mathbf{h}_\perp, \mathbf{t}_\perp, \mathbf{r}_\perp)$: $\mathbf{h}_\perp + \mathbf{r}_\perp \approx \mathbf{t}_\perp$, and the score function should be $f_r(\mathbf{h}, \mathbf{t}) = \Vert\mathbf{r}_\perp + \mathbf{h}_\perp - \mathbf{t}_\perp\Vert_{\ell_{1/2}}$.

3.2 The Proposed TransMS

Inspired by the above motivation, we propose a novel model named TransMS (translates and transmits multidirectional semantics) to deal with link prediction for complex relations (i.e. 1-N, N-1 and N-N relations). We first project every head entity h as an entity embedding vector $\mathbf{h} \in \mathbb{R}^{k_e}$, tail entity t as an entity embedding vector $\mathbf{t} \in \mathbb{R}^{k_e}$, and relation r as a relation vector $\mathbf{r} \in \mathbb{R}^{k_r}$ and a variable $\alpha \in \mathbb{R}^1$, while we can think of α as an added one dimension for relation embedding vectors where k_e and $k_r + 1$ are the dimension of entity and relation embedding vectors respectively, and $k_e = k_r \triangleq k$ in our model. Then, we transmit the semantic information from tail (or head) entities t (or h) and relations r to head (or tail) entities h (or t) by nonlinear functions $p(\mathbf{e}, \mathbf{r})$ (or $p(-\mathbf{e}, \mathbf{r})$). In particular, $tanh$ is empirically found the optimal one in our experiments on function $p(\mathbf{e}, \mathbf{r})$ among $\{tanh,\ softmax,\ ReLu,\ atan,\ tan,\ sin,\ cos\}$. Hence, entity vectors with semantic information are specified as:

$$\begin{aligned} \mathbf{h}_\perp &= \tanh(-\mathbf{t} \otimes \mathbf{r}) \otimes \mathbf{h} \\ &= -\tanh(\mathbf{t} \otimes \mathbf{r}) \otimes \mathbf{h} \qquad (1) \\ \mathbf{t}_\perp &= \tanh(\mathbf{h} \otimes \mathbf{r}) \otimes \mathbf{t} \qquad (2) \end{aligned}$$

where $\otimes$ means the Hadamard product.

And then, we transmit the semantic information from head entities h and tail entities t to relations r by adding a bias vectors constructed with multiplying the Hadamard product between head vectors $\mathbf{h} \in \mathbb{R}^k$ and tail vectors $\mathbf{t} \in \mathbb{R}^k$ with α:

$$\mathbf{r}_\perp = \mathbf{r} + \alpha \cdot (\mathbf{h} \otimes \mathbf{t}) \qquad (3)$$

where $\cdot$ means the scalar multiplication.

Finally, the score function of our model is

$$\begin{aligned} f_r(\mathbf{h}, \mathbf{t}) &= \Vert\mathbf{h}_\perp + \mathbf{r}_\perp - \mathbf{t}_\perp\Vert_{\ell_{1/2}} \\ &= \Vert - \tanh(\mathbf{t} \otimes \mathbf{r}) \otimes \mathbf{h} + \mathbf{r} + \alpha \cdot (\mathbf{h} \otimes \mathbf{t}) + \\ &\quad - \tanh(\mathbf{h} \otimes \mathbf{r}) \otimes \mathbf{t}\Vert_{\ell_{1/2}} \qquad (4) \end{aligned}$$

The score is expected to be lower for golden triplets (the triplets from the knowledge graph) than for negative triplets (the triplets outside the knowledge graph).

Data sets	**WN18**				**FB15K**			
Metric	**MeanRank**		**Hit@10**		**MeanRank**		**Hit@10**	
	Raw	**Filter**	**Raw**	**Filter**	**Raw**	**Filter**	**Raw**	**Filter**
SE [Bordes *et al.*, 2011]	1011	985	68.5	80.5	273	162	28.8	39.8
SME (linear/bilinear) [Bordes *et al.*, 2012]	545/526	533/509	65.1/54.7	74.1/61.3	274	154/158	30.7/31.3	40.8/41.3
TransE [Bordes *et al.*, 2013]	263	251	75.4	89.2	243	125	34.9	47.1
TransH (unif/bern) [Wang *et al.*, 2014]	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) [Lin *et al.*, 2015]	232/401	219/388	78.3/79.8	91.7/92.0	226/198	78/77	43.8/48.2	65.5/68.7
CTransR (unif/bern) [Lin *et al.*, 2015]	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) [Ji *et al.*, 2015]	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 (separate, US, unif/bern) [Ji *et al.*, 2016]	233/223	221/211	79.6/80.1	93.4/93.2	216/190	66/82	50.3/53.7	78.4/79.9
GTrans_DW (unif/bern) [Tan *et al.*, 2018]	**210**/180	197/166	78.4/77.1	92.2/90.3	256/235	142/126	44.1/43.1	63.4/60.5
GTrans_SW (unif/bern) [Tan *et al.*, 2018]	247/215	234/**202**	79.1/80.2	92.9/93.5	207/189	66/85	50.6/52.9	75.1/75.3
TransMS (unif/bern)	427/455	414/442	**82.5/82.5**	**94.7/94.8**	**171**/213	**63**/104	**55.0/54.3**	**86.8**/78.4

Table 3: Results of link prediction on WN18 and FB15K, under the settings of both ‘unif’ and ‘bern’.

Tasks	Head Prediction (**Hit@10**)				Tail Prediction (**Hit@10**)			
Relation Type	**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 [Bordes *et al.*, 2011]	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME (linear/bilinear) [Bordes *et al.*, 2012]	35.1/30.9	53.7/69.6	19.0/19.9	40.3/38.6	32.7/28.2	14.9/13.1	61.6/76.0	43.3/41.8
TransE [Bordes *et al.*, 2013]	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH (unif/bern) [Wang *et al.*, 2014]	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) [Lin *et al.*, 2015]	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) [Lin *et al.*, 2015]	78.6/81.5	77.8/89.0	36.4/34.7	68/71.2	77.4/80.8	37.8/38.6	78.0/90.1	70.3/73.8
TransD (unif/bern) [Ji *et al.*, 2015]	80.7/86.1	85.8/95.5	47.1/39.9	75.6/78.5	80.0/85.4	54.5/50.4	80.7/94.4	77.9/81.2
TranSparse (separate,US,unif/bern) [Ji *et al.*, 2016]	83.2/87.1	85.2/ 95.8	51.4/44.4	80.3/81.2	82.6/87.5	60.0/57.0	85.5/**94.5**	82.5/83.7
GTrans_DW (unif/bern) [Tan *et al.*, 2018]	77.4/76.3	87.1/86.0	23.4/20.3	65.6/61.2	74.3/75.6	32.3/31.7	84.4/83.5	67.2/63.9
GTrans_SW (unif/bern) [Tan *et al.*, 2018]	80.1/84.9	93.0/95.0	48.4/39.9	75.4/75.9	79.4/84.4	51.8/47.7	91.2/**94.5**	77.8/78.8
TransMS (unif/bern)	**89.5/91.4**	94.4/**95.9**	**78.5**/44.9	**85.6**/78.5	**90.0/91.6**	**84.8**/54.1	91.7/93.6	**87.7**/82.0

Table 4: Detailed results by the category of relation on FB15K, under the settings of both ‘unif’ and ‘bern’.

3.3 Learning of TransMS

Before training, it is important to obtain negative triplets and devising the loss function. In our model, we set every triplet (h_i, r_i, t_i) has a label y_i which equals to 1 when the triplet is golden or -1 when the triplet is negative. And then, we denote the golden triplet set as $\mathbb{D} = \{(h_i, r_i, t_i)|y_i = 1\}$ and negative triplet set as $\mathbb{D}' = \{(h'_i, r_i, t_i)|y_i = -1\} \bigcup \{(h_i, r_i, t'_i)|y_i = -1\}$ where h'_i or t'_i is corrupted (but not both corrupted at the same time) by randomly sampling in two means: i) uniform sampling from all entities with the equal probability denoted as ‘unif’ as in [Bordes *et al.*, 2013], and ii) Bernoulli sampling from all entities with different probabilities which depend on the projecting property of the relations denoted as ‘bern’ as in [Wang *et al.*, 2014]. The embedding vector for each given entity is a uniquely projected vector whether it is head or tail entity.

We use the following margin-based ranking loss function for our training as in [Bordes *et al.*, 2013].

$$\mathcal{L} = \sum_{(h,r,t)\in\mathbb{D}} \sum_{(h',r,t')\in\mathbb{D}'} [f_r(\mathbf{h},\mathbf{t}) + \gamma - f_r(\mathbf{h}',\mathbf{t}')]_+ \qquad (5)$$

where $[x]_+ :\triangleq \max(0, x)$ aims to get the maximum between x and 0, and γ which is the margin separating golden and negative triplets could keep far from each other between golden and negative candidates. As in previous models, we enforce constraints on the normalization of vectors $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$. i.e. $\forall(h, r, t) \in \mathbb{D} \bigcup \mathbb{D}'$, $\|\mathbf{h}\|_2 \leq 1$, $\|\mathbf{r}_{1:k}\|_2 \leq 1$, $\|\mathbf{t}\|_2 \leq 1$.

We adopt Adam [Kingma and Ba, 2015] to minimize the above loss with additional sparse norm regularizer on entities' embeddings. All embedding vectors for entities and relations are first initialized following the random procedure in [Glorot and Bengio, 2010]. Then, the entity embedding vectors $\mathbf{e} \in \mathbb{R}^k$ and the relation embedding vectors $\mathbf{r} \in \mathbb{R}^k$ are normalized at each main iteration except the additional dimension α. The set of golden triplets are randomly traversed multiple times followed by the construction of negative triplets.

Table 2 lists the score functions and the total number of parameters for some typical translation-based models and our model TransMS. It can be obviously seen that for each triplet, the number of parameters in our model is only one more than the simplest TransE while much fewer than the others, which indicates the cost-effectiveness of our approach.

4 Experiments and Analysis

4.1 Datasets and Protocols

We mainly evaluate our model on two typical knowledge graphs, which are built with WordNet [Miller, 1995] and Freebase [Bollacker *et al.*, 2008] databases used in previous models. WordNet regarded as sysnets which groups English words into sets of synonyms is a lexical database for English. In this paper, we use one subset of WordNet: WN18 [Bordes *et al.*, 2014], which has 40943 entities and 18 relations. Freebase is composed of general human knowledge. For instance, the triplet $(Obama, was\_a\_president\_of, America)$, which means Obama was a president of America, is a fact in the real world. In this paper, we employ one subset of Freebase: FB15K [Bordes *et al.*, 2014]. FB15K has 14951 entities and 1345 relations. WH18 and FB15K are two typical subsets of Knowledge graph as they have simple or complex relations respectively. In addition, we perform comparative experiments on the WN18RR [Dettmers *et al.*, 2017] and FB15K-237 [Toutanova and Chen, 2015] datasets in contrast with ConvE [Dettmers *et al.*, 2017] which is a deep learning model for knowledge graph embedding. FB15K-237 which has 237 relations and 14541 entities is a subset of FB15K. WN18RR which has 11 relations and 40943 entities is a subset of WN18. The four databases consist of training, validation and testing sets which have been well constructed as Table 1.

We use link prediction for evaluation in line with [Bordes *et al.*, 2013]. The prediction for new relations between head and tail entities when the head and tail entities are given is equivalent to the prediction for tail/head entities when the head/tail entities and relations are given.In knowledge graph, the purpose of link prediction is to predict head/tail entities when tail/head entities and relations are given. For example, for the triplet $(Obama, was\_a\_president\_of, America)$ in Freebase, link prediction aims to predict missing head entity of $(*, was\_a\_president\_of, America)$ or missing tail entity of $(Obama, was\_a\_president\_of, *)$. Link prediction emphasizes the rank of the origin golden entity in knowledge graph but not focus on the best candidate entity as there may be more than one suitable candidate. In particular, there are two popular metrics: **i) MeanRank.** The mean rank of correct entities; **ii) Hit@10.** The ratio of correct entities ranked in top 10.

The MeanRank evaluation contains two parts. One is to evaluate the head entity prediction, the other is to evaluate the tail entity prediction. For the head entity prediction, the head entity for each triplet in testing set is first removed and replaced by the entities of entity set $\mathbb{R}$ to construct the corrupted triplets. Next, the scores of these corrupted triplets whose relations and tail entities are the same as each other are computed by the score function and sorted by ascending order. Then, the ranks of the correct entities are stored. Finally, the mean rank of head entity predictions is computed and denoted as MeanRank evaluation for the head entity prediction. The whole procedure is repeated while the tail entities are replaced instead of the head entities for the tail entity prediction. The final MeanRank is the average of the MeanRank for the head and tail entity prediction among all triplets in testing set. Generally, the better model is followed by the lower MeanRank.

As MeanRank, the Hit@10 evaluation also has two parts: evaluation for the head entity prediction and evaluation for the tail entity prediction. For the head entity prediction, we count the number of the correct head entities ranked in top 10, and then the number is divided by the total number of triplets in testing set. The ratio is denoted as Hit@10 evaluation for the head entity prediction. Similarly, the Hit@10 evaluation for the tail entity prediction is calculated with the same method, and the final Hit@10 is the average of the Hit@10 for the head and tail entity prediction among all triplets in testing set. Generally, the higher Hit@10 means the better model, the higher Hit@10 is followed by the lower MeanRank. It is a flaw that the MeanRank evaluation will be abnormal as there are abnormal ranks of correct triplets such that the Hit@10 is the better metric than the MeanRank.

DataSet	**FB15K-237**	
Metric	**MeanRank**	**Hit@10**
ConvE [Dettmers *et al.*, 2017]	330	**0.458**
TransMS	**249**	0.445

DataSet	**WN18RR**	
Metric	**MeanRank**	**Hit@10**
ConvE [Dettmers *et al.*, 2017]	7323	0.411
TransMS	**6523**	**0.460**

Table 5: Filter results of link prediction on WN18 and FB15K, under the setting of ‘unif’.

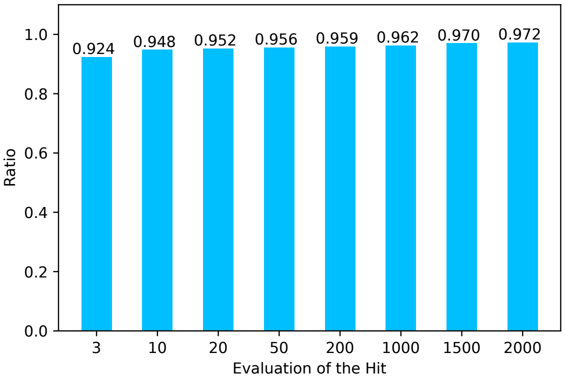

Figure 3: The results of different Hit metric on WN18 under the setting of ‘unif’ is similar to the results on WN18 under the setting of ‘bern’.

In addition, given that some corrupted triplets ranked above the right triplets in testing set may be golden ones in training, validation or testing sets, the MeanRank and Hit@10 should be divided into two types: One is the *Filter* representing statistics that the correct triplets belonged to training, validation or testing sets are first filtered out from the corrupted triplets and then the scores of another corrupted triplets are ranked, the other is the *Raw* representing the original statistics that scores of all corrupted triplets are ranked.

We conduct experiments on four knowledge graph subsets: WN18, FB15k, WN18RR and FB15K-237, and compare with methods listed in Table 2 and ConvE. We directly report the results of those methods from the literature as in [Wang *et al.*, 2014; Ji *et al.*, 2016; Zhang, 2017; Dettmers *et al.*, 2017; Tan *et al.*, 2018]. We conduct 200 iterations of training on FB15k-237 and 500 iterations of training on the other three databases. The learning rate β for Adam is set among $\{0.1, 0.01, 0.001\}$, the margin γ among $\{1.0, 1.5, 2.0, 4.0\}$, the dimension of vectors d among $\{50, 100, 150, 200\}$, the mini-batch size b among$\{200, 1200, 4800\}$ and additional regularizers among $\{\ell_1, \ell_2\}$. For both WN18 and FB15K, the optimal configurations are: $\beta = 0.001$, $\gamma = 2.0$, $d = 200$, $b = 4800$ and $L = \ell_1$ on both corrupted means: ‘unif’ and ‘bern’, which are same as the optimal configurations of WN18RR and FB15K-237 on ‘unif’.

Data sets	WN18			
Metric	MeanRank		Hit@10	
	Raw	Filter	Raw	Filter
α is same(unif/bern)	481/461	469/451	82.3/82.1	**94.8**/94.7
α is different(unif/bern)	**427/455**	**414/442**	**82.5/82.5**	94.7/**94.8**

Data sets	FB15K			
Metric	MeanRank		Hit@10	
	Raw	Filter	Raw	Filter
α is same(unif/bern)	227/230	111/128	51.8/53.7	79.8/75.1
α is different(unif/bern)	**171/213**	**63/104**	**55.0/54.3**	**86.8/78.4**

Table 6: Results of link prediction on WN18 and FB15K, under the settings of both 'unif' and 'bern'. "α *is same*" means the parameter α is shared by different relations."α *is different*" means the parameter α varies with relations.

4.2 Results and Analysis

Experimental results on both FB15K and WN18 are shown in Table 3. On FB15K, by the means of both 'unif' and 'bern', our model improves Hit@10 and MeanRank with both *Raw* method and *Filter* methods, especially our model performs better than the SE, SME and Trans(E, H, R) which are usually used as comparison because they have better scalability than the others, while on WN18, our model improves Hit@10 with *Raw* and *Filter* methods.

Regarding Hit@10 for diverse relations, Table 4 lists the results by projecting properties of relations on FB15K. We can find that Hit@10s of the head prediction are improved for 1-1, N-1 and N-N relations by the means of 'unif' while improved for 1-1 and N-1 relations by the means of 'bern', and the Hit@10s of the tail prediction are improved for 1-1, 1-N and N-N relations by the means of 'unif' while improved for 1-1 relations by the means of 'bern'. There are significant improvements for the Hit@10 of the head prediction for N-1 relations and the tail prediction for 1-N relations by about 27.1% and 24.8% respectively while both predictions always are the tough nut to crack for long time.

In Table 3, TransMS outperforms the peer methods except the MeanRank metric on WN18. We get the more Hit results showed in Figure 3, which reveales that the results of entity prediction are mainly distributed in Hit@20 and there are less than 5% outside the Hit@20. Therefore, it is reasonable for the worse MeanRank on WN18 because the MeanRank will fluctuate abnormally as abnormal ranks of corrupted triplets.

Experimental results on both FB15K-237 and WN18RR are shown in Table 5. It can be seen that our model performs better than ConvE on WN18RR as MeanRank on FB15K-237 while the Hit@10 is only slightly lower than ConvE on FB15K-237. Given that ConvE is modelled by fully-connected and convolutional layers, our model not only has better scalability than ConvE but also has similar performance to ConvE.

In addition, we also evaluate our model on FB15K and WN18 datasets when the parameter γ changes according to relations or is a constant for different relations. In these experiments, the hyperparameters remain equal as in the part *Datasets and Protocols*. As we can see from the results listed in Table 6, the performance will be better when the parameter γ varies with relations than when α keeps a same value for different relations.

Dataset	FB15K
Relation	/tv/tv_series_episode/guest_stars./tv/tv_guest_role/actor
Similar relations	/tv/tv_actor/starring_roles./tv/regular_tv_appearance/series
	/tv/tv_program/recurring_writers./tv/tv_program_writer_relationship/writer
	/tv/tv_actor/guest_roles./tv/tv_guest_role/episodes_appeared_in
	/film/film/produced_by
	/film/director/film
Relation	/baseball/baseball_team/team_stats./baseball/baseball_team_stats/season
Similar relations	/baseball/baseball_team/historical_roster.sports/sports_team_roster/player
	/baseball/baseball_player/former_teams./sports/sports_team_roster/team
	/sports/pro_athlete/teams./baseball/baseball_historical_roster_position/team
	/sports/sports_team/roster./baseball/baseball_roster_position/position
	/baseball/baseball_league/teams

Table 7: Top-5 similar relations for each relation by relation embedding vectors on FB15K [Cosine similarity].

There are two relations with top-5 similar relations on cosine similarity metric by relation embedding vectors respectively in Table 7. We can find that such candidate symbolic relations have similar semantics to the base relations. These results indicate that the semantic translations among head entities, tail entities and relations are significant for the knowledge graph representation.

5 Conclusion

We have presented a novel model named TransMS that takes the multidirectional semantic information into account. Our model transmits i) the semantics from the head/tail entity and relation to the tail/head entity by a nonlinear function and ii) the semantics from entities to relation by adding a bias vector to relation embedding vector for each triplet. TransMS has better scalability than the peer models because it only has one more parameter than the simplest translation-based embedding model (i.e. TransE) but fewer than the others. In future, we will explore the best methods to translate semantics within triplets to improve the performance of link prediction for complex relations in large-scale knowledge graphs.

Acknowledgements

This work is supported by National Key Research and Development Program of China (2018YFC0830400), (2016YFB1001003) and STCSM(18DZ1112300).

References

[Bollacker *et al.*, 2008] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase:

a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250, 2008.

[Bordes *et al.*, 2011] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011*, 2011.

[Bordes *et al.*, 2012] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012*, pages 127–135, 2012.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *27th Annual Conference on Neural Information Processing Systems 2013*, pages 2787–2795, 2013.

[Bordes *et al.*, 2014] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. *Machine Learning*, 94(2):233–259, 2014.

[Dettmers *et al.*, 2017] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. *CoRR*, abs/1707.01476, 2017.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010*, pages 249–256, 2010.

[Jenatton *et al.*, 2012] Rodolphe Jenatton, Nicolas Le Roux, Antoine Bordes, and Guillaume Obozinski. A latent factor model for highly multi-relational data. In *26th Annual Conference on Neural Information Processing Systems 2012.*, pages 3176–3184, 2012.

[Ji *et al.*, 2015] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics ACL 2015*, pages 687–696, 2015.

[Ji *et al.*, 2016] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 985–991, 2016.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2015.

[Lin *et al.*, 2015] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2181–2187, 2015.

[Miller, 1995] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[Socher *et al.*, 2013] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In *International Conference on Neural Information Processing Systems*, pages 926–934, 2013.

[Tan *et al.*, 2018] Zhen Tan, Xiang Zhao, Yang Fang, and Weidong Xiao. GTrans: generic knowledge graph embedding via multi-state entities and dynamic relation spaces. *IEEE Access*, 6(99):8232–8244, 2018.

[Toutanova and Chen, 2015] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. 2015.

[Wang *et al.*, 2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119, 2014.

[Zhang, 2017] Wen Zhang. Knowledge graph embedding with diversity of structures. In *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, pages 747–753, 2017.