Journal of Computational Science 30 (2019) 108–117

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

A neural translating general hyperplane for knowledge graph embedding[☆]

Qiannan Zhu[a,b], Xiaofei Zhou[a,b,*], Peng Zhang[c], Yong Shi[d,e,f]

[a] Institute of Information Engineering, Chinese Academy of Sciences, China
[b] University of Chinese Academy of Sciences, School of Cyber Security, China
[c] Ant Financial, China
[d] CAS Research Center on Fictitious Economy and Data Science, China
[e] Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Science, China
[f] College of Information Science and Technology, University of Nebraska at Omaha, USA

ARTICLE INFO

Article history:
Received 15 September 2018
Received in revised form 23 October 2018
Accepted 7 November 2018
Available online 15 November 2018

Keywords:
Knowledge representation
Knowledge embedding
Knowledge graph completion
Link prediction
Neural network

ABSTRACT

For completing knowledge graph, many translation-based models, like that Trans(E and H) which embed a knowledge graph into a continuous vector space and encode relations as translation operations in that space, have achieved better performance. However, most of them have limitations in expressing complex relations for knowledge graph. In this paper, we propose a translation-neural based method NTransGH for knowledge graph completion. NTransGH combines translation mechanism for modeling relations as translation operations by generalized hyperplanes, and a neural network for capturing more complex interactions between entities and relations. We conduct experiment on two tasks link prediction and triplet classification with two datasets. Experimental results show that NTransGH has strong expression in mapping properties of complex relations, and achieves significant and consistent improvements over state-of-the-art embedding methods. This paper is an extension of our previous works [1].

1. Introduction

Knowledge graphs (KGs) like Freebase [2], DBpedia [3] and NELL [4], are useful resources for numerous artificial intelligence applications, such as Question Answering System [5–8], Information Extraction [9–12] and Knowledge Management [13–16]. A knowledge graph can be formalized as a directed multi-relation graph, which is composed of facts in the form of triplets (h, r, t). The triplet represents that there is a relation r between head entity h and tail entity t, e.g., (Barack_Obama, Place_of_Birth, Hawai). Although a large amount of structured data is stored, knowledge graphs often seriously suffer from incompleteness. The aim of knowledge graph completion is to complete the missing facts of the knowledge graph based on the existing facts.

Recently an emerging research direction named *knowledge graph embedding* has attracted great attention due to its strong feasibility and robustness. It attempts to embed entities and relations of a KG into a continuous vector space while preserving KG's inherent properties. Following this approach, many methods have been presented in section "Related Work". Among these methods, translation-based models are fundamental and develop quickly. TransE [17] simply encodes relations as translation operation between entities, which has difficulties in dealing with complex relations.[1] To address this issue, TransH [18] introduces relation-specific hyperplanes with normal vector $\mathbf{w}_r$ to make entities have different representations for different relations.

But both TransE and TransH lead to limited representation for complex relations since they embed entities and relations into the same space. TransR [19] considers entities and relations as different objects, and transfers entities from entity space to relation space with a transfer matrix. Based on TransR, TransD [20] further considers the multiple types of entities and relations simultaneously, and uses the product of two projection vectors of an entity-relation pair to construct the transfer matrix. Although the transfer matrix has more general representation for mapping properties, it still costs much more computations and memories on the mappings.

☆ Fully documented templates are available in the elsarticle package on CTAN.

* Corresponding author at: Institute of Information Engineering, Chinese Academy of Sciences, China.
E-mail address: zhouxiaofei@iie.ac.cn (X. Zhou).

[1] complex relations can be reflexive, one-to-many, many-to-one, and many-to-many relations.

https://doi.org/10.1016/j.jocs.2018.11.004
1877-7503/

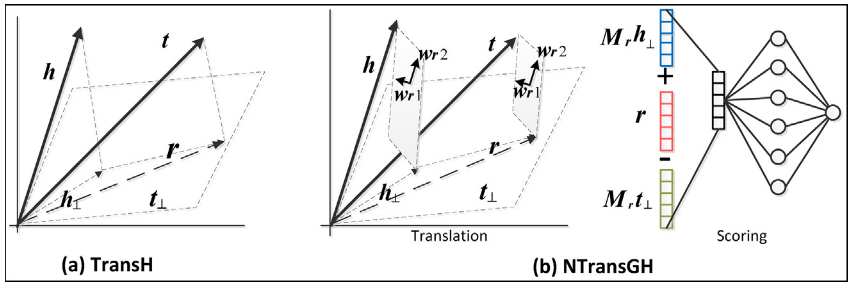


Fig. 1. Simple visualization of TransH and NTransGH.

In this paper, we propose a translation-neural based model NTransGH which improves the capability of expressing complex relations. NTransGH first conducts translation mechanism for modeling relations as translation operations by generalized hyperplanes, and then designs a neural network for capturing complex relational patterns. Specifically, for modeling translation operation, based on TransH as shown in Fig. 1(a), NTransGH uses a set of basis vectors instead of one normal vector in TransH to determine a generalized hyperplane. For expressing complex relations, NTransGH introduces a two-layer neural network to define a score function in order to find low dimensional embeddings for complex relations. The basic idea of NTransGH illustrated in Fig. 1(b) is that for a given triplet (h, r, t), in translation phrase, NTransGH firstly projects the entity embeddings $\mathbf{h}$ and $\mathbf{t}$ on the generalized hyperplane as $\mathbf{h}_\perp$ and $\mathbf{t}_\perp$ with a set of basis vectors respectively. Then it conducts translation operation with a transfer matrix $\mathbf{M}_r$ for getting the triplet representation as $\mathbf{m} = \mathbf{M}_r\mathbf{h}_\perp + \mathbf{r} - \mathbf{M}_r\mathbf{t}_\perp$. Secondly, NTransGH defines a score function $f(h, r, t)$=N($\mathbf{m}$) for measuring the triplet's compatibility.[2] The score function is expected to be large for positive triplets and small for negative triplets.

Our contributions in this paper are: (1) We propose a novel translation-neural based model NTransGH for knowledge graph embedding, which incorporates translation mechanism with a neural network for expressing complex relations. It is the first work for knowledge graph completion. (2) NTransGH introduces the generalized hyperplanes determined by a set of basis vectors for modeling each relation as a translation vector. (3) NTransGH uses a two-layer neural network to define a scoring function for capturing complex entity-relation interactions. (4) NTransGH has significant improvements comparing with previous baselines in two tasks of link prediction and triplet classification with two datasets.

In the remainder of this paper, we first briefly give the summary of embedding-based methods in Section 2. After that, we propose a translation-neural based model NTransGH for improving the performance on learning knowledge graph embeddings in Section 3. Finally we report and analyze the results of experiments on link prediction and triplet classification in Section 4, and make discussions on how parameters affect the performance of our model in Section 5, followed by the concluding remarks in Section 6.

2. Related work

Embedding-based models have three steps for knowledge graph embedding. First, they encode the entities and relations into a low-dimensional vector space. Second, they define a score function to measure the plausibility of each triplet. Third, they maximize the total plausibility of triplets to learn the embeddings of entities and relations. For the definition of score function, embedding-based models can be roughly classified into two groups: (1) *Translational distance-based models*, (2) *semantic matching-based models*. The former defines score function as distance between the latent representations of entities and relations, the latter utilizes similarity ones. Below we briefly summarize some baseline models.

2.1. Translational distance-based models

TransE [17] simply regards each relation as a translation vector between the head entity embedding and tail entity embedding. Its score function is defined as $f(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2$ for a triplet (h, r, t), where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in R^m$. TransH [18] introduces a hyperplane determined by a normal vector $\mathbf{w}_r$ and uses it to models each relation r as a vector $\mathbf{r}$. Its score function is $f(h, r, t) = \|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2^2$, where $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^T\mathbf{h}\mathbf{w}_r$, $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r{}^T\mathbf{t}\mathbf{w}_r$ and $\mathbf{h}, \mathbf{r}, \mathbf{t}, \mathbf{w}_r \in R^m$. TransR [19] respectively embeds entities and relations into entity space and relation spaces, and maps entity embeddings from entity space to r-relation space with a mapping matrix $\mathbf{M}_r$. Its score function is $f(h, r, t) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_2^2$, where $\mathbf{h}_r = \mathbf{h}\mathbf{M}_r$, $\mathbf{t}_r = \mathbf{t}\mathbf{M}_r$, $\mathbf{h}, \mathbf{r}, \mathbf{t} \in R^m$, $\mathbf{M}_r \in R^{m\times m}$. TransD [20] replaces transfer matrix in TransR by the product of two projection vectors of an entity-relation pair. Its score function is denoted as $f(h, r, t) = \|\mathbf{M}_{rh}\mathbf{h} + \mathbf{r} - \mathbf{M}_{rt}\mathbf{t}\|_2^2$, where $\mathbf{M}_{rh} = \mathbf{r}_p\mathbf{h}_p{}^T + \mathbf{I}^{n\times m}$, $\mathbf{M}_{rt} = \mathbf{r}_p\mathbf{t}_p{}^T + \mathbf{I}^{n\times m}$, and $\mathbf{h}, \mathbf{h}_p, \mathbf{t}, \mathbf{t}_p \in R^m$, $\mathbf{r}, \mathbf{r}_p \in R^n$. STransE [21] combines insights from SE and TransE models, which uses relation-specific matrices $\mathbf{M}_{rh}$ and $\mathbf{M}_{rt}$ as in the SE model to identify the relation-dependent aspects of both $\mathbf{h}$ and $\mathbf{t}$, and uses a vector $\mathbf{r}$ as in the TransE model to describe the relationship between h and t in subspace. Then the score function is $f(h, r, t) = \|\mathbf{M}_{rh}\mathbf{h} + \mathbf{r} - \mathbf{M}_{rt}\mathbf{t}\|_2^2$, where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in R^m$, $\mathbf{M}_{rh}, \mathbf{M}_{rt} \in R^{m\times m}$. Besides, TranSparse [22] adopts adaptive sparse transfer matrices to project entities from entity space to relation-specific space. TransM [23] pre-calculates the distinct weight for each triplet by leveraging the structure of knowledge graph. KG2E [24] uses Gaussian distributions for modeling entity and relation representations. TransG [25] is the Bayesian non-parametric infinite mixture embedding model to addresses the issue of multiple relation semantics. ITransF [26] designs a learning algorithm to induce the interpretable sparse representation of entities and relations.

Recently, a limit-based scoring loss [27] is proposed and used in Trans(E,H) denoted as Trans(E,H)-RS for learning knowledge embeddings, which have significant improvements compared to original baselines.

2.2. Semantic matching-based models

Unstructured Mole (UM) [28] sets all relation vectors as $\mathbf{r} = 0$, which leads to the score function $f(h, r, t) = \|\mathbf{h} - \mathbf{t}\|$. Structured Embedding (SE) [29] defines two independent matrices $\mathbf{M}_{rh}$ and $\mathbf{M}_{rt}$ for each relation to project the head entity embedding and tail entity embedding. Its score function is $f(h, r, t) = \|\mathbf{M}_{rh}\mathbf{h} - \mathbf{M}_{rt}\mathbf{t}\|$. Latent Factor Model (LFM) [30,31] models each relation as a matrix that is asymmetric and directly operates between two entity

[2] $N(\mathbf{m})$ means that the neural network takes the representation $\mathbf{m}$ as input and outputs a score.

embeddings. The score function is $f(h, r, t) = \mathbf{h}^T\mathbf{M}_r\mathbf{t}$, where $\mathbf{h}, \mathbf{t} \in R^m$, $\mathbf{M}_r \in R^{m\times m}$. Semantic Matching Energy (SME) [32,33] introduces two definitions of semantic matching energy functions for optimization, a linear form $f(h, r, t) = (\mathbf{M}_1\mathbf{h} + \mathbf{M}_2\mathbf{r} + \mathbf{b}_1)^T(\mathbf{M}_3\mathbf{t} + \mathbf{M}_4\mathbf{r} + \mathbf{b}_2)$, and a bilinear form $f(h, r, t) = (\mathbf{M}_1\mathbf{h} \otimes \mathbf{M}_2\mathbf{r} + \mathbf{b}_1)^T(\mathbf{M}_3\mathbf{t} \otimes \mathbf{M}_4\mathbf{r} + \mathbf{b}_2)$, where $\otimes$ is Hadamard product, and $M_1, M_2, M_3, M_4 \in R^{m\times m}$, $\mathbf{h}, \mathbf{r}, \mathbf{t} \in R^m$. Single Layer Model (SLM) [34] is designed as a plain baseline of NTN. It introduces nonlinear transformations by neural networks. The score function is $f(h, r, t) = \mathbf{u}_r{}^T g(\mathbf{M}_{rh}\mathbf{h} + \mathbf{M}_{rt}\mathbf{t} + \mathbf{b}_r)$, where $g(\cdot)$ is the function $\tanh(\cdot)$, $\mathbf{M}_{rh}, \mathbf{M}_{rt} \in R^{m\times m}$, $\mathbf{u}_r, \mathbf{h}, \mathbf{t}, \mathbf{b}_r \in R^m$. The Neural Tensor Network(NTN) [34] uses a bilinear tensor layer related two entity vectors to replace a standard linear neural network layer. It computes a score to measure the plausibility of a triplet (h, r, t) by the function $f(h, r, t) = \mathbf{u}_r{}^T g(\mathbf{h}^T\mathbf{M}_r\mathbf{t} + \mathbf{V}_r[\mathbf{h}\,;\mathbf{t}] + \mathbf{b}_r)$ where $g(\cdot) = \tanh(\cdot)$; $[\mathbf{h}\,;\mathbf{t}]$ denotes the vertical stacking of vectors $\mathbf{h}$ and $\mathbf{t}$, $\mathbf{M}_r$ is a 3-way tensor. RESCAL [35] is a tensor factorization model that encodes each relation r as a matrix $\mathbf{M}_r$ to model pairwise interactions between latent semantics of entities. Its score function is defined as $f(h, r, t) = \mathbf{h}^T\mathbf{M}_r\mathbf{t} = \sum_{i=0}^{m-1}\sum_{j=0}^{m-1}[\mathbf{M}_r]_{ij}[\mathbf{h}]_i[\mathbf{t}]_j.[\mathbf{h}]_i$ and $[\mathbf{t}]_j$ are the ith and jth entry of $\mathbf{h}$ and $\mathbf{t}$, and $[\mathbf{M}_r]_{ij}$ is the ijth entry of matrix $\mathbf{M}_r$. The score function can capture pairwise interactions between all the components of $\mathbf{h}$ and $\mathbf{t}$. Additional tensor factorization based works have been done as [36–44].

All above models complete knowledge graphs using one-step relational path, other models such as [45–47] based on multi-step relational path have also achieved better performance in knowledge graph completion due to multi-step paths contain more fertile information.

3. Our model

3.1. Entity representations

We consider that it is still insufficient in learning knowledge graph embeddings only using the learned representation for a single entity, since each entity is surrounded by multiple entities at the same time. Hence we extract additional contextual information for each entity to identify its position in knowledge graph. Given the entity e, its context embedding is calculated as the average of its contextual entities, i.e.,

$$\mathbf{e}' = \frac{1}{|context(e)|}\left(\sum_{e^i \in S_e^1}\alpha_1\mathbf{e}^i + \sum_{e^i \in S_e^2}\alpha_2\mathbf{e}^i\right)$$

where $context(e) = S_e^1 \cup S_e^2$ is the contextual entity set of entity e, and S_e^1 and S_e^2 are respectively the 1-hop and 2-hop neighbor entities of entity e, $\mathbf{e}^i$ is the embedding of entity e^i, α_1, α_2 are weights with restriction $\alpha_1 + \alpha_2 = 1$. Intuitively we incorporate the entities with its associated contextual information for entity representation as $\mathbf{e} = [\mathbf{e}, \mathbf{e}']$

3.2. NTransGH

NTransGH is a translation-neural based model composed of translation mechanism and a neural network. As Fig. 1(b) shown, it can be summed up in two steps: (1) Using generalized hyperplanes for translation, (2) Using neural network for defining a scoring function.

3.2.1. Translation

This phrase considers the translation operation on a generalized hyperplane determined by a set of basis vectors $\{\mathbf{w}_r^1, \mathbf{w}_r^2, \ldots, \mathbf{w}_r^\nu\}$, $(\mathbf{w}_r^i \in R^{2m}, i \in [1, \nu])$. The basis vectors are orthogonal to each other, and each vector is restricted as $\|\mathbf{w}_r^i\|_2 = 1$. For an entity representation $\mathbf{e}$, a transfer vector $\mathbf{e}_r$ on the set of basis vectors can be written as:

$$\mathbf{e}_r = \mathbf{w}_r^1{}^T\mathbf{e}\mathbf{w}_r^1 + \ldots + \mathbf{w}_r^\nu{}^T\mathbf{e}\mathbf{w}_r^\nu = \sum_i \mathbf{w}_r^i{}^T\mathbf{e}\mathbf{w}_r^i$$

where ν is the number of basis vectors. Based on the transfer vector $\mathbf{e}_r$, we can obtain the projection $\mathbf{e}_\perp$ of entity representation $\mathbf{e}$ on the generalized hyperplane as $\mathbf{e}_\perp = \mathbf{e} - \mathbf{e}_r$. Thus, the generalized hyperplane can be described as

$$\left\{\mathbf{e}_\perp | \mathbf{e}_\perp = \mathbf{e} - \sum_i \mathbf{w}_r^i{}^T\mathbf{e}\mathbf{w}_r^i\right\}$$

The proposed hyperplane is a generalization of that in TransH. Therefore, for a triplet (h, r, t) with its embedding representation $(\mathbf{h}, \mathbf{r}, \mathbf{t})$. NTransGH first gets the projections of entity representations $\mathbf{h}$ and $\mathbf{t}$ on the generalized hyperplane as

$$\mathbf{h}_\perp = \mathbf{h} - \sum_i \mathbf{w}_r^i{}^T\mathbf{h}\mathbf{w}_r^i, \qquad \mathbf{t}_\perp = \mathbf{t} - \sum_i \mathbf{w}_r^i{}^T s\mathbf{t}\mathbf{w}_r^i$$

Then NTransGH conducts translation operation by modeling relation $\mathbf{r}$ as a translation vector between entities projections, and obtains the triplet representation as

$$\mathbf{m} = \mathbf{M}_r\mathbf{h}_\perp + \mathbf{r} - \mathbf{M}_r\mathbf{t}_\perp$$

where $\mathbf{M}_r \in R^{n\times 2m}$ is the transfer matrix. Note that without considering the contextual information of entities, we represent $\mathbf{w}_r^i \in R^m$ and $\mathbf{M}_r \in R^{n\times m}$.

3.2.2. Scoring function

Different from the score function referred in our previous works [1], this phrase introduces a two-layer fully connected neural network to capture fertile mapping properties of complex relations, and use it to define a score function as

$$f(h, r, t) = sigmoid(\mathbf{W}_2^T[ReLu(\mathbf{W}_1\mathbf{m})])$$

where $\mathbf{W}_1 \in R^{n\times n}$, $\mathbf{W}_2 \in R^n$ are weight parameters. The score function is to measure the compatible of a positive triplet, and still, it is expected to be high for a positive triplet, otherwise low for a negative triplet. TransGH introduces generalized hyperplane to conduct the translation operation and uses a neural network to define a score function, which can achieve the generalized ability for expressing mapping properties of complex relation facts.

3.3. Training method and implementation details

We use the observed triplets as positive samples and unobserved triplets as negative samples to train our model. For each training sample $\{(h, r, t), y\}$, there is a label $y = 1$ for positive sample and $y = 0$ for negative sample. We use the following logistic loss function to encourage discrimination between positive triplets and negative triplets:

$$Ł = -\left\{\sum_{(h,r,t)\in P} y \quad log f(h, r, t) + \sum_{(h,r,t)\in N}(1-y) \quad \log \quad (1 - f(h, r, t))\right\}$$

where P is the set of positive triplets; N is the set of negative triplets, that is $N = \{(h', r, t) | (h' \in \mathbb{E} \wedge h' \neq h) \cup (h, r, t') | (t' \in \mathbb{E} \wedge t' \neq t)\}$, $\mathbb{E}$ is denoted as the entities set. Simultaneously we minimize the loss function considering the following constraints:

$$\forall e \in E, \|\mathbf{e}\|_2 \leq 1, \forall r \in R, \|\mathbf{r}\|_2 \leq 1 \tag{1}$$

$$\forall r \in R, i \in [1, \nu], \|\mathbf{w}_r^i\|_2 = 1 \tag{2}$$

Table 1
Complexity (the number of parameters and the number of multiplication operations).

Model	# Parameters	# Operations (time complexity)
UM [28]	$O(N_e m)$	$O(N_t)$
SE [29]	$O(N_e m+2N_r n^2)(m=n)$	$O(2m^2 N_t)$
LFM [30]	$O(N_e m+N_r n^2)(m=n)$	$O((m^2+m)N_t)$
SME(BILIN) [33]	$O(N_e m+N_r n+4mks+4k)(m=n)$	$O(4mksN_t)$
SLM [34]	$O(N_e m+N_r(2k+2nk))(m=n)$	$O((2mk+k)N_t)$
NTN [34]	$O(N_e m+N_r(n^2 s+2ns+2s))(m=n)$	$O(((m^2+m)s+2mk+k)N_t)$
TransE [17]	$O(N_e m+N_r n)(m=n)$	$O(N_t)$
TransH [18]	$O(N_e m+2N_r n)(m=n)$	$O(2mN_t)$
TransR [19]	$O(N_e m+N_r(m+1)n)$	$O(2mnN_t)$
TransD [20]	$O(2N_e m+2N_r n)$	$O(2nN_t)$
STransE [21]	$O(N_e 2m+N_r 2n)$	$O(2mN_t)$
TransE-RS [27]	$O(N_e m+N_r n)(m=n)$	$O(N_t)$
TransH-RS [27]	$O(N_e m+2N_r n)(m=n)$	$O(2mN_t)$
TransGH [1]	$O(N_e m+N_r(1+v)n)(m=n), v \ll m$	$O(2vmN_t)$
NTransGH (this paper)	$O(N_e m+N_r((2+2m+n)n+mv), v \ll m$	$O((4vm+4mn+n)N_t)$

$$s\forall r \in R, i, j \in [1, v](i \neq j), \frac{\left|\sum_{(i,j)} \mathbf{w}_r^{i\,T} \mathbf{w}_r^j\right|}{\|\mathbf{w}_r^j\|_2} \leq \epsilon \quad (3)$$

where ϵ is a small scalar, and R denotes relations set, constraint (3) guarantees each two basis vectors are orthogonals. Afterwards we directly optimize the following loss function with soft constraints:

$$Ł = -\left\{ \sum_{(h,r,t)\in P} y \;\; log f(h, r, t) + \sum_{(h,r,t)\in N} (1-y) \;\; \log \;\; (1 - f(h, r, t)) \right\} + C(A_1 + A_2) \quad (4)$$

where we set

$$A_1 = \sum_{e\in E}[\|\mathbf{e}\|_2^2 - 1]_+ + \sum_{r\in R}[\|\mathbf{r}\|_2^2 - 1]_+$$

$$sA_2 = \sum_{r\in R}\left[\left(\frac{\sum_{(i,j)} \mathbf{w}_r^{i\,T} \mathbf{w}_r^j}{\|\mathbf{w}_r^j\|_2}\right)^2 - \epsilon^2\right]_+$$

and C is a hyper-parameter used to measure the importance of soft constrains. The loss function minimized by (SGD) [48] favors the higher scores for positive triplets than that for negative triplets. We set each vector $\mathbf{w}_r^i$ to unit l_2-ball before traversing each mini-batch and ignore it in Eq. (4). Moreover, the traditional sampling method for negative samples is denoted as "unif" and the new method referred in [18] as "bern".[3] Moreover, we compare the parameters and time complexities with several baselines in Table 1. N_e, N_r and N_t respectively represent the number of entities, relations and triplets in a knowledge graph G, m and n separately denote the dimension of entity space and relation space. d is the average number of clusters of a relation. k is the number of hidden nodes of a neural network, s is the number of slice of a tensor. v is the number of vectors for a relations.

4. Experiments and analysis

We evaluate the effectiveness of our model on two tasks link prediction [17,18] and triplet classification [34] with two datasets FreeBase [2] and WordNet [49]. Experimental results show that our model achieves state-of-the-art performance.

[3] "unif" denotes the traditional way of replacing head or tail with equal probability, and bern denotes reducing false negative labels by replacing head or tail with different probabilities follow [18].

Table 2
Data sets used in the experiments.

DataSet	#Relation	#Entity	#Train	#Valid	#Test
FB15k	1345	14,541	272,115	17,535	20,466
WN18	18	40,943	141,442	5000	5000
FB13	13	75,043	316,232	5908	23,733
WN11	11	38,696	112,581	2609	10,544

4.1. Datasets

Experimental datasets from WordNet and FreeBase are illustrated in Table 2. WordNet represents each entity as a synset containing several words with different semantics, and relations as lexical relationships between synsets. An example of triplets in WordNet is (_ _*w arship* _*N N* _1, _*h yponym*, _ _*t orpedo* _*b oat* _*N N* _1). FreeBase is a large and rising knowledge graph of general facts. WN18 and WN11 are from WordNet, FB15k and FB13 are extracted from FreeBase.

4.2. Parameters settings

For two tasks, we use word embedding [50] to initialize our embeddings in experiments. Moreover, in training phase, we select the learning rate η for SGD from $\{0.001, 0.01, 0.1\}$, the weight $\alpha_1(\alpha_2)$ from $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$, the entity(relation) embedding dimension $m(n)$ from$\{50, 100, 150\}$, the number of vectors v from $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$, the batch size b from$\{480, 960, 1200, 4800\}$, the hyper-parameter C from $\{0.005, 0.0625, 0.25, 0.5\}$. The best parameter configurations are determined by validation set and illustrated in Table 3. We traverse all the training triplets for 5000 rounds and take *L1* as dissimilarity on both tasks.

4.3. Link prediction

Link prediction aims to predict the missing h or t for a positive triplet (h, r, t), which focuses more on ranking a set of candidate entities rather than obtaining the best one. The data sets used in this task are WN18 and FB15k.

4.3.1. Evaluation rules

In testing phase, we substitute the head(tail) entity of each test triplet (h, r, t) by every entity e from the set of entities of a KG, and use the score function $f(h, r, t)$ to calculate the scores of these corrupted triplets. We then rank these scores in ascending order and obtain the rank of original triplet. Here we utilize two metrics for evaluation: the average rank(Mean Rank) and the proportion

Table 3
Optimal parameter configurations.

Task	Datasets	Settings	η	α_1	α_2	$m(n)$	v	b	C
Link prediction	WN18	unif	0.01	0.8	0.2	100	4	4800	0.0625
		bern	0.01	0.8	0.2	100	4	1200	0.005
	FB15k	unif	0.001	0.9	0.1	100	6	1200	0.0625
		bern	0.001	0.9	0.1	100	6	480	0.0625
Triplet classification	WN11	unif	0.01	0.9	0.1	100	3	480	0.25
		bern	0.01	0.9	0.1	100	3	480	0.0625
	FB13	unif	0.01	0.8	0.2	100	5	1200	0.0625
		bern	0.01	0.8	0.2	100	5	1200	0.005
	FB15k	unif	0.001	0.9	0.1	100	8	480	0.0625
		bern	0.001	0.9	0.1	100	8	480	0.0625

Table 4
Link prediction results.

Dataset	WN18				FB15k			
Metric	Mean		Hits@10		Mean		Hits@10	
	Raw	Filt	Raw	Filt	Raw	Filt	Raw	Filt
RESCAL [35]	1180	1163	37.2	52.8	828	683	28.4	44.1
UM [28]	315	304	35.3	38.2	1074	979	4.5	6.3
SE [29]	1011	985	68.5	80.5	273	162	28.8	39.8
SME(LIN) [32]	545	533	65.1	74.1	274	154	30.7	40.8
SME(BILIN) [33]	526	509	54.7	61.3	284	158	31.3	41.3
BILINEAR [30]	469	456	71.4	81.6	283	164	26.0	33.1
TransE [17]	263	251	75.4	89.2	243	125	34.9	47.1
TransH(unif) [18]	318	303	75.4	86.7	211	84	42.5	58.5
TransH(bern) [18]	400.8	388	73.0	82.3	212	87	45.7	64.4
TransR(unif) [19]	232	219	78.3	91.7	226	78	43.8	65.5
TransR(bern) [19]	238	225	79.8	92.0	198	77	48.2	68.7
CTransR(unif) [19]	243	230	78.9	92.3	233	82	44.3	66.3
CTransR(bern) [19]	231	218	79.4	92.3	199	75	48.4	70.2
TransD(unif) [20]	242	229	79.2	92.5	211	67	49.4	74.2
TransD(bern) [20]	224	212	79.6	92.2	194	91	53.4	77.3
STransE(unif) [21])	224	211	80.8	93.2	220	69	51.5	78.4
STransE(bern) [21])	219	206	80.9	93.4	219	68	51.6	79.7
ITransF(bern) [26]	210	205	82.5	95.2	202	77	55.9	81.4
TransE-RS(unif) [27]	362	348	80.3	93.7	161	62	53.1	72.3
TransE-RS(bern) [27]	385	371	80.4	93.7	161	63	53.2	72.1
TransH-RS(unif) [27]	401	389	81.2	94.7	163	64	53.4	72.6
TransH-RS(bern) [27]	371	357	80.3	94.5	178	77	53.6	75.0
TransGH(unif) [1]	191	179	81.4	94.8	186	66	54.0	79.8
TransGH(bern) [1]	210	197	81.6	95.3	186	64	54.1	80.1
NTransGH(unif)(−)	**189**	**172**	**84.3**	**95.8**	163	**60**	**56.9**	**81.8**
NTransGH(bern)(−)	**186**	**167**	**84.9**	**96.0**	**158**	**58**	**57.5**	**82.4**
NTransGH(unif)	**169**	**153**	**86.6**	**97.5**	**143**	**53**	**60.2**	**85.1**
NTransGH(bern)	**165**	**150**	**87.1**	**97.8**	**139**	**51**	**61.7**	**85.6**

The bold values are the best performance for link prediction task.

of ranks not larger than 10 (Hit@10). We call this setting as “raw”. Notice that the corrupted triplets that may exit in the KG can rank them before the original triplet. To eliminate this case, these corrupted triplets existing in a KG are removed before ranking. We call this setting as “filt”. In both settings, lower Mean and higher Hit@10 are excepted.

4.3.2. Results

In this experiment, we compare our model with typical baselines, and also test the results denoted by (“-”) of our model with removing the contextual information from the input. Since the same data sets are used, we directly copy the results of baselines. Table 4 gives the results of WN18 and FB15k, which can be observed that (1) Our model NTransGH consistently and significantly achieves the highest performance among previous baselines on all the data sets with all the metrics. It indicates that NTransGH can boost the capability of finding missing facts and completing knowledge graphs. (2) The usage of contextual information of entities increases the performance on both datasets, which is indeed beneficial to express complex relations and learn knowledge graph embeddings. (3) NTransGH has respectively remarkable improvements on metrics of Mean(filt) and Hit@10(filt) comparing with TransGH, which brings a 29, 2.3% gain on WN18, and 13, 5.5% gain on FB15k based on TransGH. We attribute this to the introduction of the neural network and the logistic loss function.

Moreover we analyze Hits@10 results on FB15k with respect to the relation categories,[4] and illustrate the results in Table 5. From Table 5, we can see that NTransGH outperforms all baselines on both *unif* and *bern* settings, which has the highest accuracies on predicting head(one-to-one 93.0%, one-to-many 98.6%, many-to-one 61.4% and many-to-many 86.8%) and predicting tail (one-to-one 91.1%, one-to-many 66.3%, many-to-one 98.5% and many-to-many 87.6%). Additionally, comparing with TransGH, we also give the results on Hit@10 metric of some typical complex relations in Table 6. Table 6 shows NTransGH leads better and more stable

[4] Following the same rules in [17] on FB15k, we separate the 1345 relations into four categories, including one-to-one, one-to-many, many-to-one and many-to-many relations.

Table 5
Results on FB15k by relation category.

Dataset	Predicting left (Hit@10)				Predicting right (Hit@10)			
Relation category	1-to-1	1-to-*n*	*n*-to-1	*n*-to-*n*	1-to-1	1-to-*n*	*n*-to-1	*n*-to-*n*
RESCAL [35]	29.1	41.0	37.8	33.0	48.2	30.8	35.1	55.4
UM [28]	34.5	2.5	6.1	6.6	34.3	4.2	1.9	6.6
SE [29]	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME(LIN) [32]	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME(BILIN) [33]	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE [17]	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH(unif) [18]	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransH(bern) [18]	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR(unif) [19]	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
TransR(bern) [19]	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
CTransR(unif) [19]	78.6	77.8	36.4	68.0	77.4	37.8	78.0	70.3
CTransR(bern) [19]	81.5	89.0	34.7	71.2	80.8	38.6	90.1	73.8
TransD(unif) [20]	80.7	85.8	47.1	75.6	80.0	54.5	80.7	77.9
TransD(bern) [20]	86.1	95.5	39.8	78.5	85.4	50.6	94.4	81.2
STransE(unif) [21]	82.1	94.1	49.3	79.6	79.8	55.2	93.1	81.8
STransE(bern) [21]	82.8	94.2	50.4	80.1	82.4	56.9	93.4	83.1
ITransF(bern) [26]	85.0	95.4	52.5	82.8	84.8	57.2	94.5	84.5
TransE-RS(unif) [27]	87.2	96.2	35.9	71.8	87.0	45.0	95.5	75.4
TransE-RS(bern) [27]	87.4	96.3	35.3	71.7	86.5	44.2	95.4	75.2
TransH-RS(unif) [27]	87.6	95.9	35.6	72.5	86.3	44.9	95.5	75.8
TransH-RS(bern) [27]	85.6	95.5	37.4	75.5	85.7	47.4	94.9	78.7
TransGH(unif) [1]	86.4	95.6	47.6	80.6	85.8	55.8	94.8	83.4
TransGH(bern) [1]	87.0	95.8	47.9	80.8	86.8	55.7	94.8	84.3
NTransGH(unif)(−)	**88.2**	**96.9**	**52.8**	**83.0**	**87.7**	**58.1**	**95.3**	**84.8**
NTransGH(bern)(−)	**89.3**	**97.6**	**54.4**	**83.9**	**88.3**	**59.0**	**96.2**	**85.2**
NTransGH(unif)	**92.3**	**98.4**	**59.2**	**86.4**	**90.6**	**64.7**	**98.1**	**87.3**
NTransGH(bern)	**93.0**	**98.6**	**61.4**	**86.8**	**91.1**	**66.3**	**98.5**	**87.6**

The bold values are the best performance for link prediction task.

Table 6
Hits@10(filt)*bern* of TransGH and NTransGH on some examples of one-to-many*, many-to-one†, many-to-many‡ and symmetric§ relations.

Relations	TransGH/NTranGH on FB15k(Hit@10)	
	Predict head	Predict tail
/football_position/players*	100/100	88.9/**90.7**
/production_company/films*	96.8/**98.2**	52.4/**60.3**
/director/film*	96.2/**98.8**	94.3/**97.1**
/disease/treatments†	66.6/**70.4**	100/100
/person/place_of_birth†	77.9/**80.8**	92.0/**94.3**
/film/production_companies†	47.5/**55.9**	96.7/**99.8**
/field_of_study/students_majoring‡	92.2/**97.4**	70.5/**85.4**
/award_winner/awards_won‡	99.0/**100**	99.5/**100**
/sports_position/players‡	100/100	99.6/**100**
/person/sibling_s§	68.4/**80.1**	68.4/**85.4**
/person/spouse_s§	70.4/**84.6**	59.3/**75.3**

The bold values are the best performance for link prediction task.

results on some typical complex relations compared with TransGH. It indicates the power of neural network for capturing fertile entity-relation interactions, and the ability of modeling mapping properties of complex relation facts.

4.4. *Triplet classification*

Triplet classification is to verify whether a given triplet (h, r, t) is correct or not. This is a binary classification task that needs negative triplets. The datasets WN11 and FB13 referred in [34] have already negative triplets, and FB15k including negative triplets has not been published. We adopt the same rules for FB13 in [34] to construct negative triplets for FB15k.

4.4.1. *Evaluation rules*

We use following decision rule for triplet classification: first a relation-specific threshold δ_r is obtained by maximizing the classification accuracy on the validation set, then each triplet (h, r, t) is predicted as positive if its score calculated by the score function $f(h, r, t)$ is above δ_r, otherwise the triplet is predicted as negative.

Table 7
Triplet classification accuracies.

Dataset	WN11	FB13	FB15k
NTN [34]	70.4	87.1	–
SE [29]	53.0	75.2	72.2
SME [32]	70.0	63.7	71.6
TransE(unif) [17]	75.9	70.9	79.5
TransE(bern) [17]	75.9	81.5	80.4
TransH(unif) [18]	77.7	76.5	79.9
TransH(bern) [18]	78.8	83.3	80.0
TransR(unif) [19]	85.5	74.7	81.2
TransR(bern) [19]	85.9	82.5	82.5
TransD(unif)[20]	85.6	85.9	86.0
TransD(bern)[20]	86.4	89.1	88.2
TransE-RS(bern)[27]	85.3	83.0	81.9
TransH-RS(bern)[27]	86.4	81.6	83.2
STransE(bern)[21]	86.4	81.6	83.2
ITransF(bern)[26]	87.4	82.3	84.4
TransGH(unif)[1]	87.2	84.7	91.4
TransGH(bern)[1]	87.3	85.2	91.2
NTransGH(unif)(−)	**87.6**	85.8	**91.8**
NTransGH(bern)(−)	**87.7**	86.0	**91.7**
NTransGH(unif)	**88.4**	**87.3**	**92.6**
NTransGH(bern)	**88.8**	**87.5**	**92.9**

The bold values are the best performance for link prediction task.

4.4.2. *Results*

Table 7 gives the evaluation results of triplet classification, which indicates that (1) NTransGH consistently scores better accuracy on WN11, FB13 and FB15k than the current state-of-the-art model, where accuracies are 88.8%(WN11), 87.5%(FB13) and 92.9%(FB15k) respectively. (2) NTransGH achieves at least 0.3%, 0.6%, 0.6% higher than TransGH on three datasets. Therefore we believe the introduction of neural network is beneficial to model complex relations and learn entity and relation embeddings.

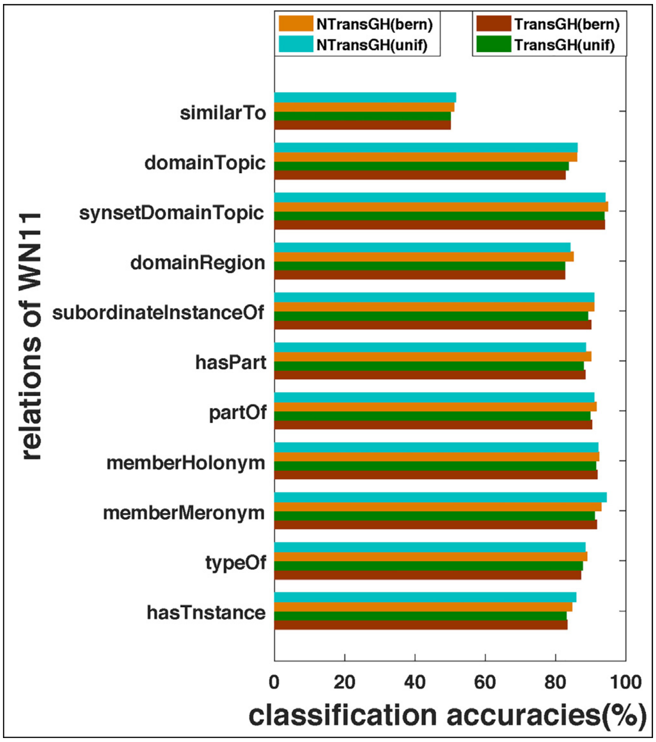


Fig. 2. Classification accuracies of on WN11.

(3) Incorporating contextual information of entities can substantially improve the performance of NTransGH. It demonstrates the effectiveness of leveraging additional information to deal with expressing complex relations. We also compare the classification accuracies of different relations by TransGH and NTransGH on WN11 in Fig. 2, which shows that NTransGH separately improves TransGH in each relation classification.

5. Discussion

Our model involves a number of hyper-parameters. In this section, we evaluate how different choices of hyper-parameters affect the performance of our model. In the following experiments, except for the parameter being tested, all other parameters are set as optimal configurations illustrated in Table 3.

5.1. The weight of contextual information of entities α_1 and α_2

As described in above, the usage of additional contextual information of entities is beneficial for improving prediction performance. This section explores the effect of the weight α_1 and α_2 from contextual information on the performance. In this experiment, considering the restriction $\alpha_1 + \alpha_2 = 1$, we select α_1 from set $\{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0\}$. Fig. 3 gives the convinced results, where FB15k(N) is used in triplet classification. From Fig. 3 we can see that: (1) NTransGH has better performance when $\alpha_1 = \{0.9, 0.8\}$ on link prediction and triplet classification tasks, indicating than such setting best express contextual information of entity space and identify the position of entities. (2) The performance of our model is first increasing and then declining with the decrease of α_1. It proves that the nearest neighbor entities play the most important for representing entity embeddings, i.e., the 1-hop ripple set of entity stores more contextual information than 2-top ripple set of entity.

5.2. Different dimension of entity embedding m and relation embedding n

In this section, we mainly explore the effect of different dimension of entity embedding m and relation embedding n on the performance. In this experiment, we just test our model on link prediction with FB15k(hit@10), and select the dimensions from all combinations of m and n in set $\{20, 50, 100, 150, 200\}$. Fig. 4 gives the convinced results, which are (1) Our model achieves best performance at $m = n = 100$, indicating that such dimension setting best express semantic information of entity space and relation space. (2) Given the dimension of entity embedding m, the performance of our model first is increasing with the growth of n and then drops with n further increases. That is, too small or too large dimension can reduce the performance.

It is because that too low dimension has insufficient capability of capturing the necessary information and properties, too large dimension introduces unnecessary noise and reduces generalization ability. The case is similar for m when n is given.

5.3. The number of basis vectors v

This section mainly investigates how different number of basis vectors v make influence on performance. In translation phrase of our model, we introduce the generalized hyperplane determined by a set of basis vector to model relations as translation operations between entities. In this experiment, we test our model with v from $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$. The results are shown in Fig. 5, which we can analyse that the performance on all datasets

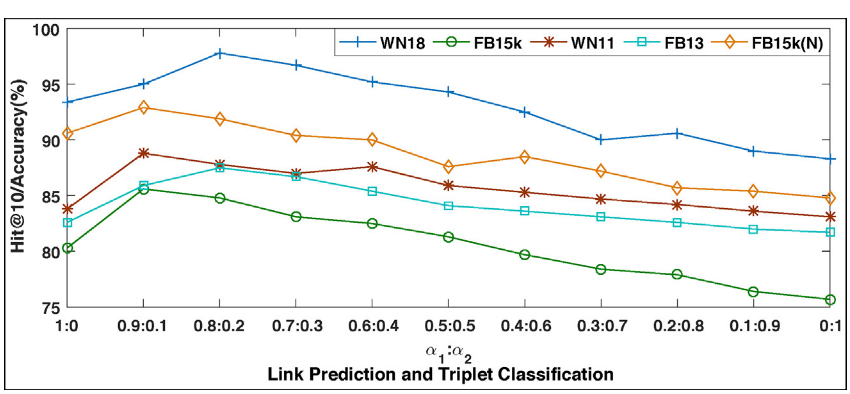


Fig. 3. Performance on different combinations of α_1 and α_2.

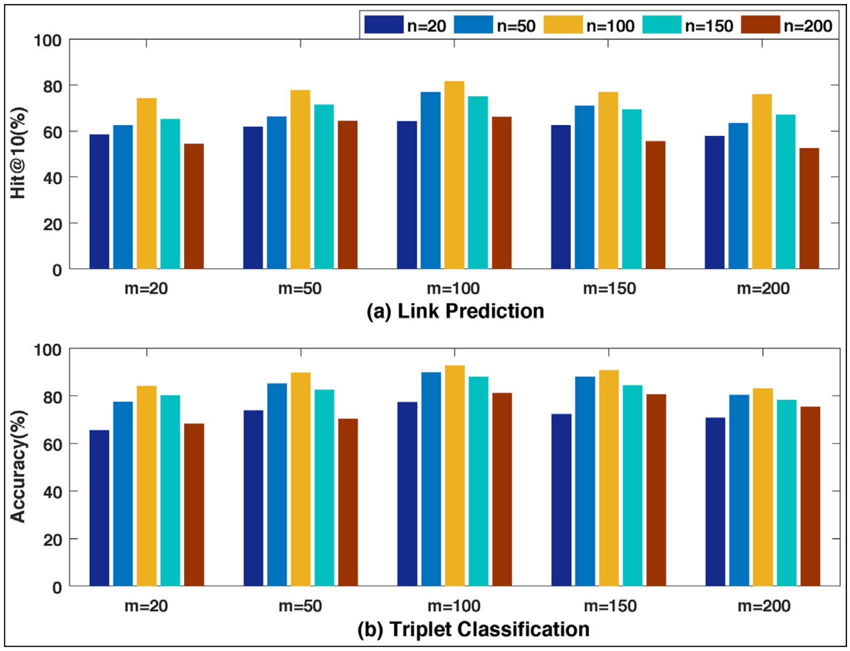


Fig. 4. Performance on different dimension of entity embedding m and relation embedding n.

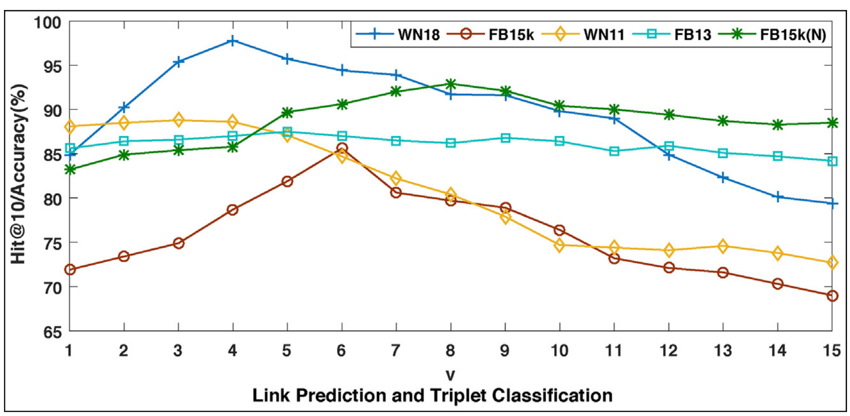


Fig. 5. Performance on different number of basis vectors v.

begins to gradually rise to the highest point and then declines as the number of basis vectors v grows. Since too small v can not capture the richer correlation between entity space and relation space, and too large v may introduce noisy and lead to over-fitting problem.

6. Conclusion

In this paper, we propose NTransGH, a translation-neural based model for learning knowledge graph embeddings. The key idea of NTransGH is first modeling relations as translation operations between entities through generalized hyperplanes and then utilizing a neural network to capture rich interactions between entities and relations. NTransGH can not only preserve but also express fertile mapping properties of complex relations. Moreover, we introduce the context of entities as complementary information for fully expressing the semantic information of the entity and improving the identifiability of entities. We empirically conduct extensive experiments on triplet classification and link prediction with two knowledge graphs FreeBase and WordNet. The experimental results show that NTransGH has significantly and consistently considerable improvement over baselines, and achieves state-of-the-art performance, which demonstrates the superiority and generality of our model.

Acknowledgements

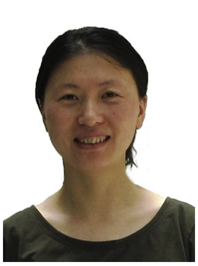
This work is supported by National Key R&D Program No. 2017YFB0803003, and the National Natural Science Foundation of China (No. 61202226), We thank all anonymous reviewers for their constructive comments.

References

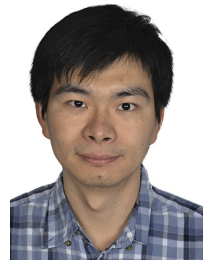
[1] Q. Zhu, X. Zhou, J.L. Tan, P. Liu, L. Guo, Learning knowledge graph embeddings via generalized hyperplanes, International Conference on Computational Science (2018) 624–638, http://dx.doi.org/10.1007/978-3-319-93698-7_48.
[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada (2008, June) 1247–1250, http://dx.doi.org/10.1145/1376616.1376746.
[3] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P.V. Kleef, S. Auer, Dbpedia – a large-scale, multilingual knowledge base extracted from wikipedia, Semant. Web 6 (2) (2015) 167–195.

[4] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, T.M. Mitchell, Toward an architecture for never-ending language learning, Twenty-Fourth AAAI Conference on Artificial Intelligence (2010) 1306–1313, http://dx.doi.org/10.1037/e660332010-001.
[5] A. Bordes, J. Weston, N. Usunier, Open question answering with weakly supervised embedding models, European Conference on Machine Learning and Knowledge Discovery in Databases (2014) 165–180, http://dx.doi.org/10.1007/978-3-662-44848-9_11.
[6] M. Yahya, K. Berberich, G. Elbassuoni, S. Weikum, Robust question answering over the web of linked data, Proceedings of the 22nd ACM international conference on CIKM, Association for Computational Linguistics (2013) 1107–1116, http://dx.doi.org/10.1145/2505515.2505677.
[7] A. Bordes, S. Chopra, J. Weston, Question answering with subgraph embeddings, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, 2014, pp. 615–620, http://dx.doi.org/10.3115/v1/d14-1067.
[8] S. He, K. Liu, Y. Zhang, L. Xu, J. Zhao, Question answering over linked data using first-order logic, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, 2014, pp. 1092–1103, http://dx.doi.org/10.3115/v1/d14-1116.
[9] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, D.S. Weld, Knowledge-based weak supervision for information extraction of overlapping relations, Meeting of the Association for Computational Linguistics: Human Language Technologies (2011) 541–550.
[10] S. Riedel, L. Yao, A. Mccallum, B.M. Marlin, Relation extraction with matrix factorization and universal schemas, in: NAACL-HLT, 2013, pp. xxi–xxii.
[11] J. Weston, A. Bordes, O. Yakhnenko, N. Usunier, Connecting language and knowledge bases with embedding models for relation extraction, Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18–21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL (2013) 1366–1371.
[12] Y. Lin, S. Shen, Z. Liu, H. Luan, M. Sun, Neural relation extraction with selective attention over instances, in: Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2016, pp. 2124–2133, http://dx.doi.org/10.18653/v1/p16-1200.
[13] R.J. Brachman, H.J. Levesque, Knowledge Representation and Reasoning, Elsevier, 2004, http://dx.doi.org/10.1016/b978-1-55860-932-7.x5083-3.
[14] R.A. Scarpel, A. Ladas, U. Aickelin, Indebted households profiling: a knowledge discovery from database approach, Ann. Data Sci. 2 (1) (2015) 1–17.
[15] B. Caroleo, E. Palumbo, M. Osella, A. Lotito, G. Rizzo, E. Ferro, A. Attanasio, S. Chiusano, G. Zuccaro, M. Leone, D.D. Gregorio, A knowledge-based multi-criteria decision support system encompassing cascading effects for disaster management, Int. J. Inf. Technol. Decis. Mak. 17 (5) (2018).
[16] J. Li, S. Zhang, W. Wang, Y. Wang, Research on the influencing factors of user's online knowledge consumption behavior, in: Proceedings of the 5th International Conference on Information Technology and Quantitative Management, ITQM 2017, Creating Knowledge and Wisdom via Big Data Analytics, December 8–10, 2017, New Delhi, India, 2017, pp. 174–179.
[17] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States, 2013, pp. 2787–2795.
[18] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27–31, 2014, Québec City, Québec, Canada, 2014, pp. 1112–1119.
[19] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25–30, 2015, Austin, Texas, USA (2015) 2181–2187.
[20] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26–31, 2015, Beijing, China, Volume 1: Long Papers (2015) 687–696.
[21] D.Q. Nguyen, K. Sirts, L. Qu, M. Johnson, Stranse: a novel embedding model of entities and relationships in knowledge bases, Twenty-Fourth AAAI Conference on Artificial Intelligence (2016) 1306–1313, http://dx.doi.org/10.18653/v1/n16-1054.
[22] G. Ji, K. Liu, S. He, J. Zhao, Knowledge graph completion with adaptive sparse transfer matrix, Thirtieth AAAI Conference on Artificial Intelligence (2016) 985–991.
[23] M. Fan, Q. Zhou, E. Chang, T.F. Zheng, Transition-based knowledge graph embedding with relational mapping properties, Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation, PACLIC 28, Cape Panwa Hotel, Phuket, Thailand, December 12–14, 2014 (2014) 328–337.
[24] S. He, K. Liu, G. Ji, J. Zhao, Learning to represent knowledge graphs with gaussian embedding, ACM International on Conference on Information and Knowledge Management (2015) 623–632, http://dx.doi.org/10.1145/2806416.2806502.
[25] H. Xiao, M. Huang, X. Zhu, Transg:, A generative model for knowledge graph embedding, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7–12, 2016, Berlin, Germany, Volume 1: Long Papers, 2016, http://dx.doi.org/10.18653/v1/p16-1219.
[26] Q. Xie, X. Ma, Z. Dai, E. Hovy, An interpretable knowledge transfer model for knowledge base completion, in: Meeting of the Association for Computational Linguistics, 2017, pp. 950–962, http://dx.doi.org/10.18653/v1/p17-1088.
[27] X. Zhou, Q. Zhu, P. Liu, L. Guo, Learning knowledge embeddings by combining limit-based scoring loss, in: CIKM 2017, ACM Press, 2017, pp. 1009–1018, http://dx.doi.org/10.1145/3132847.3132939.
[28] A. Bordes, X. Glorot, J. Weston, Y. Bengio, Joint learning of words and meaning representations for open-text semantic parsing, Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, Spain, April 21–23, 2012 (2012) 127–135.
[29] A. Bordes, J. Weston, R. Collobert, Y. Bengio, Learning structured embeddings of knowledge bases, AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA (2011, August).
[30] R. Jenatton, N.L. Roux, A. Bordes, G. Obozinski, A latent factor model for highly multi-relational data, International Conference on Neural Information Processing Systems (2012) 3167–3175.
[31] I. Sutskever, R. Salakhutdinov, J.B. Tenenbaum, Modelling relational data using Bayesian clustered tensor factorization, in: Advances in Neural Information Processing Systems 22: Conference on Neural Information Processing Systems 2009. Proceedings of A Meeting Held 7–10 December 2009, Vancouver, British Columbia, Canada, 2009, pp. 1821–1828.
[32] A. Bordes, X. Glorot, J. Weston, Joint learning of words and meaning representations for open-text semantic parsing, International Conference on Artificial Intelligence and Statistics (2011).
[33] A. Bordes, X.G.J. Weston, Y. Bengio, A semantic matching energy function for learning with multi-relational data, CoRR (2013), http://dx.doi.org/10.1007/s10994-013-5363-6, abs/1301.3485.
[34] R. Socher, D. Chen, C.D. Manning, A.Y. Ng, Reasoning with neural tensor networks for knowledge base completion, Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States (2013) 926–934.
[35] M. Nickel, V. Tresp, H. Kriegel, A three-way model for collective learning on multi-relational data, Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28–July 2, 2011 (2011) 809–816.
[36] R.A. Harshman, M.E. Lundy, Parafac: Parallel factor analysis, Comput. Stat. Data Anal. 18 (1) (1994) 39–72, http://dx.doi.org/10.1016/0167-9473(94)90132-5.
[37] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279, http://dx.doi.org/10.1007/bf02289464.
[38] R.A. Harshman, Models for analysis of asymmetrical relationships among n objects or stimuli, Psychometrika 31 (3) (1966) 279.
[39] T.G. Kolda, B.W. Bader, J.P. Kenny, Higher-order web link analysis using multilinear algebra, 2005, pp. 242–249, http://dx.doi.org/10.2172/974401.
[40] T. Franz, A. Schultz, S. Sizov, S. Staab, Triplerank: ranking semantic web data by tensor decomposition, International Semantic Web Conference (2009) 213–228, http://dx.doi.org/10.1007/978-3-642-04930-9_14.
[41] L. Drumond, S. Rendle, L. Schmidt-Thieme, Predicting rdf triples in incomplete knowledge bases with tensor factorization, ACM Symposium on Applied Computing (2012) 326–331, http://dx.doi.org/10.1145/2245276.2245341.
[42] S. Rendle, L. Schmidt-Thieme, Pairwise interaction tensor factorization for personalized tag recommendation, ACM International Conference on Web Search and Data Mining (2010) 81–90, http://dx.doi.org/10.1145/1718487.1718498.
[43] S. Rendle, Scaling factorization machines to relational data, International Conference on Very Large Data Bases (2013) 337–348, http://dx.doi.org/10.14778/2535573.2488340.
[44] P. Miettinen, Boolean tensor factorizations, IEEE International Conference on Data Mining (2012) 447–456, http://dx.doi.org/10.1109/icdm201128.
[45] A. Garca-Durn, A. Bordes, N. Usunier, Composing relationships with translations.
[46] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, S. Liu, Modeling relation paths for representation learning of knowledge bases, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17–21, 2015, 2015, pp. 705–714.
[47] Z. Wei, J. Zhao, K. Liu, Mining inference formulas by goal-directed random walks, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1–4, 2016, 2016, pp. 1379–1388.
[48] L. Bottou, Large-scale machine learning with stochastic gradient descent, Proceedings of COMPSTAT 2010, Physica-Verlag HD (2010) 177–186, http://dx.doi.org/10.1007/978-3-7908-2604-3_16.
[49] G.A. Miller, Wordnet: a lexical database for English, Commun. ACM 38 (11) (1995) 39–41.
[50] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a Meeting Held December 5–8, 2013, Lake Tahoe, Nevada, United States, 2013, pp. 3111–3119.

Qiannan Zhu received the MS degree in the Institute of Information Engineering, University of Chinese Academy of Sciences, in 2015. She is currently working toward the PhD degree at the Institute of Information Engineering, Chinese Academy of Sciences. Her current research interests include data mining, knowledge graph representation, deep learning and natural language process.



Xiaofei Zhou received the PhD degree in Pattern Recognition and Artificial Intelligence from Nanjing University of Science and Technology, in 2008. She as a Postdoctor worked in Graduate University of Chinese Academy of Sciences, from 2008 to 2010. She is currently an associate professor in the Institute of Information Engineering, Chinese Academy of Sciences. Her research interests include machine learning, knowledge graph representation, deep learning, natural language processing and pattern recognition.



Peng Zhang received PhD from the Chinese Academy of Sciences. He is a Senior Data Scientist with the Ant Financial Services Group. Before then, he was a Lecturer with the University of Technology Sydney in Australia. His research focuses on data mining and deep learning. He continuously serves as a PC Member in leading data mining and deep learning conferences, such as ICLR, KDD, NIPS, ICML, IJCAI and AAAI.

Yong Shi is a professor in School of Economics and Management, University of Chinese Academy of Sciences. He is also the executive deputy director in CAS Research Center on Fictitious Economy & Data Science. His current research interests are data mining, information overload, optimal system designs, multiple criteria decision making, decision support systems, and information and telecommunications management.