

Knowledge Graph Embedding by Flexible Translation

Jun Feng, Mantong Zhou, Yu Hao, Minlie Huang, Xiaoyan Zhu

State Key Lab. of Intelligent Technology and Systems

National Lab. for Information Science and Technology

Dept. of Computer Science and Technology, Tsinghua University, Beijing 100084, PR China

{JuneFeng.81, zmt.keke}@gmail.com, haoyu@mail.tsinghua.edu.cn, {aihuang, zxy-dcs}@tsinghua.edu.cn

Abstract

Knowledge graph embedding refers to projecting entities and relations in knowledge graph into continuous vector spaces. State-of-the-art methods, such as TransE, TransH, and TransR build embeddings by treating relation as translation from head entity to tail entity. However, previous models can not deal with reflexive/one-to-many/many-to-one/many-to-many relations properly, or lack of scalability and efficiency. Thus, we propose a novel method, flexible translation, named TransF, to address the above issues. TransF regards relation as translation between head entity vector and tail entity vector with flexible magnitude. To evaluate the proposed model, we conduct link prediction and triple classification on benchmark datasets. Experimental results show that our method remarkably improve the performance compared with several state-of-the-art baselines.

1 Introduction

Knowledge graphs such as Wordnet [Miller, 1995] and Freebase [Bollacker *et al.*, 2008] are useful resources for natural language understanding, question answering, web search, etc. A knowledge graph contains highly-structured and well-organized data, and is usually represented by directed graph in which nodes correspond to *entities* and edges correspond to *relation*, or simply by a set of triples (*head entity, relation, tail entity*) ((h, r, t) for short). Although there have been substantial achievements in building large-scale knowledge graph, the general paradigm to support computing is not clear. Indeed, traditional knowledge graphs are symbolic and logical frameworks which are not flexible enough to be fruitfully exported, especially to statistical learning approaches which require the knowledge to be computable in numerical forms.

Recently, knowledge graph embedding, which projects entity or/and relation into continuous vector spaces, has been a new proposal to offer the powerful capability of computing on knowledge graph. In this paradigm, the embedding representation of a single entity/relation encodes the global information of the entire knowledge graph, since it is obtained by minimizing a global loss function involving all entities and relations. Furthermore, the embedding representations are ben-

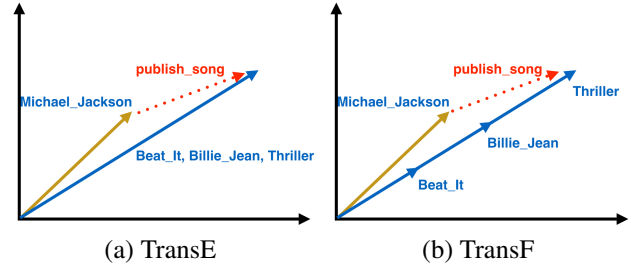


Figure 1: Illustration of TransE and our proposed model TransF. The idea of TransF is to ensure that the sum vector of a head entity vector and a relation vector has the same direction with a tail entity vector but with flexible magnitude.

eficial to a variety of applications such as question answering and web search concerning knowledge computation. Taking knowledge graph completion as an example, we can simply judge the correctness of a triple (h, r, t) by checking the compatibility of the embedding vectors of h , r and t .

A variety of approaches have been explored for knowledge graph embedding, such as neural network based [Bordes *et al.*, 2011; Socher *et al.*, 2013], and translation based [Bordes *et al.*, 2013] approaches. Some notable works, including TransE [Bordes *et al.*, 2013], TransH [Wang *et al.*, 2014], and TransR [Lin *et al.*, 2015] are simple, efficient, and effective.

Inspired by [Mikolov *et al.*, 2013], TransE learns vector embedding for both entities and relations. In TransE, relation is represented as translation from head entity to tail entity in the embedding space, which applies $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, if (h, r, t) holds. However, TransE does not work well when dealing with relations of reflexive/one-to-many/many-to-one/many-to-many. Taking a one-to-many relation *publish_song* as the example, we have triples such as $(\text{Michael_Jackson}, \text{publish_song}, \text{Beat_It})$, $(\text{Michael_Jackson}, \text{publish_song}, \text{Billie_Jean})$ and $(\text{Michael_Jackson}, \text{publish_song}, \text{Thriller})$. As shown in Figure 1(a), considering the ideal embedding where $\mathbf{h} + \mathbf{r} = \mathbf{t}$, the entities *Beat_It*, *Billie_Jean* and *Thriller* will get the same embedding vectors. Although TransE does not enforce $\mathbf{h} + \mathbf{r} = \mathbf{t}$ for golden triples during training, the tendency still exists.

As a variant of TransE, TransR models entities and relations in separate spaces and performs translation in the cor-

responding relation space. Unfortunately, since TransR applies $\mathbf{h}_r + \mathbf{r} \approx \mathbf{t}_r$, if (h, r, t) holds, the problem of TransE also exists in the TransR model. Moreover, TransR are with much more parameters and cost much more training time than TransE.

TransH is proposed to solve this problem by interpreting a relation as a translating operation on a hyperplane, but it introduces more parameters than TransE. Accordingly, the model complexity and training time of TransH is largely increased. As shown in Table 4, the training time of TransH is nearly 24 times of that of TransE and TransF.

To address the above issues, we aim to propose an effective and simple model so that it can overcome the flaws in TransE, but also with good efficiency. The central idea of our model is well-motivated: we regard the entities as vectors and relations as translation between head entity vector and tail entity vector with flexible magnitude, instead of enforcing the strict transition based on the vector addition. We call it Flexible Translation(TransF). Here, **flexible translation** means unlike conventional vector addition, say $\mathbf{h} + \mathbf{r} = \mathbf{t}$, the left and right terms need to have the same direction and the same magnitude, flexible translation only needs to keep the same direction between the two terms. Therefore, unlike TransE assumes $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ if (h, r, t) holds, our proposed model takes $\mathbf{h} + \mathbf{r} \approx \alpha \mathbf{t}$, where the flexibility is reflected in α . This relaxation not only fixes the flaws of TransE on modeling reflexive/one-to-many/many-to-one/many-to-many relations, but is practically very efficient. Taking the same one-to-many relation *publish_song* as the example, as shown in Figure 1(b), the ideal embedding where $\mathbf{h} + \mathbf{r} \approx \alpha \mathbf{t}$ is that the embedding vectors of three entities *Beat_It*, *Billie_Jean* and *Thriller* are with the same direction but are of different vector magnitudes. Actually, the idea of TransF is general. With applying to the state-of-art models, remarkable improvements can be obtained. As an example, in this paper, we upgrade the well performed TransR model to TransRF, which obeys $\mathbf{h}_r + \mathbf{r} \approx \alpha \mathbf{t}_r$. Following the principle of flexible translation, our proposed models contribute remarkable improvements over the state-of-the-art models.

To sum up, the characteristics and contributions of our work include:

- **Effectiveness:** Our TransF model successfully solve the flaws in TransE when dealing with reflexive/one-to-many/many-to-one/many-to-many relations, and achieve significant improvements over the baseline models including TransH in the link prediction and triple classification tasks. Furthermore, our model can better distinguish the golden and corrupted triples compared with TransE.
- **Efficiency:** Our TransF model learns only one embedding space for both entity and relation, without any extra parameters, and as a result, the training process is very fast.
- **Portability:** Our model is general in nature. The principle of flexible translation can be used to not only improve TransE (the resulting model is transF) but also improve TransR (the resulting model is TransRF). Both models obtain substantial improvements over the origi-

nal versions.

The rest of this paper are organized as follows. In Section 2, we introduce some related works of knowledge graph embedding. In Section 3, we explain our TransF model. Section 4 presents two experiments: link prediction, triple classification. Finally, we conclude the paper in Section 5.

2 Related Work

There are a variety of models for knowledge graph embedding. Each model projects entities and relations into a continuous vector space and the triples are assigned with scores to represent their correctness. Two lines of works are surveyed here.

2.1 Translation-based Models

TransE [Bordes *et al.*, 2013] represents relationships by translation vectors in an embedding space. It assumes that if (h, r, t) holds, the embedding of the tail entity t should be close to that of the head entity h plus the relation vector r , i.e., $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. Hence, TransE adopts $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{l_{1/2}}$ as the score which is low when the triple is correct and high otherwise. TransE is very efficient and performs well to one-to-one relations whilst bad in dealing with reflexive /one-to-many/many-to-one/many-to-many relations.

TransH [Wang *et al.*, 2014] attempts to alleviate the problems of TransE when dealing with reflexive /one-to-many/many-to-one/many-to-many relations by modeling a relation as a relation-specific hyperplane with \mathbf{w}_r as the normal vector together with a translation operation \mathbf{d}_r on it. After projecting entity embeddings \mathbf{h} and \mathbf{t} to the hyperplane, on the same assumption as in TransE, the projections \mathbf{h}_\perp and \mathbf{t}_\perp are connected by \mathbf{d}_r in a new space. Thus, TransH defines a scoring function $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h}_\perp + \mathbf{d}_r - \mathbf{t}_\perp\|_2^2$ to measure the plausibility that the triple is incorrect.

TransR [Lin *et al.*, 2015] addresses the issue that some entities are similar in the entity space but comparably different in other specific aspects. The model builds embeddings in distinct entity space and multiple relation spaces and performs translation in the relation space. In TransR, each relation r is associated with a mapping matrix \mathbf{M}_r . Entities may be projected from the entity space to the relation space as \mathbf{h}_r and \mathbf{t}_r . Similar to TransE, the score function is correspondingly defined as $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_2^2$ which produces lower scores for golden triples.

As mentioned, TransE, TransH, and TransR are all based on the score function $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{l_{1/2}}$ of TransE. However, TransE ignores the relation categories and does not work well in dealing with one-to-many/many-to-one/many-to-many relations. Although TransH is capable of fixing the flaw, it introduces additional parameters, which sacrifices the efficiency of the model. For the TransR model, it still have the same problem as TransE.

2.2 Other Related Models

Besides TransE, TransH and TransR, there are many other models proposed for knowledge graph embedding. We introduce several typical models.

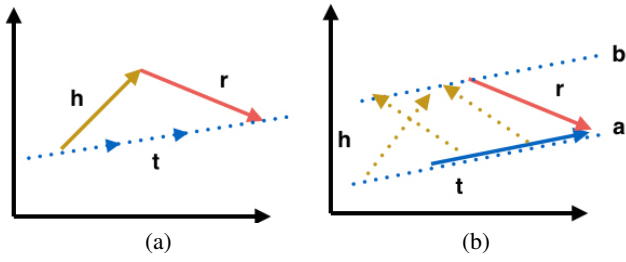


Figure 2: The different embedding vector range of tail entity(t) and head entity(h). Given h and r , the range of t is a line; given t and r , the range of h is a plane.

Unstructured Model [Bordes *et al.*, 2014] regards as the naive case of TransE, which sets all translation $\mathbf{r} = \mathbf{0}$, i.e., the score function becomes $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} - \mathbf{t}\|_{l_{1/2}}$. Obviously, it cannot distinguish different relations. Structured Embedding [Bordes *et al.*, 2011] introduces two independent projections for the entities in a relation. The basic idea is that when two entities belong to the same triple, their embedding should be close to each other in some subspace. The score function for the triple is defined as $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{W}_{r,1}\mathbf{h} - \mathbf{W}_{r,2}\mathbf{t}\|_{l_1}$. As pointed out by [Socher *et al.*, 2013], this model is weak to capture the correlations between entities and relations because of the introduction of two separate matrices. Semantic Matching Energy [Bordes *et al.*, 2014] captures the interactions of relation vectors and entity vectors through multiple matrix Hadamard products. Accordingly, the score function has the following linear form $f_r(\mathbf{h}, \mathbf{t}) = (\mathbf{W}_1\mathbf{h} + \mathbf{W}_2\mathbf{r} + \mathbf{b}_1)^\top(\mathbf{W}_3\mathbf{t} + \mathbf{W}_4\mathbf{r} + \mathbf{b}_2)$ or the bilinear form $f_r(\mathbf{h}, \mathbf{t}) = (\mathbf{W}_1\mathbf{h} \otimes \mathbf{W}_2\mathbf{r} + \mathbf{b}_1)^\top(\mathbf{W}_3\mathbf{t} \otimes \mathbf{W}_4\mathbf{r} + \mathbf{b}_2)$. Parameters are shared by all relations. Neural Tensor Network [Socher *et al.*, 2013] defines an expressive score function as $f_r(\mathbf{h}, \mathbf{t}) = \mathbf{u}_r^\top g(\mathbf{h}^\top \mathbf{M}_r \mathbf{t} + \mathbf{M}_{r,1}\mathbf{h} + \mathbf{M}_{r,2}\mathbf{t} + \mathbf{b}_r)$ where \mathbf{u}_r is a relation-specific linear layer, $g(\cdot)$ is the \tanh operation. Even when the tensor \mathbf{W}_r degenerates to a matrix, it covers most other models. However the model complexity is much higher, making it difficult to handle large scale graphs.

We also treat Single Layer Model [Socher *et al.*, 2013], Latent Factor Model [Jenatton *et al.*, 2012] and RESCAL [Nickel *et al.*, 2011; Nickel *et al.*, 2012] as our baseline in experiments.

3 Method

To address the issues of TransE, TransH, and TransR as mentioned before, we propose a knowledge graph embedding model based on flexible translation (TransF). TransF can essentially overcome the problems of TransE in modeling reflexive/one-to-many/many-to-one/many-to-many relations. The details of the model are described in Section 3.1. Following the same principle of flexible translation, we also propose TransRF model which is an improved variant of TransR, as presented in Section 3.2.

Let's introduce some notations. S denotes a set of golden triples, while S' denotes a set of corrupted triples. A triple (h, r, t) consists of two entities $h, t \in E$ (the set of entities) and relation $r \in R$ (the set of relations). We use the bold

letters \mathbf{h}, \mathbf{r} and \mathbf{t} to denote the corresponding embedding representations.

3.1 Flexible Translation: TransF

We first analyze the limitations of TransE, TransR and TransH models. Then we explain our TransF model in detail.

As mentioned before, TransE and TransR work well to irreflexive and one-to-one relations but they have problems with reflexive/one-to-many/many-to-one/many-to-many relations. The reason is that both TransE and TransR adopt the score function $\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{l_{1/2}}$. When triple (h, r, t) holds, the score function is low, meaning that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. More specifically, if with the ideal embedding using the function $\mathbf{h} + \mathbf{r} = \mathbf{t}$ when (h, r, t) holds, we can obtain: 1) if (h, r, t) and (t, r, h) are both correct, r is a reflexive relation and $\mathbf{r} = \mathbf{0}, \mathbf{h} = \mathbf{t}$; 2) if a set of triples $(h, r, t_i), \forall i \in 0, \dots, n$ hold, r is a one-to-many relation and $\mathbf{t}_0 = \dots = \mathbf{t}_n$; 3) if a set of triples $(h_i, r, t), \forall i \in 0, \dots, n$ hold, r is a many-to-one relation and $\mathbf{h}_0 = \dots = \mathbf{h}_n$. Although TransH can solve this problem, it introduces extra parameters for each relation and are not flexible and efficient enough to improve the TransR model.

To alleviate the problem of TransE and maintain the high efficiency, we apply $\mathbf{h} + \mathbf{r} \approx \alpha \mathbf{t}, \alpha > 0$, instead of $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, when (h, r, t) holds. That means we only need to maintain the directions of vectors $\mathbf{h} + \mathbf{r}$ and \mathbf{t} , but ignore their magnitudes. Therefore, 1) when r is a reflexive relation, we get $\mathbf{h} = \frac{2}{\alpha_1\alpha_2-1}\mathbf{r}, \mathbf{t} = \frac{\alpha_2+1}{\alpha_1\alpha_2-1}\mathbf{r}$, where $\mathbf{h} + \mathbf{r} \approx \alpha_1\mathbf{t}$, $\mathbf{t} + \mathbf{r} \approx \alpha_2\mathbf{h}$; 2) if r is a one-to-many relation, we get $\mathbf{t}_0 = \frac{\mathbf{h}+\mathbf{r}}{\alpha_0}, \dots, \mathbf{t}_n = \frac{\mathbf{h}+\mathbf{r}}{\alpha_n}$; 3) if r is a many-to-one relation, we get $\mathbf{h}_0 = \alpha_0\mathbf{t} - \mathbf{r}, \dots, \mathbf{h}_n = \alpha_n\mathbf{t} - \mathbf{r}$.

The score function is then defined as follows:

$$f_r(h, t) = (\mathbf{h} + \mathbf{r})^\top \mathbf{t} \quad (1)$$

However, with the score function, the constraints on head entity(h) and tail entity(t) are unbalanced. More specifically, under the constraints, the range of \mathbf{h} is a line, and the range of \mathbf{t} is a plane. Considering the perfect no-error embedding, we discuss the constraints on tail entity and head entity separately. As shown in Figure 2(a), when the embedding vectors \mathbf{h} and \mathbf{r} hold, the range of the no-error embedding vector \mathbf{t} is a vector with the right arrow on the dotted line. However, as shown in Figure 2(b), when the embedding vectors \mathbf{r} and \mathbf{t} are known, the range of the perfect embedding vector \mathbf{h} is a vector with the starting point on the dotted line a and the ending point on the dotted line b .

Since head entity and tail entity need to have the same effect during training, the constraints on them should be balanced. To this end, we design a Flexible Translation(TransF) model to address the unbalanced constraint problem. We modify the score function as follows:

$$f_r(h, t) = (\mathbf{h} + \mathbf{r})^\top \mathbf{t} + \mathbf{h}^\top (\mathbf{t} - \mathbf{r}) \quad (2)$$

The score is expected to be higher for a golden triple while lower for a corrupted one. As shown in Figure 3, both TransF and TransE are presented the score distribution on benchmark datasets. It is clear that our TransF model discriminate the golden triples and corrupted triples much better than TransE.

Table 1: The statistics of the corpora

Dataset	#R.	#Ent.	#Train	#Valid	#Test
WN18	18	40,943	141,442	5,000	5,000
FBK15K	1,345	14,951	483,142	50,000	59,071
WN11	11	38,696	112,581	2,609	10,544
FB13	13	75,043	316,232	5,908	23,733

That well illustrates the reason why our TransF model outperforms TransE. During the training process, we enforce the L_2 -norm of the embeddings \mathbf{h}_r and \mathbf{t} is smaller than 1.

3.2 Enhancement of TransR: TransRF

We can improve the TransE related models to TransF related ones, since TransF surpass TransE in the discriminating capability. Taking the TransR as an example, we enhance it to TransRF, a hybrid model which takes the advantages of both TransR and TransF, while keeping the model complexity comparable to TransR. Unlike TransR which takes $\mathbf{h}_r + \mathbf{r} \approx \mathbf{t}_r$ if (h, r, t) is a golden triple, TransRF adopts $\mathbf{h}_r + \mathbf{r} \approx \alpha \mathbf{t}_r$.

Accordingly, we can define the score function as

$$f_r(\mathbf{h}, \mathbf{t}) = (\mathbf{h}_r + \mathbf{r})^\top \mathbf{t}_r + \mathbf{h}_r^\top (\mathbf{t}_r - \mathbf{r}) \quad (3)$$

where $\mathbf{h}_r = \mathbf{h}\mathbf{M}_r$ and $\mathbf{t}_r = \mathbf{t}\mathbf{M}_r$. For each relation r , \mathbf{M}_r is the projection matrix which projects entities from the entity space to the relation space. We follow the constraints on the norms of the embeddings of h, r and t and the mapping matrices, $\forall h, r, t$, we have $\|\mathbf{h}\|_2 \leq 1$, $\|\mathbf{r}\|_2 \leq 1$, $\|\mathbf{t}\|_2 \leq 1$, $\|\mathbf{h}\mathbf{M}_r\|_2 \leq 1$ and $\|\mathbf{t}\mathbf{M}_r\|_2 \leq 1$.

3.3 Training Objective

All models are trained with contrastive max-margin objective functions. The objective is to ensure that a triple $(h, r, t) \in S$ in the golden set should have a higher score than a triple $(h', r, t') \in S'$ in the corrupted triple set, as follows:

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r,t') \in S'} \max(0, \gamma - f_r(\mathbf{h}, \mathbf{t}) + f_r(\mathbf{h}', \mathbf{t}')) \quad (4)$$

where $\gamma > 0$ is a margin hyperparameter. S is the training set of golden triples. S' is the set of corrupted triples. The corrupted triples is generated from the training triples with either the head or tail entity replaced by a random entity (but not both at the same time).

We adopt the mini-batched stochastic gradient descent (SGD) to optimize the objective function. The additional constraints on the norms of embedding parameters are explained with the model descriptions. For TransF, all embeddings are randomly initialized with a similar process of [Glorot and Bengio, 2010]. For TransRF, the initialization process is the same as TransR, and the projection matrices are initialized as identity matrix.

4 Experiments

In this section, we conduct extensive experiments to justify the proposed models. First, we evaluate our models on link prediction [Bordes *et al.*, 2013] and triple classification [Socher *et al.*, 2013] respectively. Second, to explain

the remarkable improvements, we assess how well the models can discriminate the golden triples and corrupted triples with the score function, comparing with TransE.

Three benchmark datasets are adopted in the experiments: WN18 [Bordes *et al.*, 2013] which is extracted from Wordnet [Miller, 1995]; and two dense subgraphs of Freebase [Bollacker *et al.*, 2008], FB15K [Bordes *et al.*, 2013] and FB13 [Socher *et al.*, 2013]. Table 1 shows the statistics of these data sets.

4.1 Link Prediction

As reported in [Bordes *et al.*, 2011; Bordes *et al.*, 2013], link prediction is to predict the missing h or t given (h, r) or (r, t) respectively. In this task, we conduct the evaluation by ranking the set of candidate entities in knowledge graph, instead of offering a best matching entity. This experiment is conducted on two datasets, WN18 and FB15K.

Evaluation protocol. Following the protocol in TransE [Bordes *et al.*, 2013], for each test triple (h, r, t) , we replace the head entity h by every entity in the knowledge graph, and rank these corrupted triples in descending order by the similarity score which is given by f_r . Similarly, we repeat this procedure by replacing the tail entity t . After collecting all these triples, we use two evaluation metrics: the mean rank of the correct entities (denotes as *Mean Rank*); the proportion of correct entity ranks within 10 (denotes as *Hits@10*). We expect lower *Mean Rank* and higher *Hits@10* for a better predictor. However, some corrupted triples should be considered as correct ones, since they actually exist in knowledge graph. Ranking such triples ahead of the original correct one should not be counted as an error. To eliminate such cases, we filter out those corrupted triples which appear either in the training, validation or test datasets. We term the former evaluation setting as "Raw" and the latter as "Filter".

Implementation. By sharing the same data sets, we directly refer to the baselines and experimental results reported in [Lin *et al.*, 2015]. We select learning rate λ for SGD among $\{0.0001, 0.001, 0.0035, 0.01\}$, the margin γ among $\{0.25, 0.45, 0.6, 0.9\}$, the embedding dimension k among $\{20, 50, 100, 200\}$ and batch size B among $\{120, 480, 960, 1440\}$. In training TransF, the best configurations are $\lambda = 0.01, \gamma = 0.9, k = 50, B = 960$ on WN18; $\lambda = 0.0035, \gamma = 0.45, k = 100, B = 960$ on FB15K. For experiments with TransRF, the optimal configurations are: $\lambda = 0.0001, \gamma = 0.25, k = 50, B = 960$ on WN18; $\lambda = 0.001, \gamma = 0.25, k = 50, B = 1440$ on FB15K. For all the datasets, we scan all the training triples for 1000 rounds for training TransF and 500 rounds for training TransRF.

Experiment Results. Table 2 lists the results on WN18 and FB15K. It demonstrates that TransF, which take relation categories into consideration, outperforms TransE and other baselines on WN18 and FB15K. Furthermore, our TransF model, which is more efficient with less parameters, obtains comparable results to TransH. Comparing the TransR to the TransRF model, TransRF is consistently outperforming TransR on WN18 and FB15K. The reason why our models fail on *Mean Rank* on WN18 may be due to the small number of relations in the WN18 dataset. For a fair comparison, the reason why we first compare the results of TransE,

Table 2: Experiment Result on link prediction

Data Sets	WN18				FB15K			
Metric	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
Unstructured [Bordes <i>et al.</i> , 2014]	315	304	35.3	38.2	1,074	979	4.5	6.3
RESCAL [Nickel <i>et al.</i> , 2011]	1,180	1,163	37.2	52.8	828	683	28.4	44.1
SE [Bordes <i>et al.</i> , 2011]	1,011	985	68.5	80.5	273	162	28.8	39.8
SME(linear) [Bordes <i>et al.</i> , 2014]	545	533	65.1	74.1	274	154	30.7	40.8
SME(bilinear) [Bordes <i>et al.</i> , 2014]	526	509	54.7	61.3	284	158	31.3	41.3
LFM [Jenatton <i>et al.</i> , 2012]	469	456	71.4	81.6	283	164	26.0	33.1
TransE [Bordes <i>et al.</i> , 2013]	263	251	75.4	89.2	243	125	34.9	47.1
TransH [Wang <i>et al.</i> , 2014]	318	303	75.4	86.7	211	84	42.5	58.5
TransF	403	390	76.0	90.2	220	89	40.5	61.2
TransR [Lin <i>et al.</i> , 2015]	232	219	78.3	91.7	226	78	43.8	65.5
TransRF	343	342	95.2	95.3	54	49	72.7	73.5

Table 3: Detailed results on FB15K in terms of different relation categories.(%)

Tasks	Predicting Head(Hits@10)				Predicting Tail(Hits@10)			
Relation Category	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
Unstructured [Bordes <i>et al.</i> , 2014]	34.5	2.5	6.1	6.6	34.3	4.2	1.9	6.6
SE [Bordes <i>et al.</i> , 2011]	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME(linear) [Bordes <i>et al.</i> , 2014]	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME(bilinear) [Bordes <i>et al.</i> , 2014]	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE [Bordes <i>et al.</i> , 2013]	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH [Wang <i>et al.</i> , 2014]	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransF	74.2	65.6	50.4	60.5	74.9	59.8	60.4	63.4
TransR [Lin <i>et al.</i> , 2015]	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
TransRF	74.3	66.8	50.5	76.6	75.4	55.9	67.1	79.0

TransH, and our TransF model then analyze the results between TransR and our proposed TransRF. is that TransR builds multiple relation spaces instead of one embedding space.

We dig into detailed prediction results with different types of relations including one-to-one/one-to-many/many-to-one/many-to-many relations, since TransF aims to handle these relation categories which TransE cannot. We classify the relations by following the same rules in [Bordes *et al.*, 2013]. Then, we obtain 24% one-to-one relations, 23% one-to-many relations, 29% many-to-one relations and 24% many-to-many relations within 1,345 relations on the FB15K dataset. As shown in Table 3, TransF model consistently outperforms the TransE model. Meanwhile, TransF outperforms TransH in one-to-one/many-to-one/many-to-many relations for predicting tail and one-to-one/one-to-many/many-to-many relations for predicting tail. TransRF also outperforms TransR in dealing with many-to-one/many-to-many relations for predicting head and one-to-many/many-to-many relations for predicting tail. Specifically, TransF and TransRF models bring promising improvements on many-to-many relations, and additionally, the performance on one-to-one is also significantly improved, compared with TransE and TransH. This may be due to the fact that our TransF model also relaxes the geometric assumption of TransE.

Model Complexity Analysis. To analyze the efficiency of our model, we compare the theoretical number of parameters of the baselines and record the running time of TransF, TransRF and other baseline models for each training epoch. As

Table 4: The running time of models for each training epoch(s) and model complexity(the number of parameters). n_e and n_r are the number of unique entities and relations, respectively. k is the dimension of embedding space.

Method	#Parameters	Running Time
TransE	$O(n_e k + n_r k)$	$\approx 0.6s$
TransH	$O(n_e k + 2n_r k)$	$\approx 14s$
TransR	$O(n_e k + n_r k^2 + n_r k)$	$\approx 128s$
TransF	$O(n_e k + n_r k)$	$\approx 0.6s$
TransRF	$O(n_e k + n_r k^2 + n_r k)$	$\approx 130s$

demonstrated in Table 4, our TransF model keeps the same number of parameters as the TransE model, and TransRF also keeps the same number of parameters as TransR. However, TransH needs more $n_r k$ parameters than TransF. Furthermore, the training time of TransH is nearly 24 times of that of TransE and TransF. The training time of TransRF is nearly the same as TransR. Therefore, our models are able to improve the knowledge graph embedding without sacrificing the efficiency.

4.2 Triple Classification

Following the experiment in [Bordes *et al.*, 2013; Wang *et al.*, 2014; Lin *et al.*, 2015], we also evaluate our model on triple classification. Triple classification is a binary classification task which predict whether a given triple (h, r, t) is correct or not. This task is applied for answering question such as *Does Michael Jackson publish the song Beat it?*. We use three data

Table 5: Comparison of accuracy on triple classification(%)

DataSets	WN11	FB13	FB15K
SE [Bordes <i>et al.</i> , 2011]	53.0	75.2	-
SME(bilinear) [Bordes <i>et al.</i> , 2014]	70.0	63.7	-
SLM [Socher <i>et al.</i> , 2013]	69.9	85.3	-
LFM [Jenatton <i>et al.</i> , 2012]	73.8	84.3	-
NTN [Socher <i>et al.</i> , 2013]	70.4	87.1	68.5
TransE [Bordes <i>et al.</i> , 2013]	75.9	70.9	79.6
TransH [Wang <i>et al.</i> , 2014]	77.7	76.5	79.0
TransF	86.4	82.1	90.5
TransR [Lin <i>et al.</i> , 2015]	85.5	74.7	81.7
TransRF	86.6	82.9	88.8

sets in this task: WN11 and FB13 released in NTN [Socher *et al.*, 2013]; FB15K used in TransR [Lin *et al.*, 2015].

Evaluation protocol. Following the protocol in NTN [Socher *et al.*, 2013], we set a relation-specific threshold T_r for prediction and then, for a triple (h, r, t) , if the similarity score obtained by f_r is above T_r , the triple (h, r, t) is predicted as positive, otherwise negative. The relation-specific threshold T_r is determined by maximizing the classification accuracy on a validation set.

Implementation. We compare our models with the baseline methods reported in [Lin *et al.*, 2015] for WN11, FB13 and FB15K. We enumerate learning rate λ for SGD among $\{0.0001, 0.001, 0.002, 0.01\}$, the margin γ among $\{0.1, 0.5, 1, 2, 2.25, 2.5\}$, the embedding dimension k among $\{20, 50, 100\}$ and the batch size B among $\{120, 480, 960, 1440\}$. For training TransF, the optimal configuration are: $\lambda = 0.01, \gamma = 2.25, k = 100, B = 960$ on WN11; $\lambda = 0.005, \gamma = 2, k = 100, B = 960$ on FB13; $\lambda = 0.002, \gamma = 0.5, k = 100, B = 960$ on FB15K. For experiments with TransRF, the best configurations are: $\lambda = 0.0001, \gamma = 2.5, k = 50, B = 960$ on WN11; $\lambda = 0.001, \gamma = 2.5, k = 50, B = 960$ on FB13; $\lambda = 0.001, \gamma = 0.1, d = 50, B = 120$ on FB15K. The number of training epochs is limited to 1,000 for TransF and 500 for TransRF.

Experiment Result. Evaluation results are reported in Tabel 5. It demonstrates that our TransF model outperforms all the baseline models significantly including TransE, TransH, and even TransR on WN11 and FB15K. On FB13, TransF beats all baseline models except the NTN model. As described in [Wang *et al.*, 2014; Lin *et al.*, 2015], FB13 is much denser than WN11 and FB15K where strong correlations exist between entities, and NTN can achieve better results by learning complicated correlations using tensor transformation from dense graph of FB13. Comparing TransRF and TransR, our TransRF model has higher accuracy than TransR on all datasets, especially on FB13 and FB15K.

4.3 Score Distribution

To explain the remarkable improvements obtained by our model on these tasks, we assess whether our model can better distinguish the positive and negative triples, compared with TransE. For embedding vectors, we directly use the embedding results which are trained for the link prediction task on WN18 and FB15K. Then, two triple sets of positive and neg-

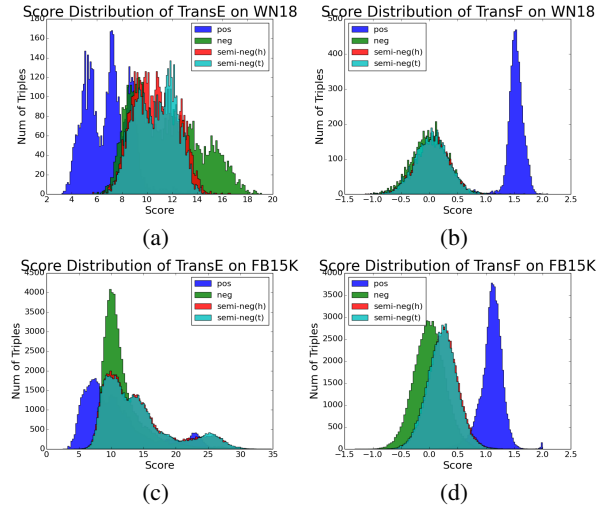


Figure 3: Score distributions of positive and negative triples counted by TransF and TransE on WN18 and FB15K datasets. TransF can better distinguish the positive and negative triples compared to TransE.

ative are built for comparison, where the positive triples are the ones in the test dataset. The negative triple set is constructed in three ways: replacing the head entity randomly from each of the positive triple; replacing the tail entity randomly; and randomly selecting two entities and a relation to assemble a new triple. The triples generated by the first two methods is called semi-negative triples and those by the third method negative ones. While generating negative triples, we guarantee the them do not exist in the knowledge graph.

As shown in Figure 3, the TransF model distinguish positive and negative triples remarkably better than TransE. In the TransF model, the boundary of scores for positive and negative triples is clear on both WN18 and FB15K datasets, as depicted in Figure 3(b)(d). In comparison, in the TransE model, we cannot find a clear boundary of scores on both WN18 and FB15K datasets, as can be seen from Figure 3(a)(c). For the TransF model, the score of positive triple is expected to be higher than that of negative triple. For the TransE model, the score of positive triple is expected to be lower than that of negative triple.

5 Conclusion

In this paper, we propose knowledge embedding models with flexible translation, which are effective, efficient and portable. The idea of flexible translation is to ensure that the sum vector of a head entity vector and a relation vector has the same direction with a tail entity vector but with flexible magnitude. The proposed models, TransF and TransRF, not only well address some existing issues in previous models when dealing with reflexive/one-to-many/many-to-one/many-to-many relations, but also maintain low complexity and high efficiency. We conduct extensive experiments on benchmark datasets for the tasks of link prediction and triple classification, and results show that our TransF and TransRF models obtain substantial improvements over baselines.

References

- [Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.
- [Bordes *et al.*, 2011] Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, et al. Learning structured embeddings of knowledge bases. In *AAAI*, 2011.
- [Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, 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. *Machine Learning*, 94(2):233–259, 2014.
- [Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feed-forward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.
- [Hoffmann *et al.*, 2011] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics, 2011.
- [Jenatton *et al.*, 2012] Rodolphe Jenatton, Nicolas L Roux, Antoine Bordes, and Guillaume R Obozinski. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems*, pages 3167–3175, 2012.
- [Lin *et al.*, 2015] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. 2015.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.
- [Miller, 1995] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
- [Mintz *et al.*, 2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.
- [Nickel *et al.*, 2011] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 809–816, 2011.
- [Nickel *et al.*, 2012] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, pages 271–280. ACM, 2012.
- [Riedel *et al.*, 2010] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2010.
- [Socher *et al.*, 2013] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, 2013.
- [Surdeanu *et al.*, 2012] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465. Association for Computational Linguistics, 2012.
- [Sutskever *et al.*, 2009] Ilya Sutskever, Joshua B Tenenbaum, and Ruslan Salakhutdinov. Modelling relational data using bayesian clustered tensor factorization. In *Advances in neural information processing systems*, pages 1821–1828, 2009.
- [Wang *et al.*, 2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119, 2014.
- [Weston *et al.*, 2013] Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973*, 2013.