Transition-based Knowledge Graph Embedding with Relational Mapping Properties

Miao Fan[†,*], **Qiang Zhou**[†], **Emily Chang**[‡], **Thomas Fang Zheng**[†,◇]
[†]CSLT, Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China.
[‡]Emory University, U.S.A.
[*]fanmiao.cslt.thu@gmail.com, [◇]fzheng@tsinghua.edu.cn

Abstract

Many knowledge repositories nowadays contain billions of triplets, i.e. (head-entity, relationship, tail-entity), as relation instances. These triplets form a directed graph with entities as nodes and relationships as edges. However, this kind of symbolic and discrete storage structure makes it difficult for us to exploit the knowledge to enhance other intelligence-acquired applications (e.g. the Question-Answering System), as many AI-related algorithms prefer conducting computation on continuous data. Therefore, a series of emerging approaches have been proposed to facilitate knowledge computing via encoding the knowledge graph into a low-dimensional embedding space. **TransE** is the latest and most promising approach among them, and can achieve a higher performance with fewer parameters by modeling the relationship as a *transitional vector* from the head entity to the tail entity. Unfortunately, it is not flexible enough to tackle well with the various mapping properties of triplets, even though its authors spot the harm on performance. In this paper, we thus propose a superior model called **TransM** to leverage the structure of the knowledge graph via pre-calculating the distinct weight for each training triplet according to its *relational mapping property*. In this way, the optimal function deals with each triplet depending on its own weight. We carry out extensive experiments to compare **TransM** with the state-of-the-art method **TransE** and other prior arts. The performance of each approach is evaluated within two different application scenarios on several benchmark datasets. Results show that the model we proposed significantly outperforms the former ones with lower parameter complexity as **TransE**.

1 Introduction

Many knowledge repositories have been constructed either by experts with long-term funding (e.g. WordNet[1] and OpenCyc[2]) or by crowds with collaborative contribution (e.g. Freebase[3] and DBpedia[4]). Most of them store billions of triplets. Each triplet, abbreviated as (*h, r, t*), is composed by two entities (i.e the head entity *h* and the tail entity *t*), and the relationship *r* between them. These triplets can form a huge directed graph for each knowledge repository with millions of entities as nodes and thousands of relationships as edges.

Ideally, we can take advantages of these knowledge graphs to enhance many other intelligence-dependent systems, such as Information Retrieval Systems (Wical, 1999; Wical, 2000), Question-Answering Systems (Pazzani and Engelman, 1983; Rinaldi et al., 2003; Hermjakob et al., 2000), etc. However, the graph-based knowledge representation is some kind of rigid. More specifically, this symbolic and discrete storage structure makes it hard for us to exploit great knowledge treasures, as many AI-related algorithms prefer conducting computations on continuous data. Some recent literatures on **Language Modeling** by means of learning *distributed word representation* (Bengio et al., 2003; Huang et al., 2012; Mikolov et al., 2013), have proved that embedding each word into a low-dimensional continuous vector could achieve better performance, because the global context information for each word can be better leveraged in this way. Therefore, in-

[1]http://www.princeton.edu/wordnet
[2]http://www.cyc.com/platform/opencyc
[3]http://www.freebase.com
[4]http://wiki.dbpedia.org

28th Pacific Asia Conference on Language, Information and Computation pages 328–337

spired by the idea of distributed representation, researchers have begun to explore approaches on embedding knowledge graphs and several canonical solutions (Bordes et al., 2011; Bordes et al., 2013b; Bordes et al., 2014a; Socher et al., 2013) have emerged recently to facilitate the knowledge computing via encoding both entities and relationships into low-dimensional continuous vectors which belong to the same embedding space.

Among prior arts, the latest **TransE** is a promising model which can achieve a higher performance than the other previously proposed approaches. Moreover, **TransE** is more efficient because the model holds fewer parameters to be decided, which makes it possible to deploy the algorithm on learning large-scale knowledge graph (e.g. Freebase[5]) embeddings. Unfortunately, it is not flexible enough to tackle well with the various relational mapping properties of triplets, even though Bordes et al. (2013b; 2013a) realize the harm on performance through splitting the dataset into different mapping-property categories, i.e. ONE-TO-ONE (*husband-to-wife*), MANY-TO-ONE (*children-to-father*), ONE-TO-MANY (*mother-to-children*), MANY-TO-MANY (*parents-to-children*). Bordes et al (2013b; 2013a) conduct experiments on each subset respectively. However, the result shows that **TransE** can only achieve less than 20% accuracy[6] when predicting the entities on the MANY-side, even though it can process ONE-TO-ONE triplets well. However, Bordes et al. (2013b) point out that there are roughly only 26.2% ONE-TO-ONE triplets. Therefore, the remainders, i.e. **73.8%** triplets with multi-mapping properties, are expected to be better processed.

In this paper, we propose a superior model named **TransM** which aims at leveraging the structure information of the knowledge graph. Precisely speaking, we keep the transition-based modeling for triplets proposed by **TransE** (Bordes et al., 2013b; Bordes et al., 2013a), i.e. $||\mathbf{h}+\mathbf{r}-\mathbf{t}||_{L_1/L_2}$. Meanwhile, our optimal function will give different respects for each training triplet via the pre-calculated weight corresponding to the relationship. Our intuition is that the *mapping property* of each triplet is decided by the relationship *r*, e.g. *husband-to-wife* is commonly known as ONE-TO-ONE relationship, while *parents-to-children* is naturally MANY-TO-MANY. Differing from **TransE**, **TransM** will concern more about the diverse contribution (i.e. various relational mapping properties) of each training triplet to the optimization target, i.e. minimizing the margin-based hinge loss function, so that the proposed model will be more flexible when dealing with heterogeneous mapping-properties of knowledge graphs.

We carry out extensive experiments in two different application scenarios, i.e. *link prediction* and *triplet classification*. For each task, we compare the proposed **TransM** with the state-of-the-art method **TransE** and other prior arts on several large-scale benchmark datasets. Results of both tasks demonstrate that our model significantly outperforms the others. Moreover, **TransM** has the comparable parameter complexity with **TransE**. we thus conclude that **TransM** is the most effective model so far while keeping the same efficiency with the state-of-the-art **TransE**.

2 Related Work

Almost all the related works take efforts on embedding each entity or relationship into a low-dimensional continuous space. To achieve this goal, each of them defines a distinct scoring function $f_r(h,t)$ to measure the compatibility of a given triplet (h,r,t).

Unstructured (Bordes et al., 2013b) is a naive model which just exploits the occurrence information of the head and the tail entities without considering the relationship between them. It defines a scoring function $||\mathbf{h}-\mathbf{t}||$, and obversely this model can not discriminate entity-pairs with different relationships. Therefore, **Unstructured** is commonly regarded as the baseline approach.

Distance Model (SE) (Bordes et al., 2011) uses a pair of matrix, i.e (W_{rh}, W_{rt}), to represent the relationship r. The dissimilarity[7] of a triplet (h,r,t) is calculate by the L_1 distance of $||W_{rh}\mathbf{h}-W_{rt}\mathbf{t}||$. Even though the model takes the relationships into

[5] So far, Freebase contains 1.9 billion triplets in total.

[6] Referring to the Table 4 in (Bordes et al., 2013b).

[7] Usually, $f_r(h,t)$ is a distance-measuring function and the lower dissimilarity means the higher compatibility of the triplet (h,r,t)

Model	Scoring Function	Parameter Complexity
Unstructured	$\|\|\mathbf{h}-\mathbf{t}\|\|$	$n_e d$
Distance Model (SE)	$\|\|W_{rh}\mathbf{h}-W_{rt}\mathbf{t}\|\|$; $(W_{rh}, W_{rt}) \in \mathbb{R}^{d\times d}$	$n_e d + 2n_r d^2$
Single Layer Model	$\mathbf{u}_r^T \tanh(W_{rh}\mathbf{h}+W_{rt}\mathbf{t}+\mathbf{b}_r)$; $(W_{rh}, W_{rt}) \in \mathbb{R}^{s\times d}$, $(\mathbf{u_r}, \mathbf{b}_r) \in \mathbb{R}^s$	$n_e d + 2n_r(sd+s)$
Bilinear Model	$\mathbf{h}^T W_r \mathbf{t}$; $W_r \in \mathbb{R}^{d\times d}$	$n_e d + n_r d^2$
Neural Tensor Network	$\mathbf{u}_r^T \tanh(\mathbf{h}^T W_r \mathbf{t} + W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$; $W_r \in \mathbb{R}^{d\times d\times s}$, $(W_{rh}, W_{rt}) \in \mathbb{R}^{s\times d}$, $(\mathbf{u_r}, \mathbf{b}_r) \in \mathbb{R}^s$	$n_e d + n_r(sd^2 + 2sd + 2s)$
TransE	$\|\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|\|$; $\mathbf{r} \in \mathbb{R}^d$	$n_e d + n_r d$
TransM	$w_r\|\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|\|$; $\mathbf{r} \in \mathbb{R}^d, w_r \in \mathbb{R}$	$n_e d + n_r d$ $(+n_r)$

Table 1: The scoring function and parameter complexity analysis for each related work. For all the models, we assume that there are a total of n_e entities, n_r relations (In most cases, $n_e \gg n_r$.), and each entity is embedded into a d-dimensional vector space, i.e $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$. We also suppose that there are s slices in a tensor for the neural-network related models, i.e *Single Layer Model* and *Neural Tensor Network*.

consideration, the separating matrices, i.e. W_{rh} and W_{rt}, as pointed out by Socher et al. (Socher et al., 2013), weaken the capable of capturing correlations between entities and relationships.

Single Layer Model proposed by Socher et al. (Socher et al., 2013) aims to alleviate the shortcomings of **Distance Model** by means of the nonlinearity of a standard, single layer neural network $g(W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$, where $g = tanh$. Then the linear output layer gives the score: $\mathbf{u}_r^T g(W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$.

Bilinear Model (Sutskever et al., 2009; Jenatton et al., 2012) is another model that tries to fix the issue of weak entity embedding vector interaction caused by **Distance Model (SE)** (Bordes et al., 2011) with the help of a relation-specific bilinear form: $f_r(h,t) = \mathbf{h}^T W_r \mathbf{t}$.

Neural Tensor Network (NTN) (Socher et al., 2013) mixes the **Single Layer Model** and the **Bilinear Model** and gives a general function: $f_r(h,t) = \mathbf{u}_r^T g(\mathbf{h}^T W_r \mathbf{t} + W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$, in which the second-order correlations are also considered into the nonlinear transformation function. This model is more expressive indeed, but the computation complexity is rather high.

TransE (Bordes et al., 2013b) is a simple but effective model which finds out that most of the relation instances in the knowledge graph are hierarchical and irreflexive (Bordes et al., 2013a). Therefore, Bordes et al. propose to embed relationship $\mathbf{r}$ as a transitional vector into the same continuous space with the entities, i.e. $\mathbf{h}$ and $\mathbf{t}$. They believe that if a triplet (h, r, t) does stand for a relation instance, then $\mathbf{h} + \mathbf{r} = \mathbf{t}$. Therefore, the scoring function of **TransE** is $||\mathbf{h} + \mathbf{r} - \mathbf{t}||$. Experiments show that **TransE** is state-of-the-art compared to the other related models. Moreover, its lower parameter complexity implies the capability of learning the embeddings for large-scale knowledge graphs. Therefore, it is a promising model which is both effective and efficient. This model works well on ONE-TO-ONE relation instances, as minimizing the global loss function will impose $\mathbf{h} + \mathbf{r}$ close to $\mathbf{t}$. However, the model will confuse about the other relation instances with multi-mapping properties, i.e MANY-TO-MANY, MANY-TO-ONE and ONE-TO-MANY, as entities locates on MANY-side will finally be trained extremely close to each other in the embedding space and also hard to be discriminated.

Therefore, we propose a superior model (**TransM**) in the next section, to give different roles to various training triplets based on their corresponding mapping properties while successively approaching the global optimal target.

Overall, Table 1 lists the scoring functions of all the works mentioned above. We furthermore analyse the parameter complexity of each prior mod-

el and conclude that **TransE** (Bordes et al., 2013b; Bordes et al., 2013a) is the most lightweight one so far.

3 TransM

In this section, we will narrate the intuition of our work at first, and then describe the proposed model **TransM** that formulates our idea. Finally, we give the detail algorithm about how to solve the proposed optimal model step by step.

3.1 Intuition

We agree with Bordes et al. (Bordes et al., 2013a; Bordes et al., 2013b) that most of the relation instances in the knowledge graph are hierarchical and irreflexive. Therefore, the relationship of each triplet (h, r, t) can be regarded as a directed transition $\mathbf{r}$ in the embedding space from the head entity $\mathbf{h}$ to the tail entity $\mathbf{t}$. Ideally, if all the correct triplets follow the assumption that every relation instance is strictly single-mapping (i.e. ONE-TO-ONE), $\mathbf{h} + \mathbf{r}$ will equal to $\mathbf{t}$ without conflicts.

In reality, however, there are roughly only 26.2% ONE-TO-ONE triplets that are suitable to be modeled by **TransE**. On the other hand, the remainder triplets (**73.8%**) suffer as illustrated on the left hand side of Figure 1, where the tail entities $(t_1, t_2, ..., t_m)$ are all pushed into a cramped range because minimizing loss function impels every training triplet to satisfy $||\mathbf{h} + \mathbf{r} - \mathbf{t}|| = 0$, leading to $h_1 = h_2 = ... = h_m$ in the worst case. Intuitively, we expect to lose the constrain and give more flexibility to the MANY-side as shown on the right side of Figure 1.

3.2 Model

A simple way to model our intuition is to associate each training triplet with a weight which represents the degree of mapping. According to our observation, the mapping property of a triplet depends much on its relationship. For example, *husband-to-wife* is a typical ONE-TO-ONE relationship in most cases, and *parents-to-children* is a MANY-TO-MANY relationship on the other hand. Therefore, the weights are relation-specific and the new scoring function we propose for a triplet (h, r, t) is,

$$f_r(h, t) = w_{\mathbf{r}}||\mathbf{h} + \mathbf{r} - \mathbf{t}||_{L_1/L_2} \quad (1)$$

For a correct triplet (h, r, t) in the training set Δ, we expect that the score of $f_r(h, t)$ is much lower than any corrupted triplet (h', r, t') that we randomly construct[8]. $\Delta'_{(h,r,t)}$ denotes the set of corrupted triplets for the correct one (h, r, t). Moreover, we use E (i.e. $(h, t) \in E$) and R (i.e. $r \in R$) to respectively denote the set of entities and relationships in the training set Δ.

To discriminate the correct and corrupted triplets, minimizing the margin-based hinge loss function is a simple but effective optimal model

$$\mathcal{L} = \min \sum_{(h,r,t)\in\Delta} \sum_{(h',r,t')\in\Delta'_{(h,r,t)}} [\gamma + f_r(h, t) - f_r(h', t')]_+$$
$$s.t. \qquad \forall e \in E, ||e||_2 = 1 \quad (2)$$

where $[\quad]_+$ is the hinge loss function, e.g. $[x]_+ = \max(x, 0)$, and γ is the margin. The reason that we constrain each entity located on the unit-ball is to guarantee that they can be updated in the same scale without being either wildly too large or small to satisfy the optimal target.

A simple way to measure the degree of mapping property for a relationship is to count the average number of tail entities per each distinct head entity and vice versa. We thus define hpt_r[9] (i.e. heads per tail) and tph_r[10] (i.e. tails per head) to jointly represent the mapping degree of relationship r. In this case, MANY-TO-MANY relation instances achieve much higher hpt and tph than ONE-TO-ONEs do. We would like to constrain ONE-TO-ONE instances more than MANY-TO-MANYs. Therefore, we design a formula to measure the weights as follows,

$$w_r = \frac{1}{\log(hpt_r + tph_r)} \quad (3)$$

The scoring function of **TransM** shown in Table 1 indicates that the parameter complexity of **TransM**

[8]The detail of constructing corrupted triplet is described in (Bordes et al., 2013b). Briefly speaking, the head or the tail entity (but not the both) of a gold triplet (h, r, t) is randomly replaced by other ones. In the meanwhile, we must make sure that the corrupted triplet (h', r, t') does not appear in the training set Δ.

[9]$hpt_r = \frac{\#(\Delta_r)}{\#(distinct(t_r))}$, where t_r represents the tail entities belonging to relationship r , and Δ_r denotes the training triplets containing the relationship r.

[10]$tph_r = \frac{\#(\Delta_r)}{\#(distinct(h_r))}$.

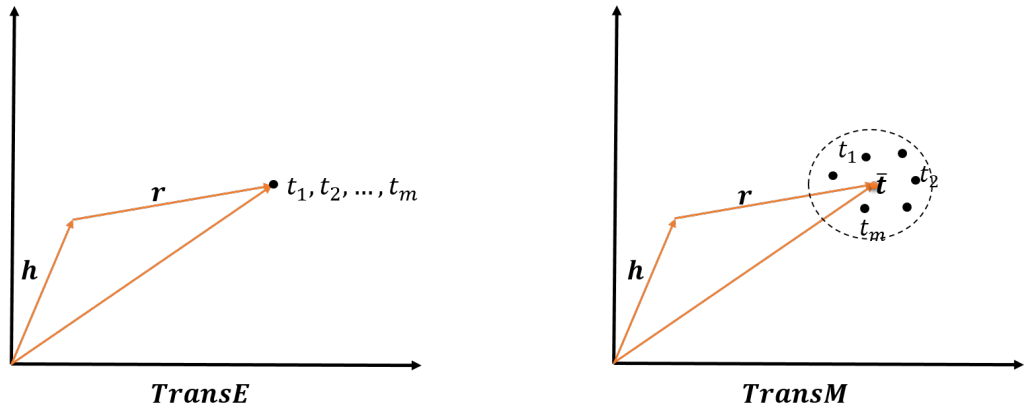

Figure 1: The differences between **TransE** and **TransM** when modeling ONE-TO-MANY relation instances, i.e. (h, r, t_1), (h, r, t_2), ..., (h, r, t_m).

is comparable with **TransE**, as the amount of entities is much larger than relationships in most cases. Moreover, as we can pre-compute the weight w_r for each relationship r, those parameters n_r can be ignored.

3.3 Algorithm

We use SGD (Stochastic Gradient Descent) to search the optimal solution in the iterative fashion. Algorithm 1 gives the pseudocodes that describe the procedure of learning **TransM**.

There are two key points we would like to clarify. First, we adopt projection method to pull back each updated entity to the uni-ball in order to satisfy the constraints in Equation (2). Second, we use the inner product ($f_r(h,t) = w_{\mathbf{r}}||\mathbf{h}+\mathbf{r}-\mathbf{t}||_2^2$) instead of L_2 norm ($f_r(h,t) = w_{\mathbf{r}}||\mathbf{h}+\mathbf{r}-\mathbf{t}||_2$) for facilitating the derivation of gradients.

4 Experiments

Embedding the knowledge into low-dimensional space makes it much easier to conduct further AI-related computing issues, such as *link prediction* (i.e. predicting t given h and r) and *triplet classification* (i.e. to discriminate whether a triplet (h, r, t) is correct or wrong). Two latest related works (Bordes et al., 2013b; Socher et al., 2013) evaluate their model on the subsets of WordNet (**WN**) and Freebase (**FB**) data, respectively. In order to conduct solid experiments, we compare our model with many related works including state-of-the-art and baseline

Algorithm 1 Learning **TransM**

Input:

Training set $\Delta = \{(h, r, t)\}$, entity set E, relation set R and weight set W;
Dimension of embeddings d, margin γ, step size s, convergence threshold ϵ, maximum epoches n.

foreach $\mathbf{r} \in R$ **do**
  $\mathbf{r} :=$ Uniform$(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}})$
  $\mathbf{r} :=$ Normalize$(\mathbf{r})$
end foreach

foreach $w_r \in W$ **do**
  Weighting$(\mathbf{r})$ according to Equation (3)
end foreach

$i := 0$
while $Rel.loss > \epsilon$ and $i < n$ **do**
  foreach $\mathbf{e} \in E$ **do**
    $\mathbf{e} :=$ Uniform$(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}})$
    $\mathbf{e} :=$ Normalize$(\mathbf{e})$
  end foreach

  foreach $(h, r, t) \in \Delta$ **do**
    $(h', r, t') :=$ Sampling$(\Delta'_{(h,r,t)})$
    if $\gamma + f_r(h,t) - f_r(h',t') \geq 0$ **then**
      Updating : $\bigtriangledown_{(h,r,t,h',t')}(\gamma + f_r(h,t) - f_r(h',t'))$
    end if
  end foreach
end while

DATASET	WN18	FB15K
#(ENTITIES)	40,943	14,951
#(RELATIONS)	18	1,345
#(TRAINING EX.)	141,442	483,142
#(VALIDATING EX.)	5,000	50,000
#(TESTING EX.)	5,000	59,071

Table 2: Statistics of the datasets used for link prediction task.

approaches in those two tasks. All the datasets, the source codes and the learnt embeddings for entities and relations can be downloaded from `http://1drv.ms/1nA2Vht`.

4.1 Link Prediction

One of the benefits of knowledge embedding is that we can apply simple mathematical operations to many reasoning tasks. For example, link prediction is a valuable task that contributes to completing the knowledge graph. Specifically, it aims at predicting the missing entity or the relationship given the other two elements in a fragmented triplet. For example, if we would like to tell whether the entity h has the relationship r with the entity t, we just need to calculate the distance between $\mathbf{h}+\mathbf{r}$ and $\mathbf{t}$. The closer they are, the more possibility the triplet (h, r, t) exists.

4.1.1 Benchmark Datasets

Bordes et al. (Bordes et al., 2013a; Bordes et al., 2013b) released two benchmark datasets[11] which are extracted from WordNet (**WN18**) and Freebase (**FB15K**), respectively. Table 2 shows the statistics of these two datasets. The size of **WN18** dataset is smaller than **FB15K**, with much fewer relationships but more entities.

4.1.2 Evaluation Protocol

For each testing triplet, the head entity is replaced by all the entities in the dictionary iteratively. The dissimilarity of each triplet candidate is firstly computed by the scoring functions, then sorted in ascending order, and finally the rank of the ground truth one is stored. This whole procedure is applied on the tail entity in the same way to gain the mean results. We use two metrics, i.e. *Mean Rank* and *Mean Hit@10* (i.e. the proportion of ground truth triplets that rank in Top-10), to measure the performance. However, those metrics are relatively raw, as the procedure above tends to bring in the false negative triplets, especially for multi-mapping relation instances. We thus filter out those triplets which appear in the training set and generate more reasonable results.

4.1.3 Experimental Results

We compare our model **TransM** with the state-of-the-art **TransE** and other models mentioned in (Bordes et al., 2013a) and (Bordes et al., 2014a) on the **WN18** and **FB15K**. We tune the parameters of each former model[12] based on the validation set and select the parameter combination which leads to the best performance. The results are almost the same as (Bordes et al., 2013b). We tried several parameter combinations, e.g. $d = \{20, 50, 100\}$, $\gamma = \{0.1, 1.0, 2.0, 10.0\}$ and $s = \{0.01, 0.1, 1.0\}$, for **TransM**, and finally select $d = 20$, $\gamma = 2.0$, $s = 0.01$ for **WN18** dataset; $d = 50$, $\gamma = 1.0$, $s = 0.01$ for **FB15K** dataset. Table 3 and Table 4 show the comparison between **TransM** and **TransE** on the performance of the two metrics when the scoring function is L_1 norm and L_2 norm. Results show that **TransM** outperforms **TransE** when we choose L_1 norm. These parameter combinations are also adopted by the *Triplet Classification* task to search other parameters, which we will describe in the next section. Moreover, Table 5 demonstrates that our model **TransM** outperforms the all the prior arts (i.e. the baseline model **Unstructured** (Bordes et al., 2014a), **RESCAL** (Nickel et al., 2011), **SE** (Bordes et al., 2011), **SME (LINEAR)** (Bordes et al., 2014a), **SME (BILINEAR)** (Bordes et al., 2014a), **LFM** (Jenatton et al., 2012) and the state-of-the-art **TransE** (Bordes et al., 2013a; Bordes et al., 2013b)) by evaluating them on the two benchmark datasets (i.e. **WN18** and **FB15K**).

Moreover, we divide **FB15K** into different categories (i.e. ONE-TO-ONE, ONE-TO-MANY, MANY-TO-ONE and MANY-TO-MANY) according to the mapping properties[13] of relationships, and

[11] The datasets can be downloaded from `https://www.hds.utc.fr/everest/doku.php?id=en:transe`

[12] All the codes for the related models can be downloaded from `https://github.com/glorotxa/SME`

[13] According to (Bordes et al., 2013b), we set 1.5 as the threshold to discriminate the single and the multi mapping prop-

DATASET	WN18							
NORM	L_1				L_2			
METRIC	*MEAN RANK*		*MEAN HIT@10*		*MEAN RANK*		*MEAN HIT@10*	
	Raw	*Filter*	*Raw*	*Filter*	*Raw*	*Filter*	*Raw*	*Filter*
TransE	294.4	283.2	70.38%	80.23%	**377.1**	**366.5**	38.56%	40.15%
TransM	**292.5**	**280.8**	**75.67%**	**85.38%**	440.4	429.4	**40.55%**	**42.43%**

Table 3: The detail results of link prediction between TransM and TransE on WN18 dataset when adopting L_1 and L_2 norm for the scoring function.

DATASET	FB15K							
NORM	L_1				L_2			
METRIC	*MEAN RANK*		*MEAN HIT@10*		*MEAN RANK*		*MEAN HIT@10*	
	Raw	*Filter*	*Raw*	*Filter*	*Raw*	*Filter*	*Raw*	*Filter*
TransE	243.3	139.9	36.86%	44.33%	254.6	146.3	37.26%	44.96%
TransM	**196.8**	**93.8**	**44.64%**	**55.15%**	**217.3**	**118.4**	**41.71%**	**50.40%**

Table 4: The detail results of link prediction between TransM and TransE on FB15K dataset when adopting L_1 and L_2 norm for the scoring function.

DATASET	WN18				FB15K			
METRIC	*MEAN RANK*		*MEAN HIT@10*		*MEAN RANK*		*MEAN HIT@10*	
	Raw	*Filter*	*Raw*	*Filter*	*Raw*	*Filter*	*Raw*	*Filter*
Unstructured	315	304	35.3%	38.2%	1,074	979	4.5%	6.3%
RESCAL	1,180	1,163	37.2%	52.8%	828	683	28.4%	44.1%
SE	1,011	985	68.5%	80.5%	273	162	28.8%	39.8%
SME(LINEAR)	545	533	65.1%	74.1%	274	154	30.7%	40.8%
SME(BILINEAR)	526	509	54.7%	61.3%	284	158	31.3%	41.3%
LFM	469	456	71.4%	81.6%	283	164	26.0%	33.1%
TransE	294.4	283.2	70.4%	80.2%	243.3	139.9	36.7%	44.3%
TransM	**292.5**	**280.8**	**75.7%**	**85.4%**	**196.8**	**93.8**	**44.6%**	**55.2%**

Table 5: Link prediction results. We compared our proposed TransM with the state-of-the-art method (TransE) and other prior arts.

TASK	*Predicting head*				*Predicting tail*			
REL. Mapping	1-TO-1	1-TO-M.	M.-TO-1	M.-TO-M.	1-TO-1	1-TO-M.	M.-TO-1	M.-TO-M.
Unstructured	34.5%	2.5%	6.1%	6.6%	34.3%	4.2%	1.9%	6.6%
SE	35.6%	62.6%	17.2%	37.5%	34.9%	14.6%	68.3%	41.3%
SME (LINEAR)	35.1%	53.7%	19.0%	40.3%	32.7%	14.9%	61.6%	43.3%
SME (BILINEAR)	30.9%	69.6%	19.9%	38.6%	28.2%	13.1%	76.0%	41.8%
TransE	59.7%	77.0%	14.7%	41.1%	58.5%	18.3%	80.2%	44.7%
TransM	**76.8%**	**86.3%**	**23.1%**	**52.3%**	**76.3%**	**29.0%**	**85.9%**	**56.7%**

Table 6: The detail results of *Filter Hit@10* (in %) on FB15K categorized by different mapping properties of relationship (M. stands for MANY).

analyse the performance of **Filter Hit@10** metric on each set. Table 6 shows that **TransM** outperforms on all categories, which proves that the proposed approach can not only maintain the characteristic of modeling the ONE-TO-ONE, but also better handle the multi-mapping relation instances.

4.2 Triplet Classification

Triplet classification is another task proposed by Socher et al. (Socher et al., 2013) which focuses on searching a relation-specific distance threshold σ_r to determine whether a triplet (h, r, t) is plausible.

4.2.1 Benchmark Datasets

Similar to Bordes et al. (Bordes et al., 2013a; Bordes et al., 2013b), Socher et al.(Socher et al., 2013) also constructed two standard datasets[14] (i.e. **WN11** and **FB13**) sampled from WordNet and Freebase. However, both of the benchmark datasets contain much fewer relationships. Therefore, we build another dataset obeying the principle proposed by Socher et al. (2013) based on **FB15K** which possesses much more relations. It is emphasized that the head or the tail entity can be randomly replaced with another one to produce a negative example, but in order to build much tough validation and testing datasets, we constrain that the picked entity should once appear at the same position. For example, *(Pablo Picaso, nationality, U.S.)* is a potential negative example rather than the obvious nonsense *(Pablo Picaso, nationality, Van Gogh)*, given a positive triplet *(Pablo Picaso, nationality, Spain)*. Table 7 shows the statistics of the standard datasets that we used for evaluating models on the triplet classification task.

4.2.2 Evaluation Protocol

The decision strategy for binary classification is simple: If the dissimilarity of a testing triplet (h, r, t) computed by $f_r(h, t)$ is below the relation-specific threshold σ_r, we predict it as positive, otherwise negative. The relation-specific threshold σ_r can be searched by maximizing the classification accuracy of the validation triplets which belongs to the relation r.

DATASET	WN11	FB13	FB15K
#(ENTITIES)	38,696	75,043	14,951
#(RELATIONS)	11	13	1,345
#(TRAINING EX.)	112,581	316,232	483,142
#(VALIDATING EX.)	5,218	11,816	100,000
#(TESTING EX.)	21,088	47,466	118,142

Table 7: Statistics of the datasets used for triplet classification task.

DATASET	WN11	FB13	FB15K
Distance Model	53.0%	75.2%	-
Hadamard Model	70.0%	63.7%	-
Single Layer Model	69.9%	85.3%	-
Bilinear Model	73.8%	84.3%	-
NTN	70.4%	**87.1%**	66.7%
TransE	77.5%	67.5%	85.8%
TransM	**77.8%**	72.1%	**89.9%**

Table 8: The accuracy of triplet classification compared with the state-of-the-art method (TransE) and other prior arts.

4.2.3 Experimental Results

We use the best parameter combination settings in the Link prediction task ($d = 20$, $\gamma = 2.0$, $s = 0.01$ for **WN11** dataset; $d = 50$, $\gamma = 1.0$, $s = 0.01$ for **FB13** and **FB15K** datasets.) to generate the entity and relation embeddings, and learn the best classification threshold σ_r for each relation r. Compared with the state-of-the-art, i.e. **TransE** (Bordes et al., 2013b; Bordes et al., 2013a) and other prior arts (i.e. **Distance Model** (Bordes et al., 2011), **Hadamard Model** (Bordes et al., 2012), **Single Layer Model** (Socher et al., 2013), **Bilinear Model** (Sutskever et al., 2009; Jenatton et al., 2012) and **Neural Tensor Network (NTN)**[15] (Socher et al., 2013)), our model **TransM** still achieves better performance as shown in Table 8.

Table 8 shows the best performance of **TransM** and **TransE** when selecting L_1 norm as the distance metric of the scoring functions. To display more de-

erties, i.e. for a triplet (h, r, t), if $h_r pt_r \leq 1.5$ and $t_r ph_r \leq 1.5$ in the meanwhile, we can categorize this triplet as ONE-TO-ONE relation instance.

[14]Those datasets can be download from the website `http://www.socher.org/index.php`

[15]Socher et al. reported higher classification accuracy in (Socher et al., 2013) with word embeddings. In order to conduct a fair comparison, the accuracy of **NTN** reported in Table 6 is same with the EV (entity vectors) results in Figure 4 of (Socher et al., 2013).

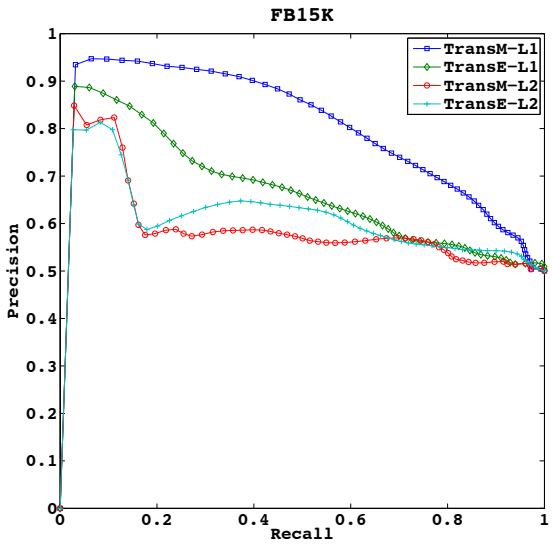

Figure 2: The Precision-Recall curves of TransE and TransM on the testing set of FB15K.

tails, we take the largest dataset as an example. We draw the Precision-Recall curves for all the positive testing triplets in the **FB15K** dataset while choosing L_1 and L_2 norm as the distance metric for the scoring functions of **TransM** and **TransE**. Figure 2 illustrates that the embeddings learned by **TransM** gain better capability of discriminating positive and negative triplets.

5 Conclusion and Future Work

TransM is a superior model that is not only expressive to represent the hierarchical and irreflexive characteristics but also flexible to adapt various mapping properties of the knowledge triplets. The results of extensive experiments on several benchmark datasets prove that our model can achieve higher performance without sacrificing efficiency. Moreover, we provide an insight that the relational mapping properties of a knowledge graph can be exploited to enhance the model.

Furthermore, we concern about two open questions in the following work:

- How to *learn* the specific weights for each triplet, so that the training examples can self-organize well with fewer conflict triplets.
- How to parallelize the algorithm without losing much performance, so that we can truly compute the world knowledge in the future.

In addition, we look forward to applying *Knowledge Graph Embedding* to reinforce some other related fields, such as *Relation Extraction* from free texts (Weston et al., 2013) and *Open Question Answering* (Bordes et al., 2014b).

6 Acknowledgments

This work is supported by National Program on Key Basic Research Project (973 Program) under Grant 2013CB329304, National Science Foundation of China (NSFC) under Grant No.61373075.

References

Yoshua Bengio, Rjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, et al. 2011. Learning structured embeddings of knowledge bases. In *AAAI*.

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

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013a. Irreflexive and hierarchical relations as translations. *arXiv preprint arXiv:1304.7158*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013b. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014a. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259.

Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014b. Open question answering with weakly supervised embedding models. *CoRR*, abs/1404.4326.

Ulf Hermjakob, Eduard H Hovy, and Chin-Yew Lin. 2000. Knowledge-based question answering. In *Proceedings of the Sixth World Multiconference on Systems, Cybernetics, and Informatics (SCI-2002)*.

Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. Improving word representations

via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea, July. Association for Computational Linguistics.

Rodolphe Jenatton, Nicolas Le Roux, Antoine Bordes, Guillaume Obozinski, et al. 2012. A latent factor model for highly multi-relational data. In *NIPS*, pages 3176–3184.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. 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.

Michael J Pazzani and Carl Engelman. 1983. Knowledge based question answering. In *Proceedings of the first conference on Applied natural language processing*, pages 73–80. Association for Computational Linguistics.

Fabio Rinaldi, James Dowdall, Michael Hess, Diego Mollá, Rolf Schwitter, and Kaarel Kaljurand. 2003. Knowledge-based question answering. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 785–792. Springer.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934.

Ilya Sutskever, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2009. Modelling relational data using bayesian clustered tensor factorization. In *NIPS*, pages 1821–1828.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1366–1371, Seattle, Washington, USA, October. Association for Computational Linguistics.

Kelly Wical. 1999. Information presentation in a knowledge base search and retrieval system, August 17. US Patent 5,940,821.

Kelly Wical. 2000. Concept knowledge base search and retrieval system, March 14. US Patent 6,038,560.