Knowledge Graph Embedding by Translating on Hyperplanes

Zhen Wang[1]*, **Jianwen Zhang**[2], **Jianlin Feng**[1], **Zheng Chen**[2]
[1]Department of Information Science and Technology, Sun Yat-sen University, Guangzhou, China
[2]Microsoft Research, Beijing, China
[1]{wangzh56@mail2, fengjlin@mail}.sysu.edu.cn
[2]{jiazhan, zhengc}@microsoft.com

Abstract

We deal with embedding a large scale knowledge graph composed of entities and relations into a continuous vector space. TransE is a promising method proposed recently, which is very efficient while achieving state-of-the-art predictive performance. We discuss some mapping properties of relations which should be considered in embedding, such as reflexive, one-to-many, many-to-one, and many-to-many. We note that TransE does not do well in dealing with these properties. Some complex models are capable of preserving these mapping properties but sacrifice efficiency in the process. To make a good trade-off between model capacity and efficiency, in this paper we propose TransH which models a relation as a hyperplane together with a translation operation on it. In this way, we can well preserve the above mapping properties of relations with almost the same model complexity of TransE. Additionally, as a practical knowledge graph is often far from completed, how to construct negative examples to reduce false negative labels in training is very important. Utilizing the one-to-many/many-to-one mapping property of a relation, we propose a simple trick to reduce the possibility of false negative labeling. We conduct extensive experiments on link prediction, triplet classification and fact extraction on benchmark datasets like WordNet and Freebase. Experiments show TransH delivers significant improvements over TransE on predictive accuracy with comparable capability to scale up.

Introduction

Knowledge graphs such as Freebase (Bollacker et al. 2008), WordNet (Miller 1995) and GeneOntology (Ashburner et al. 2000) have become very important resources to support many AI related applications, such as web/mobile search, Q&A, etc. A knowledge graph is a multi-relational graph composed of entities as nodes and relations as different types of edges. An instance of edge is a triplet of fact (*head entity, relation, tail entity*) (denoted as (h, r, t)). In the past decade, there have been great achievements in building large scale knowledge graphs, however, the general paradigm to support computing is still not clear. Two major difficulties are: (1) A knowledge graph is a symbolic and logical system while applications often involve numerical computing in continuous spaces; (2) It is difficult to aggregate global knowledge over a graph. The traditional method of reasoning by formal logic is neither tractable nor robust when dealing with long range reasoning over a real large scale knowledge graph. Recently a new approach has been proposed to deal with the problem, which attempts to embed a knowledge graph into a continuous vector space while preserving certain properties of the original graph (Socher et al. 2013; Bordes et al. 2013a; Weston et al. 2013; Bordes et al. 2011; 2013b; 2012; Chang, Yih, and Meek 2013). For example, each entity h (or t) is represented as a point $\mathbf{h}$ (or $\mathbf{t}$) in the vector space while each relation r is modeled as an operation in the space which is characterized by an a vector $\mathbf{r}$, such as translation, projection, etc. The representations of entities and relations are obtained by minimizing a global loss function involving all entities and relations. As a result, even the embedding representation of a single entity/relation encodes global information from the whole knowledge graph. Then the embedding representations can be used to serve all kinds of applications. A straightforward one is to complete missing edges in a knowledge graph. For any candidate triplet (h, r, t), we can confirm the correctness simply by checking the compatibility of the representations $\mathbf{h}$ and $\mathbf{t}$ under the operation characterized by $\mathbf{r}$.

Generally, knowledge graph embedding represents an entity as a k-dimensional vector $\mathbf{h}$ (or $\mathbf{t}$) and defines a scoring function $f_r(\mathbf{h}, \mathbf{t})$ to measure the plausibility of the triplet (h, r, t) in the embedding space. The score function implies a transformation $\mathbf{r}$ on the pair of entities which characterizes the relation r. For example, in translation based method (TransE) (Bordes et al. 2013b), $f_r(\mathbf{h}, \mathbf{t}) \triangleq \|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{\ell_{1/2}}$, i.e., relation r is characterized by the translating (vector) $\mathbf{r}$. With different scoring functions, the implied transformations vary between simple difference (Bordes et al. 2012), translation (Bordes et al. 2013b), affine (Chang, Yih, and Meek 2013), general linear (Bordes et al. 2011), bilinear (Jenatton et al. 2012; Sutskever, Tenenbaum, and Salakhutdinov 2009), and nonlinear transformations (Socher et al. 2013). Accordingly the model complexities (in terms of number of parameters) vary significantly. (Please refer to Table 1 and Section "Related Works" for details.)

Among previous methods, TransE (Bordes et al. 2013b) is a promising one as it is simple and efficient while achieving

*This work was done during Zhen Wang's internship in Microsoft Research.

state-of-the-art predictive performance. However, we find that there are flaws in TransE when dealing with relations with mapping properties of reflexive/one-to-many/many-to-one/many-to-many. Few previous work discuss the role of these mapping properties in embedding. Some advanced models with more free parameters are capable of preserving these mapping properties, however, the model complexity and running time is significantly increased accordingly. Moreover, the overall predictive performances of the advanced models are even worse than TransE (Bordes et al. 2013b). This motivates us to propose a method which makes a good trad-off between model complexity and efficiency so that it can overcome the flaws of TransE while inheriting the efficiency.

In this paper, we start by analyzing the problems of TransE on reflexive/one-to-many/many-to-one/many-to-many relations. Accordingly we propose a method named *translation on hyperplanes* (TransH) which interprets a relation as a translating operation on a hyperplane. In TransH, each relation is characterized by *two* vectors, the norm vector ($\mathbf{w}_r$) of the hyperplane, and the translation vector ($\mathbf{d}_r$) on the hyperplane. For a golden triplet (h, r, t), that it is correct in terms of worldly facts, the *projections* of $\mathbf{h}$ and $\mathbf{t}$ on the hyperplane are expected to be connected by the translation vector $\mathbf{d}_r$ with low error. This simple method overcomes the flaws of TransE in dealing with reflexive/one-to-many/many-to-one/many-to-many relations while keeping the model complexity almost the same as that of TransE. Regarding model training, we point out that carefully constructing negative labels is important in knowledge embedding. By utilizing the mapping properties of relations in turn, we propose a simple trick to reduce the chance of false negative labeling. We conduct extensive experiments on the tasks of link prediction, triplet classification and fact extraction on benchmark datasets like WordNet and Freebase, showing impressive improvements on different metrics of predictive accuracy. We also show that the running time of TransH is comparable to TransE.

Related Work

The most related work is briefly summarized in Table 1. All these methods embed entities into a vector space and enforce the embedding compatible under a scoring function. Different models differ in the definition of scoring functions $f_r(\mathbf{h}, \mathbf{r})$ which imply some transformations on $\mathbf{h}$ and $\mathbf{t}$.

TransE (Bordes et al. 2013b) represents a relation by a translation vector $\mathbf{r}$ so that the pair of embedded entities in a triplet (h, r, t) can be connected by $\mathbf{r}$ with low error. TransE is very efficient while achieving state-of-the-art predictive performance. However, it has flaws in dealing with reflexive/one-to-many/many-to-one/many-to-many relations.

Unstructured is a simplified case of TransE, which considers the graph as mono-relational and sets all translations $\mathbf{r} = 0$, i.e., the scoring function is $\|\mathbf{h} - \mathbf{t}\|$. It is used as a naive baseline in (Bordes et al. 2012; 2013b). Obviously it cannot distinguish different relations.

Distant Model (Bordes et al. 2011) introduces two independent projections to the entities in a relation. It represents a relation by a left matrix W_{rh} and a right matrix W_{rt}. Dissimilarity is measured by L_1 distance between $W_{rh}\mathbf{h}$ and $W_{rt}\mathbf{t}$. As pointed out by (Socher et al. 2013), this model is weak in capturing correlations between entities and relations as it uses two separate matrices.

Bilinear Model (Jenatton et al. 2012; Sutskever, Tenenbaum, and Salakhutdinov 2009) models second-order correlations between entity embeddings by a quadratic form: $\mathbf{h}^\top W_r \mathbf{t}$. Thus, each component of an entity interacts with each component of the other entity.

Single Layer Model (Socher et al. 2013) introduces nonlinear transformations by neural networks. It concatenates $\mathbf{h}$ and $\mathbf{t}$ as an input layer to a non-linear hidden layer then the linear output layer gives the resulting score: $\mathbf{u}_r^\top f(W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$. A similar structure is proposed in (Collobert and Weston 2008).

NTN (Socher et al. 2013) is the most expressive model so far. It extends the Single Layer Model by considering the second-order correlations into nonlinear transformation (neural networks). The score function is $\mathbf{u}_r^\top f(\mathbf{h}^\top \mathbf{W}_r \mathbf{t} + W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$. As analyzed by the authors, even when the tensor $\mathbf{W}_r$ degenerates to a matrix, it covers all the above models. However, the model complexity is much higher, making it difficult to handle large scale graphs.

Beyond these works directly targeting the same problem of embedding knowledge graphs, there are extensive related works in the wider area of multi-relational data modeling, matrix factorization, and recommendations. Please refer to the Introduction part of (Bordes et al. 2013b).

Embedding by Translating on Hyperplanes

We first describe common notations. h denotes a head entity, r denotes a relation and t denotes a tail entity. The bold letters $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$ denote the corresponding embedding representations. Δ denotes the set of golden triplets, and Δ' denotes the set of incorrect triplets. Hence we use $(h, r, t) \in \Delta$ to state "(h, r, t) is correct". E is the set of entities. R is the set of relations.

Relations' Mapping Properties in Embedding

As introduced in Introduction & Related Work (Table 1), TransE models a relation r as a translation vector $\mathbf{r} \in \mathbb{R}^k$ and assumes the error $\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{\ell_1/\ell_2}$ is low if (h, r, t) is a golden triplet. It applies well to irreflexive and one-to-one relations but has problems when dealing with reflexive or many-to-one/one-to-many/many-to-many relations.

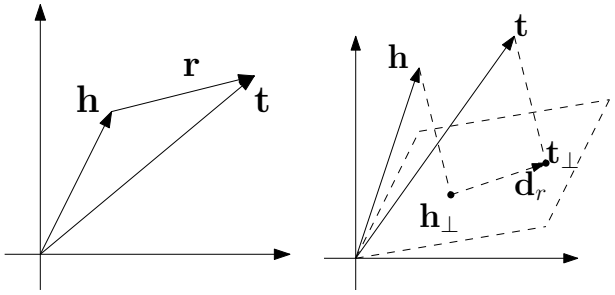
Considering the ideal case of no-error embedding where $\mathbf{h} + \mathbf{r} - \mathbf{t} = \mathbf{0}$ if $(h, r, t) \in \Delta$, we can get the following consequences directly from TransE model.

- If $(h, r, t) \in \Delta$ and $(t, r, h) \in \Delta$, i.e., r is a reflexive map, then $\mathbf{r} = \mathbf{0}$ and $\mathbf{h} = \mathbf{t}$.
- If $\forall i \in \{0, \ldots, m\}, (h_i, r, t) \in \Delta$, i.e., r is a many-to-one map, then $\mathbf{h}_0 = \ldots = \mathbf{h}_m$. Similarly, if $\forall i, (h, r, t_i) \in \Delta$, i.e., r is a one-to-many map, then $\mathbf{t}_0 = \ldots = \mathbf{t}_m$.

The reason leading to the above consequences is, in TransE, the representation of an entity is the same when involved in any relations, ignoring *distributed representations*

Table 1: Different embedding models: the scoring functions $f_r(\mathbf{h}, \mathbf{t})$ and the model complexity (the number of parameters). n_e and n_r are the number of unique entities and relations, respectively. It is the often case that $n_r \ll n_e$. k is the dimension of embedding space. s is the number of hidden nodes of a neural network or the number of slices of a tensor.

Model	Score function $f_r(\mathbf{h}, \mathbf{t})$	# Parameters
TransE (Bordes et al. 2013b)	$\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{\ell_{1/2}}$, $\mathbf{r} \in \mathbb{R}^k$	$O(n_e k + n_r k)$
Unstructured (Bordes et al. 2012)	$\|\mathbf{h} - \mathbf{t}\|_2^2$	$O(n_e k)$
Distant (Bordes et al. 2011)	$\|W_{rh}\mathbf{h} - W_{rt}\mathbf{t}\|_1$, $W_{rh}, W_{rt} \in \mathbb{R}^{k\times k}$	$O(n_e k + 2n_r k^2)$
Bilinear (Jenatton et al. 2012)	$\mathbf{h}^\top W_r \mathbf{t}, W_r \in \mathbb{R}^{k\times k}$	$O(n_e k + n_r k^2)$
Single Layer	$\mathbf{u}_r^\top f(W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$ $\mathbf{u}_r, \mathbf{b}_r \in \mathbb{R}^s, W_{rh}, W_{rt} \in \mathbb{R}^{s\times k}$	$O(n_e k + n_r(sk + s))$
NTN (Socher et al. 2013)	$\mathbf{u}_r^\top f(\mathbf{h}^\top \mathbf{W}_r \mathbf{t} + W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$ $\mathbf{u}_r, \mathbf{b}_r \in \mathbb{R}^s, \mathbf{W}_r \in \mathbb{R}^{k\times k\times s}, W_{rh}, W_{rt} \in \mathbb{R}^{s\times k}$	$O(n_e k + n_r(sk^2 + 2sk + 2s))$
TransH (this paper)	$\|(\mathbf{h} - \mathbf{w}_r^\top \mathbf{h}\mathbf{w}_r) + \mathbf{d}_r - (\mathbf{t} - \mathbf{w}_r^\top \mathbf{t}\mathbf{w}_r)\|_2^2$ $\mathbf{w}_r, \mathbf{d}_r \in \mathbb{R}^k$	$O(n_e k + 2n_r k)$



(a) TransE (b) TransH

Figure 1: Simple illustration of TransE and TransH.

of entities when involved in different relations. Although TransE does not enforce $\mathbf{h} + \mathbf{r} - \mathbf{t} = \mathbf{0}$ for golden triplets, it uses a ranking loss to encourage lower error for golden triplets and higher error for incorrect triplets (Bordes et al. 2013b), the tendency in the above propositions still exists.

Translating on Hyperplanes (TransH)

To overcome the problems of TransE in modeling reflexive/one-to-many/many-to-one/many-to-many relations, we propose a model which enables an entity to have distributed representations when involved in different relations. As illustrated in Figure 1, for a relation r, we position the relation-specific translation vector $\mathbf{d}_r$ in the relation-specific hyperplane $\mathbf{w}_r$ (the normal vector) rather than in the same space of entity embeddings. Specifically, for a triplet (h, r, t), the embedding $\mathbf{h}$ and $\mathbf{t}$ are first projected to the hyperplane $\mathbf{w}_r$. The projections are denoted as $\mathbf{h}_\perp$ and $\mathbf{t}_\perp$, respectively. We expect $\mathbf{h}_\perp$ and $\mathbf{t}_\perp$ can be connected by a translation vector $\mathbf{d}_r$ *on the hyperplane* with low error if (h, r, t) is a golden triplet. Thus we define a scoring function $\|\mathbf{h}_\perp + \mathbf{d}_r - \mathbf{t}_\perp\|_2^2$ to measure the plausibility that the triplet is incorrect. By restricting $\|\mathbf{w}_r\|_2 = 1$, it is easy to get

$$\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^\top \mathbf{h}\mathbf{w}_r, \quad \mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^\top \mathbf{t}\mathbf{w}_r.$$

Then the score function is

$$f_r(\mathbf{h}, \mathbf{t}) = \|(\mathbf{h} - \mathbf{w}_r^\top \mathbf{h}\mathbf{w}_r) + \mathbf{d}_r - (\mathbf{t} - \mathbf{w}_r^\top \mathbf{t}\mathbf{w}_r)\|_2^2.$$

The score is expected to be lower for a golden triplet and higher for an incorrect triplet. We name this model TransH. The model parameters are, all the entities' embeddings, $\{\mathbf{e}_i\}_{i=1}^{|E|}$, all the relations' hyperplanes and translation vectors, $\{(\mathbf{w}_r, \mathbf{d}_r)\}_{r=1}^{|R|}$.

In TransH, by introducing the mechanism of projecting to the relation-specific hyperplane, it enables different roles of an entity in different relations/triplets.

Training

To encourage discrimination between golden triplets and incorrect triplets, we use the following margin-based ranking loss:

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

where $[x]_+ \triangleq \max(0, x)$, Δ is the set of positive (golden) triplets, $\Delta'_{(h,r,t)}$ denotes the set of negative triplets constructed by corrupting (h, r, t), γ is the margin separating positive and negative triplets. The next subsection will introduce the details of constructing $\Delta'_{(h,r,t)}$.

The following constraints are considered when we minimize the loss $\mathcal{L}$:

$$\forall e \in E, \|\mathbf{e}\|_2 \le 1, \text{//scale} \quad (1)$$

$$\forall r \in R, |\mathbf{w}_r^\top \mathbf{d}_r| / \|\mathbf{d}_r\|_2 \le \epsilon, \text{//orthogonal} \quad (2)$$

$$\forall r \in R, \|\mathbf{w}_r\|_2 = 1, \text{//unit normal vector} \quad (3)$$

where the constraint (2) guarantees the translation vector $\mathbf{d}_r$ is in the hyperplane. Instead of directly optimizing the loss

function with constraints, we convert it to the following unconstrained loss by means of soft constraints:

$$\mathcal{L} = \sum_{(h,r,t)\in\Delta} \sum_{(h',r',t')\in\Delta'_{(h,r,t)}} \left[f_r(\mathbf{h},\mathbf{t}) + \gamma - f_{r'}(\mathbf{h}',\mathbf{t}')\right]_+ + C\left\{\sum_{e\in E}\left[\|\mathbf{e}\|_2^2 - 1\right]_+ + \sum_{r\in R}\left[\frac{(\mathbf{w}_r^\top \mathbf{d}_r)^2}{\|\mathbf{d}_r\|_2^2} - \epsilon^2\right]_+\right\}, \quad (4)$$

where C is a hyper-parameter weighting the importance of soft constraints.

We adopt stochastic gradient descent (SGD) to minimize the above loss function. The set of golden triplets (the triplets from the knowledge graph) are randomly traversed multiple times. When a golden triplet is visited, a negative triplet is randomly constructed (according to the next section). After a mini-batch, the gradient is computed and the model parameters are updated. Notice that the constraint (3) is missed in Eq. (4). Instead, to satisfy constraint (3), we project each $\mathbf{w}_r$ to unit ℓ_2-ball before visiting each mini-batch.

Reducing False Negative Labels

As described in the previous section, training involves constructing negative triplets for a golden triplet. Previous methods simply get negative triplets by randomly corrupting the golden triplet. For example, in TransE, for a golden triplet (h, r, t), a negative triplet (h', r, t') is obtained by randomly sampling a pair of entities (h', t') from E. However, as a real knowledge graph is often far from completed, this way of randomly sampling may introduce many false negative labels into training.

We adopt a different approach for TransH. Basically, we set different probabilities for replacing the head or tail entity when corrupting the triplet, which depends on the mapping property of the relation, i.e., one-to-many, many-to-one or many-to-many. We tend to give more chance to replacing the head entity if the relation is one-to-many and give more chance to replacing the tail entity if the relation is many-to-one. In this way, the chance of generating false negative labels is reduced. Specifically, among all the triplets of a relation r, we first get the following two statistics: (1) the average number of tail entities per head entity, denoted as tph; (2) the average number of head entities per tail entity, denoted as hpt. Then we define a Bernoulli distribution with parameter $\frac{tph}{tph+hpt}$ for sampling: given a golden triplet (h, r, t) of the relation r, with probability $\frac{tph}{tph+hpt}$ we corrupt the triplet by replacing the head, and with probability $\frac{hpt}{tph+hpt}$ we corrupt the triplet by replacing the tail.

Experiments

We empirically study and evaluate related methods on three tasks: link prediction (Bordes et al. 2013b), triplets classification (Socher et al. 2013), and relational fact extraction (Weston et al. 2013). All three tasks evaluate the accuracy of predicting unseen triplets, from different viewpoints and application context.

Table 2: Data sets used in the experiments.

Dataset	#R	#E	#Trip. (Train / Valid / Test)		
WN18	18	40,943	141,442	5,000	5,000
FB15k	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
FB5M	1,192	5,385,322	19,193,556	50,000	59,071

Link Prediction

Used in (Bordes et al. 2011; 2013b), this task is to complete a triplet (h, r, t) with h or t missing, i.e., predict t given (h, r) or predict h given (r, t). Rather than requiring one best answer, this task emphasizes more on ranking a set of candidate entities from the knowledge graph.

We use the same two data sets which are used in TransE (Bordes et al. 2011; 2013b): WN18, a subset of Wordnet; FB15k, a relatively dense subgraph of Freebase where all entities are present in Wikilinks database [1]. Both are released in (Bordes et al. 2013b). Please see Table 2 for more details.

Evaluation protocol. We follow the same protocol in TransE (Bordes et al. 2013b): For each testing triplet (h, r, t), we replace the tail t by every entity e in the knowledge graph and calculate a dissimilarity score (according to the scoring function f_r) on the corrupted triplet (h, r, e) . Ranking the scores in ascending order, we then get the rank of the original correct triplet. Similarly, we can get another rank for (h, r, t) by corrupting the head h. Aggregated over all the testing triplets, two metrics are reported: the *averaged rank* (denoted as *Mean*), and the *proportion of ranks not larger than 10* (denoted as *Hits@10*). This is called the "raw" setting. Notice that if a corrupted triplet exists in the knowledge graph, as it is also correct, ranking it before the original triplet is not wrong. To eliminate this factor, we remove those corrupted triplets which exist in either training, valid, or testing set before getting the rank of each testing triplet. This setting is called "filt". In both settings, a lower *Mean* is better while a higher *Hits@10* is better.

Implementation. As the data sets are the same, we directly copy experimental results of several baselines from (Bordes et al. 2013b). In training TransH, we use learning rate α for SGD among $\{0.001, 0.005, 0.01\}$, the margin γ among $\{0.25, 0.5, 1, 2\}$, the embedding dimension k among $\{50, 75, 100\}$, the weight C among $\{0.015625, 0.0625, 0.25, 1.0\}$, and batch size B among $\{20, 75, 300, 1200, 4800\}$. The optimal parameters are determined by the validation set. Regarding the strategy of constructing negative labels, we use "unif" to denote the traditional way of replacing head or tail with equal probability, and use "bern." to denote reducing false negative labels by replacing head or tail with different probabilities. Under the "unif" setting, the optimal configurations are: $\alpha = 0.01$, $\gamma = 1$, $k = 50$, $C = 0.25$, and $B = 75$ on WN18; $\alpha = 0.005$, $\gamma = 0.5$, $k = 50$, $C = 0.015625$, and $B = 1200$ on FB15k. Under "bern" setting, the optimal configurations

[1] http://code.google.com/p/wiki-links/

are: $\alpha = 0.01$, $\gamma = 1$, $k = 50$, $C = 0.25$, and $B = 1200$ on WN18; $\alpha = 0.005$, $\gamma = 0.25$, $k = 100$, $C = 1.0$, and $B = 4800$ on FB15k. For both datasets, we traverse all the training triplets for 500 rounds.

Results. The results are reported in Table 3. The simple models TransE, TransH, and even the naive baseline Unstructured (i.e., TransE without translation) outperform other approaches on WN18 in terms of the *Mean* metric. This may be because the number of relations in WN18 is quite small so that it is acceptable to ignore the different types of relations. On FB15k, TransH consistently outperforms the counterparts. We hypothesize that the improvements are due to the relaxed geometric assumption compared with TransE so that the reflexive/one-to-many/many-to-one/many-to-many relations can be better handled. To confirm the point, we dig into the detailed results of different mapping categories of relations, as reported in Table 4. Within the 1,345 relations, 24% are one-to-one, 23% are one-to-many, 29% are many-to-one, and 24% are many-to-many[2]. Overall, TransE is the runner up on FB15k. However, its relative superiorities on one-to-many and many-to-one relations are not as good as those on one-to-one relations. TransH brings promising improvements to TransE on one-to-many, many-to-one, and many-to-many relations. Outstripping our expectations, the performance on one-to-one is also significantly improved ($> 60\%$). This may be due to the "graph" property: *entities are connected with relations so that better embeddings of some parts lead to better results on the whole.* Table 5 reports the results of Hits@10 on some typical one-to-many/many-to-one/many-to-many/reflexive relations. The imrovement of TransH over TransE on these relations are very promising.

Table 5: Hits@10 of TransE and TransH on some examples of one-to-many*, many-to-one†, many-to-many‡, and reflexive§ relations.

Relation	Hits@10 (TransE / TransH) Left	Right
football_position/players*	100 / 100	16.7 / **22.2**
production_company/films*	65.6 / **85.6**	9.3 / **16.0**
director/film*	75.8 / **89.6**	50.5 / **80.2**
disease/treatments†	33.3 / **66.6**	100 / 100
person/place_of_birth†	30.0 / **37.5**	72.1 / **87.6**
film/production_companies†	11.3 / **21.0**	77.6 / **87.8**
field_of_study/students_majoring‡	24.5 / **66.0**	28.3 / **62.3**
award_winner/awards_won‡	40.2 / **87.5**	42.8 / **86.6**
sports_position/players‡	28.6 / **100**	64.3 / **86.2**
person/sibling_s§	21.1 / **63.2**	21.1 / **36.8**
person/spouse_s§	18.5 / **35.2**	18.5 / **42.6**

Triplets Classification

This task is to confirm whether a given triplet (h, r, t) is correct or not, i.e., binary classification on a triplet. It is used in (Socher et al. 2013) to evaluate NTN model.

Three data sets are used in this task. Two of them are the same as in NTN (Socher et al. 2013): WN11, a subset of WordNet; FB13, a subset of Freebase. As WN11 and FB13 contain very small number of relations, we also use the FB15k data set which contains much more relations. See Table 2 for details.

Evaluation protocol. We follow the same protocol in NTN (Socher et al. 2013). Evaluation of classification needs negative labels. The released sets of WN11 and FB13 already contain negative triplets which are constructed by (Socher et al. 2013), where each golden triplet is corrupted to get one negative triplet. For FB15k, we construct the negative triplets following the same procedure used for FB13 in (Socher et al. 2013).

The decision rule for classification is simple: for a triplet (h, r, t), if the dissimilarity score (by the score function f_r) is below a relation-specific threshold σ_r, then predict positive. Otherwise predict negative. The relation-specific threshold σ_r is determined according to (maximizing) the classification accuracy on the validation set.

Implementation. For WN11 and FB13, as we use the same data sets, directly copying the results of different methods from (Socher et al. 2013). For FB15k not used in (Socher et al. 2013), we implement TransE and TransH by ourselves, and use the released code for NTN.

For TransE, we search learning rate α in $\{0.001, 0.005, 0.01, 0.1\}$, margin γ in $\{1.0, 2.0\}$, embedding dimension k in $\{20, 50, 100\}$, and batch size B in $\{30, 120, 480, 1920\}$. We also apply the trick of reducing false negative labels to TransE. The optimal configurations of TransE (bern.) are: $\alpha = 0.01$, $k = 20$, $\gamma = 2.0$, $B = 120$, and L_1 as dissimilarity on WN11; $\alpha = 0.001$, $k = 100$, $\gamma = 2.0$, $B = 30$, and L_1 as dissimilarity on FB13; $\alpha = 0.005$, $k = 100$, $\gamma = 2.0$, $B = 480$, and L_1 as dissimilarity on FB15k. For TransH, the search space of hyperparameters is identical to link prediction. The optimal hyperparameters of TransH (bern.) are: $\alpha = 0.01$, $k = 100$, $\gamma = 2.0$, $C = 0.25$, and $B = 4800$ on WN11; $\alpha = 0.001$, $k = 100$, $\gamma = 0.25$, $C = 0.0625$, and $B = 4800$ on FB13; $\alpha = 0.01$, $k = 100$, $\gamma = 0.25$, $C = 0.0625$, and $B = 4800$ on FB15k. We didn't change the configuration of NTN code on FB113 where dimension $k = 100$, number of slices equals 3. Since FB15k is relatively large, we limit the number of epochs to 500.

Results. Accuracies are reported in Table 6. On WN11, TransH outperforms all the other methods. On FB13, the powerful model NTN is indeed the best one. However, on the larger set FB15k, TransE and TransH are much better than NTN. Notice that the number (1,345) of relations of FB15k is much larger than that (13) of FB13 while the number of entities are close (see Table 2). This means FB13 is a very dense subgraph where strong correlations exist between entities. In this case, modeling the complex correlations between entities by tensor and nonlinear

[2]For each relation r, we compute averaged number of tails per head (tph_r), averaged number of head per tail (hpt_r). If $tph_r < 1.5$ and $hpt_r < 1.5$, r is treated as one-to-one. If $tph_r \geq 1.5$ and $hpt_r \geq 1.5$, r is treated as a many-to-many. If $hpt_r < 1.5$ and $tph_r \geq 1.5$, r is treated as one-to-many. If $hpt_r \geq 1.5$ and $tph_r < 1.5$, r is treated as many-to-one.

Table 3: Link prediction results

Dataset	WN18				FB15k			
Metric	MEAN		HITS@10		MEAN		HITS@10	
	Raw	Filt.	Raw	Filt.	Raw	Filt.	Raw	Filt.
Unstructured (Bordes et al. 2012)	315	304	35.3	38.2	1,074	979	4.5	6.3
RESCAL (Nickel, Tresp, and Kriegel 2011)	1,180	1,163	37.2	52.8	828	683	28.4	44.1
SE (Bordes et al. 2011)	1,011	985	68.5	80.5	273	162	28.8	39.8
SME (Linear) (Bordes et al. 2012)	545	533	65.1	74.1	274	154	30.7	40.8
SME (Bilinear) (Bordes et al. 2012)	526	509	54.7	61.3	284	158	31.3	41.3
LFM (Jenatton et al. 2012)	469	456	71.4	81.6	283	164	26.0	33.1
TransE (Bordes et al. 2013b)	263	**251**	75.4	**89.2**	243	125	34.9	47.1
TransH (unif.)	318	**303**	75.4	**86.7**	211	**84**	42.5	**58.5**
TransH (bern.)	400.8	388	73.0	82.3	212	**87**	45.7	**64.4**

Table 4: Results on FB15k by relation category

Task	Predicting left (HITS@10)				Predicting right (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 et al. 2012)	34.5	2.5	6.1	6.6	34.3	4.2	1.9	6.6
SE (Bordes et al. 2011)	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME (Linear) (Bordes et al. 2012)	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME (Bilinear) (Bordes et al. 2012)	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE (Bordes et al. 2013b)	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH (unif.)	**66.7**	**81.7**	**30.2**	**57.4**	**63.7**	**30.1**	**83.2**	**60.8**
TransH (bern.)	**66.8**	**87.6**	**28.7**	**64.5**	**65.5**	**39.8**	**83.3**	**67.2**

transformation helps with embedding. However, on the much sparser subgraph of FB15k, it seems the simple assumption of translation or translation on hyperplanes is enough while the complex model of NTN is not necessary. Concerning running time, the cost of NTN is much higher than TransE/TransH. In addition, on all the three data sets, the trick of reducing false negative labeling (the results with "bern.") helps both TransE and TransH.

In NTN (Socher et al. 2013), the results of combining it with word embedding (Mikolov et al. 2013) are also reported. However, how best to combine word embedding is model dependent and also an open problem that goes beyond the scope of this paper. For a clear and fair comparison, all the results in Table 6 are without combination with word embedding.

Table 6: Triplet classification: accuracies (%). "40h", "5m" and "30m" in the brackets are the running (wall clock) time.

Dataset	WN11	FB13	FB15k
Distant 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**	66.5 ($\approx 40h$)
TransE (unif.)	75.85	70.9	79.7 ($\approx 5m$)
TransE (bern.)	75.87	81.5	**87.3** ($\approx 5m$)
TransH (unif.)	**77.68**	76.5	80.2 ($\approx 30m$)
TransH (bern.)	**78.80**	83.3	**87.7** ($\approx 30m$)

Relational Fact Extraction from Text

Extracting relational facts from text is an important channel for enriching a knowledge graph. Most existing extracting methods (Mintz et al. 2009; Riedel, Yao, and McCallum 2010; Hoffmann et al. 2011; Surdeanu et al. 2012) distantly collect evidences from an external text corpus for a candidate fact, ignoring the capability of the knowledge graph itself to reason the new fact. Actually, knowledge graph embedding is able to score a candidate fact, without observing any evidence from external text corpus. Recently (Weston et al. 2013) combined the *score from TransE* (evidence from knowledge graphs) with the *score from a text side extraction model* (evidence from text corpus) and observed promising improvement. In this experiment, we compare the contribution of TransH and TransE to improve relational fact extraction.

This experiment involves two major parts: text side extraction model and knowledge graph embedding.

For text side, we use the same data set in (Weston et al. 2013)—NYT+FB [3] released by (Riedel, Yao, and McCallum 2010). They aligned Freebase relations with the New York Times corpus by tagging entities in text using Stanford NER (Finkel, Grenager, and Manning 2005) and linking them to Freebase IDs through string matching on names. We only consider the most popular 50 predicates in the data set

[3] http://iesl.cs.umass.edu/riedel/data-univSchema/

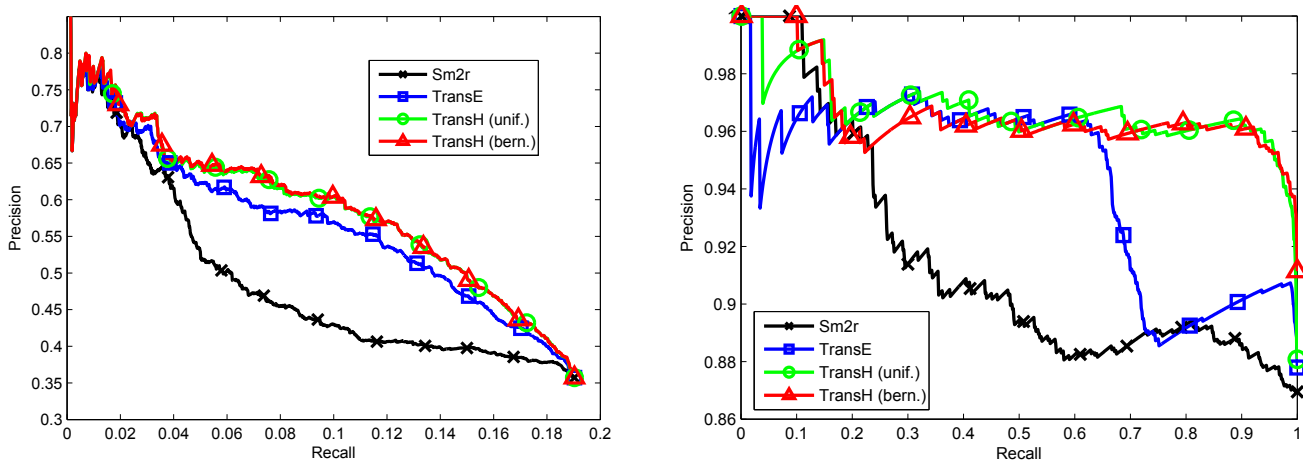


Figure 2: Precision-recall curves of TransE/TransH for fact extraction. (a) Combining the score from TransE/TransH and the score from Sm2r using the same rule in (Weston et al. 2013). (b) On the candidate facts accepted by Sm2r, we only use the score from TransE/TransH for prediction.

including the negative class—"NA". Then the data set is split into two parts: one for training, another for testing. As to the text side extraction method, both TransE and TransH can be used to provide prior scores for any text side methods. For a clear and fair comparison with TransE reported in (Weston et al. 2013), we implement the same text side method *Wsabie M2R* in (Weston et al. 2013), which is denoted as *Sm2r* in this paper.

For knowledge graph embedding, (Weston et al. 2013) used a subset of Freebase consisting of the most popular 4M entities and all the 23k Freebase relations. As they have not released the subset used in their experiment, we follow a similar procedure to produce a subset FB5M (Table 2) from Freebase. What is important is, we remove all the entity pairs that appear in the testing set from FB5M so that the generalization testing is not fake. We choose parameters for TransE/TransH without a comprehensive search due to the scale of FB5M. For simplicity, in both TransE and TransH, we set the embedding dimension k to be 50, the learning rate for SGD α to 0.01, the margin γ to 1.0, and dissimilarity of TransE to L_2.

Following the same rule of combining the score from knowledge graph embedding with the score from the text side model, we can obtain the precision-recall curves for TransE and TransH, as shown in Figure 2 (a). From the figure we can see TransH consistently outperforms TransE as a "prior" model on improving the text side extraction method Sm2r.

The results in Figure 2 (a) depend on the specific rule of combining the score from knowledge graph embedding with the score from text side model. Actually the combining rule in (Weston et al. 2013) is quite ad-hoc, which may not be the best way. Thus Figure 2 (a) does not clearly demonstrate the separate capability of TransE/TransH as a stand-alone model for relational fact prediction. To clearly demonstrate the stand-alone capability of TransE/TransH, we first use the text side model Sm2r to assign each entity pair to the relation with the highest confidence score, then keep those facts where the assigned relation is not "NA". On these accepted candidate facts, we only use the score of TransE/TransH to predict. The results are illustrated in Figure 2 (b). Both TransE and TransH perform better than the text side model Sm2r on this subset of candidates. TransH performs much better than TransE when recall is higher than 0.6.

Conclusion

In this paper, we have introduced TransH, a new model to embed a knowledge graph in a continuous vector space. TransH overcomes the flaws of TransE concerning the reflexive/one-to-many/many-to-one/many-to-many relations while inheriting its efficiency. Extensive experiments on the tasks of link prediction, triplet classification, and relational fact extraction show that TransH brings promising improvements to TransE. The trick of reducing false negative labels proposed in this paper is also proven to be effective.

References

Ashburner, M.; Ball, C. A.; Blake, J. A.; Botstein, D.; Butler, H.; Cherry, J. M.; Davis, A. P.; Dolinski, K.; Dwight, S. S.; Eppig, J. T.; et al. 2000. Gene ontology: Tool for the unification of biology. *Nature genetics* 25(1):25–29.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 1247–1250. ACM.

Bordes, A.; Weston, J.; Collobert, R.; and Bengio, Y. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*.

Bordes, A.; Glorot, X.; Weston, J.; and Bengio, Y. 2012. A semantic matching energy function for learning with multi-relational data. *Machine Learning* 1–27.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013a. Irreflexive and hierarchical relations as translations. *arXiv preprint arXiv:1304.7158*.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013b. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc. 2787–2795.

Chang, K.-W.; Yih, W.-t.; and Meek, C. 2013. Multi-relational latent semantic analysis. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1602–1612. Seattle, Washington, USA: Association for Computational Linguistics.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, 160–167. Omnipress.

Finkel, J. R.; Grenager, T.; and Manning, C. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 363–370. Association for Computational Linguistics.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L. S.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting on Association for Computational Linguistics*, 541–550. Association for Computational Linguistics.

Jenatton, R.; Roux, N. L.; Bordes, A.; and Obozinski, G. R. 2012. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc. 3167–3175.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc. 3111–3119.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Communications of the ACM* 38(11):39–41.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. 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*, 1003–1011. Association for Computational Linguistics.

Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, 809–816. New York, NY, USA: ACM.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 148–163.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc. 926–934.

Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Manning, C. D. 2012. 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*, 455–465. Association for Computational Linguistics.

Sutskever, I.; Tenenbaum, J. B.; and Salakhutdinov, R. 2009. Modelling relational data using bayesian clustered tensor factorization. In *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc. 1821–1828.

Weston, J.; Bordes, A.; Yakhnenko, O.; and Usunier, N. 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*, 1366–1371. Seattle, Washington, USA: Association for Computational Linguistics.