Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Learning Entity and Relation Embeddings for Knowledge Graph Completion

Yankai Lin[1], **Zhiyuan Liu**[1*], **Maosong Sun**[1,2], **Yang Liu**[3], **Xuan Zhu**[3]
[1] Department of Computer Science and Technology, State Key Lab on Intelligent Technology and Systems, National Lab for Information Science and Technology, Tsinghua University, Beijing, China
[2] Jiangsu Collaborative Innovation Center for Language Competence, Jiangsu, China
[3] Samsung R&D Institute of China, Beijing, China

Abstract

Knowledge graph completion aims to perform link prediction between entities. In this paper, we consider the approach of knowledge graph embeddings. Recently, models such as TransE and TransH build entity and relation embeddings by regarding a relation as translation from head entity to tail entity. We note that these models simply put both entities and relations within the same semantic space. In fact, an entity may have multiple aspects and various relations may focus on different aspects of entities, which makes a common space insufficient for modeling. In this paper, we propose TransR to build entity and relation embeddings in separate entity space and relation spaces. Afterwards, we learn embeddings by first projecting entities from entity space to corresponding relation space and then building translations between projected entities. In experiments, we evaluate our models on three tasks including link prediction, triple classification and relational fact extraction. Experimental results show significant and consistent improvements compared to state-of-the-art baselines including TransE and TransH. The source code of this paper can be obtained from https://github.com/mrlyk423/relation_extraction.

Introduction

Knowledge graphs encode structured information of entities and their rich relations. Although a typical knowledge graph may contain millions of entities and billions of relational facts, it is usually far from complete. Knowledge graph completion aims at predicting relations between entities under supervision of the existing knowledge graph. Knowledge graph completion can find new relational facts, which is an important supplement to relation extraction from plain texts.

Knowledge graph completion is similar to link prediction in social network analysis, but more challenging for the following reasons: (1) nodes in knowledge graphs are entities with different types and attributes; and (2) edges in knowledge graphs are relations of different types. For knowledge graph completion, we not only determine whether there is a relation between two entities or not, but also predict the specific type of the relation.

For this reason, traditional approach of link prediction is not capable for knowledge graph completion. Recently, a promising approach for the task is embedding a knowledge graph into a continuous vector space while preserving certain information of the graph. Following this approach, many methods have been explored, which will be introduced in detail in Section "Related Work".

Among these methods, TransE (Bordes et al. 2013) and TransH (Wang et al. 2014) are simple and effective, achieving the state-of-the-art prediction performance. TransE, inspired by (Mikolov et al. 2013b), learns vector embeddings for both entities and relationships. These vector embeddings are set in $\mathbb{R}^k$ and we denote with the same letters in boldface. The basic idea behind TransE is that, the relationship between two entities corresponds to a translation between the embeddings of entities, that is, $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when (h, r, t) holds. Since TransE has issues when modeling 1-to-N, N-to-1 and N-to-N relations, TransH is proposed to enable an entity having different representations when involved in various relations.

Both TransE and TransH assume embeddings of entities and relations being in the same space $\mathbb{R}^k$. However, an entity may have multiple aspects, and various relations focus on different aspects of entities. Hence, it is intuitive that some entities are similar and thus close to each other in the entity space, but are comparably different in some specific aspects and thus far away from each other in the corresponding relation spaces. To address this issue, we propose a new method, which models entities and relations in distinct spaces, i.e., **entity space** and multiple **relation spaces** (i.e., relation-specific entity spaces), and performs translation in the corresponding relation space, hence named as TransR.

The basic idea of TransR is illustrated in Fig. 1. For each triple (h, r, t), entities in the entity space are first projected into r-relation space as h_r and t_r with operation M_r, and then $\mathbf{h}_r + \mathbf{r} \approx \mathbf{t}_r$. The relation-specific projection can make the head/tail entities that actually hold the relation (denoted as colored circles) close with each other, and also get far away from those that do not hold the relation (denoted as colored triangles).

Moreover, under a specific relation, head-tail entity pairs usually exhibit diverse patterns. It is insufficient to build

*Corresponding author: Zhiyuan Liu (liuzy@tsinghua.edu.cn).

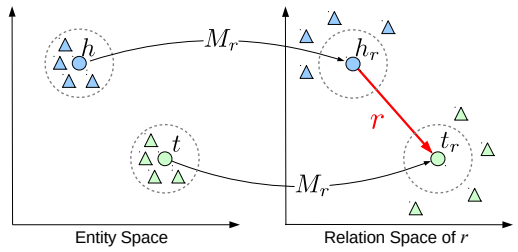

Figure 1: Simple illustration of TransR.

only a single relation vector to perform all translations from head to tail entities. For example, the head-tail entities of the relation "location_location_contains" have many patterns such as country-city, country-university, continent-country and so on. Following the idea of piecewise linear regression (Ritzema and others 1994), we extend TransR by clustering diverse head-tail entity pairs into groups and learning distinct relation vectors for each group, named as cluster-based TransR (CTransR).

We evaluate our models with the tasks of link prediction, triple classification and relation fact extraction on benchmark datasets of WordNet and Freebase. Experiment results show significant and consistent improvements compared to state-of-the-art models.

Related Models

TransE and TransH

As mentioned in Section "Introduction", TransE (Bordes et al. 2013) wants $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$ when (h, r, t) holds. This indicates that $(\mathbf{t})$ should be the nearest neighbor of $(\mathbf{h}+\mathbf{r})$. Hence, TransE assumes the score function

$$f_r(h,t) = \|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_2^2 \tag{1}$$

is low if (h, r, t) holds, and high otherwise.

TransE applies well to 1-to-1 relations but has issues for N-to-1, 1-to-N and N-to-N relations. Take a 1-to-N relation r for example. $\forall i \in \{0, \ldots, m\}, (h_i, r, t) \in S$. This indicates that $\mathbf{h}_0 = \ldots = \mathbf{h}_m$, which does not comport with the facts.

To address the issue of TransE when modeling N-to-1, 1-to-N and N-to-N relations, TransH (Wang et al. 2014) is proposed to enable an entity to have distinct distributed representations when involved in different relations. For a relation r, TransH models the relation as a vector $\mathbf{r}$ on a hyperplane with $\mathbf{w}_r$ as the normal vector. For a triple (h, r, t), the entity embeddings $\mathbf{h}$ and $\mathbf{t}$ are first projected to the hyperplane of $\mathbf{w}_r$, denoted as $\mathbf{h}_\perp$ and $\mathbf{t}_\perp$. Then the score function is defined as

$$f_r(h,t) = \|\mathbf{h}_\perp+\mathbf{r}-\mathbf{t}_\perp\|_2^2. \tag{2}$$

If we restrict $\|\mathbf{w}_r\|_2 = 1$, we will have $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^\top \mathbf{h}\mathbf{w}_r$ and $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^\top \mathbf{t}\mathbf{w}_r$. By projecting entity embeddings into relation hyperplanes, it allows entities playing different roles in different relations.

Other Models

Besides TransE and TransH, there are also many other methods following the approaches of knowledge graph embedding. Here we introduce several typical models, which will also be compared as baselines with our models in experiments.

Unstructured Model (UM). UM model (Bordes et al. 2012; 2014) was proposed as a naive version of TransE by assigning all $\mathbf{r} = \mathbf{0}$, leading to score function $f_r(h,t) = \|\mathbf{h}-\mathbf{t}\|_2^2$. This model cannot consider differences of relations.

Structured Embedding (SE). SE model (Bordes et al. 2011) designs two relation-specific matrices for head and tail entities, i.e., $\mathbf{M}_{r,1}$ and $\mathbf{M}_{r,2}$, and defines the score function as an L_1 distance between two projected vectors, i.e., $f_r(h,t) = \|\mathbf{M}_{r,1}\mathbf{h} - \mathbf{M}_{r,2}\mathbf{t}\|_1$. Since the model has two separate matrices for optimization, it cannot capture precise relations between entities and relations.

Single Layer Model (SLM). SLM model was proposed as a naive baseline of NTN (Socher et al. 2013). The score function of SLM model is defined as

$$f_r(h,t) = \mathbf{u}_r^\top g(\mathbf{M}_{r,1}\mathbf{h} + \mathbf{M}_{r,2}\mathbf{t}), \tag{3}$$

where $\mathbf{M}_{r,1}$ and $\mathbf{M}_{r,2}$ are weight matrices, and $g()$ is the `tanh` operation. SLM is a special case of NTN when the tensor in NTN is set to $\mathbf{0}$.

Semantic Matching Energy (SME). SME model (Bordes et al. 2012; 2014) aims to capture correlations between entities and relations via multiple matrix products and Hadamard product. SME model simply represents each relation using a single vector, which interacts with entity vectors via linear matrix products, with all relations share the same parameters. SME considers two definitions of semantic matching energy functions for optimization, including the linear form

$$f_r(h,t) = (\mathbf{M}_1\mathbf{h} + \mathbf{M}_2\mathbf{r} + \mathbf{b}_1)^\top (\mathbf{M}_3\mathbf{t} + \mathbf{M}_4\mathbf{r} + \mathbf{b}_2), \tag{4}$$

and the bilinear form

$$f_r(h,t) = \big((\mathbf{M}_1\mathbf{h}) \otimes (\mathbf{M}_2\mathbf{r}) + \mathbf{b}_1\big)^\top \big((\mathbf{M}_3\mathbf{t}) \otimes (\mathbf{M}_4\mathbf{r}) + \mathbf{b}_2\big), \tag{5}$$

where $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_3$ and $\mathbf{M}_4$ are weight matrices, $\otimes$ is the Hadamard product, $\mathbf{b}_1$ and $\mathbf{b}_2$ are bias vectors. In (Bordes et al. 2014), the bilinear form of SME is re-defined with 3-way tensors instead of matrices.

Latent Factor Model (LFM). LFM model (Jenatton et al. 2012; Sutskever, Tenenbaum, and Salakhutdinov 2009) considers second-order correlations between entity embeddings using a quadratic form, and defines a bilinear score function $f_r(h,t) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t}$.

Neural Tensor Network (NTN). NTN model (Socher et al. 2013) defines an expressive score function for graph embedding as follows,

$$f_r(h,t) = \mathbf{u}_r^\top g(\mathbf{h}^\top \mathbf{M}_r \mathbf{t} + \mathbf{M}_{r,1}\mathbf{h} + \mathbf{M}_{r,2}\mathbf{t} + \mathbf{b}_r), \tag{6}$$

where $\mathbf{u}_r$ is a relation-specific linear layer, $g()$ is the `tanh` operation, $\mathbf{M}_r \in \mathbb{R}^{d \times d \times k}$ is a 3-way tensor, and $\mathbf{M}_{r,1}, \mathbf{M}_{r,2} \in \mathbb{R}^{k \times d}$ are weight matrices. Meanwhile, the

corresponding high complexity of NTN may prevent it from efficiently applying on large-scale knowledge graphs.

In experiments we will also compare with **RESCAL**, a collective matrix factorization model presented in (Nickel, Tresp, and Kriegel 2011; 2012).

Our Method

To address the representation issue of TransE and TransH, we propose TransR, which represent entities and relations in distinct semantic space bridged by relation-specific matrices.

TransR

Both TransE and TransH assume embeddings of entities and relations within the same space $\mathbb{R}^k$. But relations and entities are completely different objects, it may be not capable to represent them in a common semantic space. Although TransH extends modeling flexibility by employing relation hyperplanes, it does not perfectly break the restrict of this assumption. To address this issue, we propose a new method, which models entities and relations in distinct spaces, i.e., **entity space** and **relation spaces**, and performs translation in relation space, hence named as TransR.

In TransR, for each triple (h, r, t), entities embeddings are set as $\mathbf{h}, \mathbf{t} \in \mathbb{R}^k$ and relation embedding is set as $\mathbf{r} \in \mathbb{R}^d$. Note that, the dimensions of entity embeddings and relation embeddings are not necessarily identical, i.e., $k \neq d$.

For each relation r, we set a projection matrix $\mathbf{M}_r \in \mathbb{R}^{k \times d}$, which may projects entities from entity space to relation space. With the mapping matrix, we define the projected vectors of entities as

$$\mathbf{h}_r = \mathbf{h}\mathbf{M}_r, \quad \mathbf{t}_r = \mathbf{t}\mathbf{M}_r. \tag{7}$$

The score function is correspondingly defined as

$$f_r(h, t) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_2^2. \tag{8}$$

In practice, we enforce constraints on the norms of the embeddings h, r, t and the mapping matrices, i.e. $\forall h, r, t$, we have $\|\mathbf{h}\|_2 \leq 1, \|\mathbf{r}\|_2 \leq 1, \|\mathbf{t}\|_2 \leq 1, \|\mathbf{h}\mathbf{M}_\mathbf{r}\|_2 \leq 1, \|\mathbf{t}\mathbf{M}_\mathbf{r}\|_2 \leq 1$.

Cluster-based TransR (CTransR)

The above mentioned models, including TransE, TransH and TransR, learn a unique vector for each relation, which may be under-representative to fit all entity pairs under this relation, because these relations are usually rather diverse. In order to better model these relations, we incorporate the idea of piecewise linear regression (Ritzema and others 1994) to extend TransR.

The basic idea is that, we first segment input instances into several groups. Formally, for a specific relation r, all entity pairs (h, t) in the training data are clustered into multiple groups, and entity pairs in each group are expected to exhibit similar r relation. All entity pairs (h, t) are represented with their vector offsets $(\mathbf{h} - \mathbf{t})$ for clustering, where $\mathbf{h}$ and $\mathbf{t}$ are obtained with TransE. Afterwards, we learn a separate relation vector $\mathbf{r}_c$ for each cluster and matrix $\mathbf{M}_r$ for each relation, respectively. We define the projected vectors of entities as $\mathbf{h}_{r,c} = \mathbf{h}\mathbf{M}_r$ and $\mathbf{t}_{r,c} = \mathbf{t}\mathbf{M}_r$, and the score function is defined as

$$f_r(h, t) = \|\mathbf{h}_{r,c} + \mathbf{r}_c - \mathbf{t}_{r,c}\|_2^2 + \alpha\|\mathbf{r}_c - \mathbf{r}\|_2^2, \tag{9}$$

where $\|\mathbf{r}_c - \mathbf{r}\|_2^2$ aims to ensure cluster-specific relation vector $\mathbf{r}_c$ not too far away from the original relation vector $\mathbf{r}$, and α controls the effect of this constraint. Besides, same to TransR, CTransR also enforce constraints on norm of embeddings h, r, t and mapping matrices.

Training Method and Implementation Details

We define the following margin-based score function as objective for training

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r,t') \in S'} \max\big(0, f_r(h, t) + \gamma - f_r(h', t')\big), \tag{10}$$

where $\max(x, y)$ aims to get the maximum between x and y, γ is the margin, S is the set of correct triples and S' is the set of incorrect triples.

Existing knowledge graphs only contain correct triples. It is routine to corrupt correct triples $(h, r, t) \in S$ by replacing entities, and construct incorrect triples $(h', r, t') \in S'$. When corrupting the triple, we follow (Wang et al. 2014) and assign different probabilities for head/tail entity replacement. For those 1-to-N, N-to-1 and N-to-N relations, by giving more chance to replace the "one" side, the chance of generating false-negative instances will be reduced. In experiments, we will denote the traditional sampling method as "unif" and the new method in (Wang et al. 2014) as "bern".

The learning process of TransR and CTransR is carried out using stochastic gradient descent (SGD). To avoid overfitting, we initialize entity and relation embeddings with results of TransE, and initialize relation matrices as identity matrices.

Experiments and Analysis

Data Sets and Experiment Setting

In this paper, we evaluate our methods with two typical knowledge graphs, built with WordNet (Miller 1995) and Freebase (Bollacker et al. 2008). WordNet provides semantic knowledge of words. In WordNet, each entity is a *synset* consisting of several words, corresponding to a distinct *word sense*. Relationships are defined between synsets indicating their lexical relations, such as `hypernym`, `hyponym`, `meronym` and `holonym`. In this paper, we employ two data sets from WordNet, i.e., `WN18` used in (Bordes et al. 2014) and `WN11` used in (Socher et al. 2013). `WN18` contains 18 relation types and `WN11` contains 11. Freebase provides general facts of the world. For example, the triple (Steve Jobs, founded, Apple Inc.) builds a relation of `founded` between the name entity *Steve Jobs* and the organization entity *Apple Inc*. In this paper, we employ two data sets from Freebase, i.e., `FB15K` used in (Bordes et al. 2014) and `FB13` used in (Socher et al. 2013). We list statistics of these data sets in Table 1.

Table 1: Statistics of data sets.

Dataset	#Rel	#Ent	#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
FB40K	1,336	39528	370,648	67,946	96,678

Link Prediction

Link prediction aims to predict the missing h or t for a relation fact triple (h, r, t), used in (Bordes et al. 2011; 2012; 2013). In this task, for each position of missing entity, the system is asked to rank a set of candidate entities from the knowledge graph, instead of only giving one best result. As set up in (Bordes et al. 2011; 2013), we conduct experiments using the data sets WN18 and FB15K.

In testing phase, for each test triple (h, r, t), we replace the head/tail entity by all entities in the knowledge graph, and rank these entities in descending order of similarity scores calculated by score function f_r. Following (Bordes et al. 2013), we use two measures as our evaluation metric: (1) mean rank of correct entities; and (2) proportion of correct entities in top-10 ranked entities (Hits@10). A good link predictor should achieve lower mean rank or higher Hits@10. In fact, a corrupted triple may also exist in knowledge graphs, which should be also considered as correct. However, the above evaluation may under-estimate those systems that rank these corrupted but correct triples high. Hence, before ranking we may filter out these corrupted triples which have appeared in knowledge graph. We name the first evaluation setting as "Raw" and the latter one as "Filter".

Since we use the same data sets, we compare our models with baselines reported in (Bordes et al. 2013; Wang et al. 2014). For experiments of TransR and CTransR, we select learning rate λ for SGD among $\{0.1, 0.01, 0.001\}$, the margin γ among $\{1, 2, 4\}$, the dimensions of entity embedding k and relation embedding d among $\{20, 50, 100\}$, the batch size B among $\{20, 120, 480, 1440, 4800\}$, and α for CTransR among $\{0.1, 0.01, 0.001\}$. The best configuration is determined according to the mean rank in validation set. The optimal configurations are $\lambda = 0.001$, $\gamma = 4$, $k = 50$, $d = 50$, $B = 1440$, $\alpha = 0.001$ and taking L_1 as dissimilarity on WN18; $\lambda = 0.001$, $\gamma = 1$, $k = 50$, $d = 50$, $B = 4800$, $\alpha = 0.01$ and taking L_1 as dissimilarity on FB15K. For both datasets, we traverse all the training triplets for 500 rounds.

Evaluation results on both WN18 and FB15K are shown in Table 2. From the table we observe that: (1) TransR and CTransR outperform other baseline methods including TransE and TransH significantly and consistently. It indicates that TransR finds a better trade-off between model complexity and expressivity. (2) CTransR performs better than TransR, which indicates that we should build fine-grained models to handle complicated internal correlations under each relation type. CTransR is a preliminary exploration; it will be our future work to build more sophisticated models for this purpose. (3) The "bern" sampling trick works well for both TransH and TransR, especially on FB15K which have much more relation types.

In Table 3, we show separate evaluation results by mapping properties of relations [1] on FB15K. We can see TransR achieves great improvement consistently on all mapping categories of relations, especially when (1) predicting "1-to-1" relations, which indicates that TransR provides more precise representation for both entities and relation and their complex correlations, as illustrated in Fig. 1; and (2) predicting the *1* side for "1-to-N" and "N-to-1" relations, which shows the ability of TransR to discriminate relevant from irrelevant entities via relation-specific projection.

Table 4: ⟨Head, Tail⟩ examples of some clusters for the relation "location_location_contains".

	⟨Head, Tail⟩
1	⟨Africa, Congo⟩, ⟨Asia, Nepal⟩, ⟨Americas, Aruba⟩, ⟨Oceania, Federated States of Micronesia⟩
2	⟨United States of America, Kankakee⟩, ⟨England, Bury St Edmunds⟩, ⟨England, Darlington⟩, ⟨Italy, Perugia⟩
3	⟨Georgia, Chatham County⟩, ⟨Idaho, Boise⟩, ⟨Iowa, Polk County⟩, ⟨Missouri, Jackson County⟩, ⟨Nebraska, Cass County⟩
4	⟨Sweden, Lund University⟩, ⟨England, King's College at Cambridge⟩, ⟨Fresno, California State University at Fresno⟩, ⟨Italy, Milan Conservatory⟩

Table 4 gives some cluster examples for the relation "location_location_contains" in FB15K training triples. We can find obvious patterns that: Cluster#1 is about continent containing country, Cluster#2 is about country containing city, Cluster#3 is about state containing county, and Cluster#4 is about country containing university. It is obvious that by clustering we can learn more precise and fine-grained relation embeddings, which can further help improve the performance of knowledge graph completion.

Triple Classification

Triple classification aims to judge whether a given triple (h, r, t) is correct or not. This is a binary classification task, which has been explored in (Socher et al. 2013; Wang et al. 2014) for evaluation. In this task, we use three data sets, WN11, FB13 and FB15K, following (Wang et al. 2014), where the first two datasets are used in (Socher et al. 2013).

We need negative triples for evaluation of binary classification. The data sets WN11 and FB13 released by NTN (Socher et al. 2013) already have negative triples, which are obtained by corrupting correct triples. As FB15K with negative triples has not been released by previous work, we construct negative triples following the same setting in (Socher et al. 2013). For triple classification, we set a relation-specific threshold δ_r. For a triple (h, r, t), if the dissimilarity score obtained by f_r is below δ_r, the triple will be classified

[1] Mapping properties of relations follows the same rules in (Bordes et al. 2013).

Table 2: Evaluation results 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 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. 2013)	263	251	75.4	89.2	243	125	34.9	47.1
TransH (unif) (Wang et al. 2014)	318	303	75.4	86.7	211	84	42.5	58.5
TransH (bern) (Wang et al. 2014)	401	388	73.0	82.3	212	87	45.7	64.4
TransR (unif)	232	219	78.3	91.7	226	78	43.8	65.5
TransR (bern)	238	225	**79.8**	92.0	**198**	77	48.2	68.7
CTransR (unif)	243	230	78.9	**92.3**	233	82	44	66.3
CTransR (bern)	**231**	**218**	79.4	**92.3**	199	**75**	**48.4**	**70.2**

Table 3: Evaluation results on FB15K by mapping properties of relations. (%)

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 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. 2013)	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH (unif) (Wang et al. 2014)	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransH (bern) (Wang et al. 2014)	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR (unif)	76.9	77.9	**38.1**	66.9	76.2	38.4	76.2	69.1
TransR (bern)	78.8	**89.2**	34.1	69.2	79.2	37.4	**90.4**	72.1
CTransR (unif)	78.6	77.8	36.4	68.0	77.4	37.8	78.0	70.3
CTransR (bern)	**81.5**	89.0	34.7	**71.2**	**80.8**	38.6	90.1	**73.8**

as positive, otherwise negative. δ_r is optimized by maximizing classification accuracies on the validation set.

For WN11 and FB13, we compare our models with baseline methods reported in (Wang et al. 2014) who used the same data sets. As mentioned in (Wang et al. 2014), for a fair comparison, all reported results are without combination with word embedding.

Since FB15K is generated by ourselves according to the strategy in (Socher et al. 2013), the evaluation results are not able to compare directly with those reported in (Wang et al. 2014). Hence, we implement TransE and TransH, and use the NTN code released by (Socher et al. 2013), and evaluate on our FB15K data set for comparison.

For experiments of TransR we select learning rate λ for SGD among $\{0.1, 0.01, 0.001, 0.0001\}$, the margin γ among $\{1, 2, 4\}$, the dimensions of entity embedding k , relation embedding d among $\{20, 50, 100\}$ and the batch size B among $\{20, 120, 480, 960, 4800\}$. The best configuration is determined according to accuracy in validation set.The optimal configurations are: $\lambda = 0.001$, $\gamma = 4$, $k, d = 20$, $B = 120$ and taking L_1 as dissimilarity on WN11; $\lambda = 0.0001$, $\gamma = 2$, $k, d = 100$ and $B = 480$ and taking L_1 as dissimilarity on FB13. For both datasets, we traverse all the training triples for 1000 rounds.

Evaluation results of triple classification is shown in Table 5. From Table 5, we observe that: (1) On WN11, TransR significantly outperforms baseline methods including TransE and TransH. (2) Neither TransE, TransH nor TransR can outperform the most expressive model NTN on FB13. In contrast, on the larger data set FB15K, TransE, TransH and TransR perform much better than NTN. The results may correlate with the characteristics of data sets: There are $1,345$ relation types in FB15K and only 13 relations types in FB13. Meanwhile, the number of entities and relational facts in the two data sets are close. As discussed in (Wang et al. 2014), the knowledge graph in FB13 is much denser than FB15K and even WN11. It seems that the most expressive model NTN can learn complicated correlations using tensor transformation from the dense graph of FB13. In contrast, simpler models are able to better handle the sparse graph of FB15K with good generalization. (3) Moreover, the "bern" sampling technique improves the performance of TransE, TransH and TransR on all three data sets.

As shown in (Wang et al. 2014), the training time of TransE and TransH are about 5 and 30 minutes, respectively. The computation complexity of TransR is higher than both TransE and TransH, which takes about 3 hours for training.

Table 5: Evaluation results of triple classification. (%)

Data Sets	WN11	FB13	FB15K
SE	53.0	75.2	-
SME (bilinear)	70.0	63.7	-
SLM	69.9	85.3	-
LFM	73.8	84.3	-
NTN	70.4	**87.1**	68.5
TransE (unif)	75.9	70.9	79.6
TransE (bern)	75.9	81.5	79.2
TransH (unif)	77.7	76.5	79.0
TransH (bern)	78.8	83.3	80.2
TransR (unif)	85.5	74.7	81.7
TransR (bern)	**85.9**	82.5	83.9
CTransR (bern)	85.7	-	**84.5**

Relation Extraction from Text

Relation extraction aims to extract relational fact from large-scale plain text, which is an important information source to enrich knowledge graphs. Most exiting methods (Mintz et al. 2009; Riedel, Yao, and McCallum 2010; Hoffmann et al. 2011; Surdeanu et al. 2012) take knowledge graphs as distant supervision to automatically annotate sentences in large-scale text corpora as training instances, and then extract textual features to build relation classifiers. These methods only use plain texts to reason new relational fact; meanwhile knowledge graph embeddings perform link prediction only based on existing knowledge graphs.

It is straightforward to take advantage of both plain texts and knowledge graphs to infer new relational facts. In (Weston et al. 2013), TransE and text-based extraction model were combined to rank candidate facts and achieved promising improvement. Similar improvements are also found over TransH (Wang et al. 2014). In this section, we will investigate the performance of TransR when combining with text-based relation extraction model.

We adopt NYT+FB, also used in (Weston et al. 2013), to build text-based relation extraction model. In this data set, entities in New York Times Corpus are annotated with Stanford NER and linked to Freebase.

In our experiments, we implement the same text-based extraction model proposed in (Weston et al. 2013) which is named as Sm2r. For the knowledge graph part, (Weston et al. 2013) used a subset restricted to the top 4 million entities with 23 thousand relation types. As TransH has not released the dataset and TransR will take too long to learn from 4 million entities, we generate a smaller data set FB40K ourselves, which contains all entities in NYT and $1,336$ relation types. For test fairness, from FB40K we remove all triples whose entity pairs have appeared in the testing set of NYT. As compared to previous results in (Weston et al. 2013; Wang et al. 2014), we find that learning with FB40K does not significantly reduce the effectiveness of TransE and TransH. Hence we can safely use FB40K to demonstrate the effectiveness of TransR.

Following the same method in (Weston et al. 2013), we combine the scores from text-based relation extraction model with the scores from knowledge graph embeddings to rank test triples, and get precision-recall curves for TransE, TransH and TransR. Since the freebase part of our data set is built by ourselves, different from the ones in (Wang et al. 2014), the evaluation results cannot be compared directly with those reported in (Wang et al. 2014). Hence, we implement TransE, TransH and TransR by ourselves. We set the embedding dimensions $k, d = 50$, the learning rate $\lambda = 0.001$, the margin $\gamma = 1.0$, $B = 960$, and dissimilarity metric as L_1. Evaluation curves are shown in Figure 2.

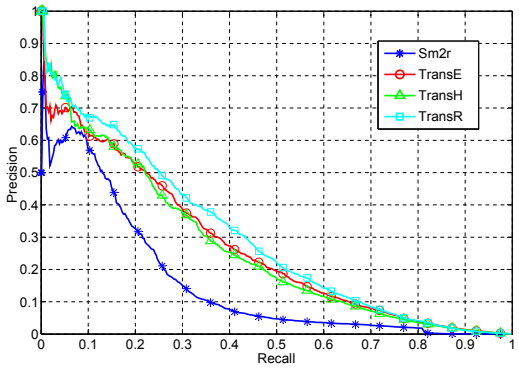

Figure 2: Precision-recall curves of TransE, TransH and TransR for relation extraction from text.

From the table we observe that TransR outperforms TransE and is comparable with TransH when recall ranges $[0, 0.05]$, and outperforms all baselines including TransE and TransH when recall ranges $[0.05, 1]$.

Recently, the idea of embeddings has also been widely used for representing words and texts (Bengio et al. 2003; Mikolov et al. 2013a; 2013b; Mikolov, Yih, and Zweig 2013), which may be used for text-based relation extraction.

Conclusion and Future Work

In this paper, we propose TransR, a new knowledge graph embedding model. TransR embeds entities and relations in distinct entity space and relation space, and learns embeddings via translation between projected entities. In addition, we also propose CTransR, which aims to model internal complicated correlations within each relation type based on the idea of piecewise linear regression. In experiments, we evaluate our models on three tasks including link prediction, triple classification and fact extraction from text. Experiment results show that TransR achieves consistent and significant improvements compared to TransE and TransH.

We will explore the following further work:

- Existing models including TransR consider each relational fact separately. In fact, relations correlate with each other with rich patterns. For example, if we know (goldfish, `kind_of`, fish) and (fish, `kind_of`, animal), we can infer (goldfish, `kind_of`, animal) since the relation type `kind_of` is *transitive*. We may take advantages of these relation patterns for knowledge graph embeddings.
- In relational fact extraction from text, we simply perform linear weighted average to combine the scores from text-

side extraction model and the knowledge graph embedding model. In future, we may explore a unified embedding model of both text side and knowledge graph.

- CTransR is an initial exploration for modeling internal correlations within each relation type. In future, we will investigate more sophisticated models for this purpose.

Acknowledgments

This work is supported by the 973 Program (No. 2014CB340501), the National Natural Science Foundation of China (NSFC No. 61133012 and 61202140) and Tsinghua-Samsung Joint Lab. We thank all anonymous reviewers for their constructive comments.

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *JMLR* 3:1137–1155.

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 KDD*, 1247–1250.

Bordes, A.; Weston, J.; Collobert, R.; Bengio, Y.; et al. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of AAAI*, 301–306.

Bordes, A.; Glorot, X.; Weston, J.; and Bengio, Y. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In *Proceedings of AISTATS*, 127–135.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, 2787–2795.

Bordes, A.; Glorot, X.; Weston, J.; and Bengio, Y. 2014. A semantic matching energy function for learning with multi-relational data. *Machine Learning* 94(2):233–259.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL-HLT*, 541–550.

Jenatton, R.; Roux, N. L.; Bordes, A.; and Obozinski, G. R. 2012. A latent factor model for highly multi-relational data. In *Proceedings of NIPS*, 3167–3175.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *Proceedings of ICLR*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, 3111–3119.

Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of HLT-NAACL*, 746–751.

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 ACL-IJCNLP*, 1003–1011.

Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of ICML*, 809–816.

Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2012. Factorizing yago: scalable machine learning for linked data. In *Proceedings of WWW*, 271–280.

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

Ritzema, H., et al. 1994. *Drainage principles and applications.*

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of NIPS*, 926–934.

Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Manning, C. D. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP*, 455–465.

Sutskever, I.; Tenenbaum, J. B.; and Salakhutdinov, R. 2009. Modelling relational data using bayesian clustered tensor factorization. In *Proceedings of NIPS*, 1821–1828.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*, 1112–1119.

Weston, J.; Bordes, A.; Yakhnenko, O.; and Usunier, N. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of EMNLP*, 1366–1371.