TransT: Type-Based Multiple Embedding Representations for Knowledge Graph Completion

Shiheng Ma[1], Jianhui Ding[1], Weijia Jia[1(✉)], Kun Wang[1,2], and Minyi Guo[1]

[1] Shanghai Jiao Tong University, Shanghai 200240, China
{ma-shh,ding-jh}@sjtu.edu.cn, {jia-wj,guo-my}@cs.sjtu.edu.cn
[2] Nanjing University of Posts and Telecommunications, Nanjing 210042, China
kwang@njupt.edu.cn

Abstract. Knowledge graph completion with representation learning predicts new entity-relation triples from the existing knowledge graphs by embedding entities and relations into a vector space. Most existing methods focus on the structured information of triples and maximize the likelihood of them. However, they neglect semantic information contained in most knowledge graphs and the prior knowledge indicated by the semantic information. To overcome this drawback, we propose an approach that integrates the structured information and entity types which describe the categories of entities. Our approach constructs relation types from entity types and utilizes type-based semantic similarity of the related entities and relations to capture prior distributions of entities and relations. With the type-based prior distributions, our approach generates multiple embedding representations of each entity in different contexts and estimates the posterior probability of entity and relation prediction. Extensive experiments show that our approach outperforms previous semantics-based methods. The source code of this paper can be obtained from https://github.com/shh/transt.

Keywords: Knowledge graph · Representation learning
Multiple embedding

1 Introduction

Knowledge graphs (KGs) have become a key resource for artificial intelligence applications including question answering, recommendation system, knowledge inference, etc. Recently years, several large-scale KGs, such as Freebase [2], DBpedia [1], NELL [4], and Wikidata [25], have been built by automatically extracting structured information from text and manually adding structured information according to human experiences. Although large-scale KGs have contained billions of triples, the extracted knowledge is still a small part of the real-world knowledge and probably contains errors and contradictions. For example, 71% of people in Freebase have no known place of birth, and 75% have

M. Ceci et al. (Eds.): ECML PKDD 2017, Part I, LNAI 10534, pp. 717–733, 2017.
https://doi.org/10.1007/978-3-319-71249-9_43

no known nationality [5]. Therefore, knowledge graph completion (KGC) is a crucial issue of KGs to complete or predict the missing structured information based on existing KGs.

A typical KG transforms real-world and abstract information into triples denoted as (head entity, relation, tail entity), (h, r, t) for short. To complete or predict the missing element of triples, such as $(h, r, ?)$, $(h, ?, t)$, $(?, r, t)$, representation learning (RL) is widely deployed. RL embeds entities and relations into a vector space, and has produced many successful translation models including TransE [3], TransH [26], TransR [16], TransG [29], etc. These models aim to generate precise vectors of entities and relations following the principle $h + r \approx t$, which means t is translated from h by r.

Most RL-based models concentrate on structured information in triples and neglect the rich semantic information of entities and relations, which is contained in most KGs. Semantic information includes types, descriptions, lexical categories and other textual information. Although these models have significantly improved the embedding representations and increased the prediction accuracy, there is still room for improvement by exploiting semantic information in the following two aspects.

Representation of entities. One of the main obstacles of KGC is the polysemy of entities or relations, i.e., each entity or relation may have different semantics in different triples. For example, in the triple (Isaac_Newton, birthplace, Lincolnshire), Newton is a person, while in (Isaac_Newton, author_of, Opticks), Newton is a writer or physicist. This is a very common phenomenon in KGs and it causes difficulty in vector representations. Most works focus on entity polysemy and utilize linear transformations to model different semantics of an entity in different triples to attain the high accuracy. However, they represent each entity as a single vector which cannot capture the uncertain semantics of entities. This is a critical limitation for modeling the rich semantics.

Estimation of posterior probability. Another problem of most previous works is the neglect of prior probability of known triples. Most previous works optimize the maximum likelihood (ML) estimation of vector representations. Few models discuss the posterior probability, which incorporates a prior distribution to augment optimization objectives. Specifically, previous ML models essentially maximize the probability $p(h, r, t)$ that h, r, t form a triple $p(h, r, t)$. When predicting the missing tail of $(h, r, ?)$, however, h and r are already known and they may influence the possible choices of t. Thus, the posterior probability $p(t \mid h, r)$ of predicting t is a more accurate expression of optimization goals than $p(h, r, t)$. In another word, we could prune the possible choices based on the prior probability of the missing element in a triple.

To address the two issues above, we propose a type-based multiple embedding model (TransT). TransT fully utilizes the entity type information which represents the categories of entities in most KGs. Compared with descriptions and other semantic information, types are simpler and more specific because types of an entity are unordered and contain less noise. Moreover, we can construct or extend entity types from other semantic information, if there is no

explicit type information in a KG. For example, in Wordnet, we can construct types from the lexical categories of entities. Other semantic information does not have this advantage. In addition to entity types, we construct multiple types of relations from common types of related entities. We measure the semantic similarity of entities and relations based on entity types and relation types. With this type-based semantic similarity, we integrate type information into entity representations and prior estimation which are detailed below.

We model each entity as multiple semantic vectors with type information to represent entities more accurately. Different from using semantics-based linear transformations to separate the mixed representation [16,26,27,31], TransT models the multiple semantics separately and utilizes the semantic similarity to distinguish entity semantics. In order to capture entity semantics accurately, we dynamically generate new semantic vectors for different contexts.

We utilize the type-based semantic similarity to incorporate prior probability in the optimization objective. It is inspired by the observation that the missing element of a triple semantically correlates to the other two elements. Specifically, all entities appearing in the head (or tail) with the same relation have some common types, or these entities have some common type owned by the entities appearing in the tail (or head). In the "Newton" example mentioned above, if the head of (Isaac_Newton, author_of, Opticks) is missing, we can predict the head is an entity with "author" or "physicist" since we know the relation is "author_of" and the tail is "Opticks", a physics book. Therefore, we design a type-based semantic similarity based on the similarity of type sets. With this similarity, TransT captures the prior probability of missing elements in triples for the accurate posterior estimation.

Our contributions are summarized as follows:

- We propose a new approach for fusing structured information and type information. We construct multiple types of relations from entity types and design the type-based semantic similarity for multiple embedding representations and prior knowledge discovering.
- We propose a multiple embedding model that represents each entity as multiple vectors with specific semantics.
- We estimate prior probabilities for entity and relation predictions based on the semantic similarity between elements of triples in KGs.

The rest of this paper is organized as follows. Section 2 shows the recent studies of KGC. Section 3 introduces our approach including multiple embedding model, prior probability estimation, and objective function optimization. Section 4 displays the evaluation of our approach on FB15K and WN18. Section 5 concludes the paper.

2 Related Work

TransE [3] proposes the principle $h + r \approx t$ to assign a single vector for each entity and relation by minimizing the energy function $\|\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}\|$ of every triple.

It is a simple and efficient model but unable to capture the rich semantics of entities and relations.

Some models revise function $\|\cdot\|$ in the energy functions for the complex structures in KGs. TransA [13] adaptively finds the optimal loss function without changing the norm function. Tatec [7] utilizes canonical dot products to design different energy functions for different relations. HolE [20] designs the energy function based on a tensor product which captures the interaction in features of entities and relations. ComplEx [24] represents entities and relations as complex-number vectors and calculates Hermitian dot product in the energy function. ManifoldE [28] expands the position of triples from one point to a hyperplane or sphere and calculates energy function for the two manifolds. KG2E [9] models the uncertainty of entities and relations by Gaussian embedding and defines KL divergence of entity and relation distributions as the energy function. ProjE [22] proposes a neural network model to calculate the difference between $h+r$ and t.

Some models design $\|\boldsymbol{h_r} + \boldsymbol{r} - \boldsymbol{t_r}\|$ to make an entity vector adaptive to different relations. They aim to find appropriate representations of $\boldsymbol{h_r}$ and $\boldsymbol{t_r}$. TransH [26] projects entity vector into hyperplanes of different relations. It represents $\boldsymbol{h_r}$ as the projection vector of $\boldsymbol{h_r}$ on the relation hyperplanes. TransR [16] adjusts entity vectors by transform matrices instead of projections. It represents $\boldsymbol{h_r}$ as the result of linear transformation of $\boldsymbol{h}$. TranSparse [12] considers the transform matrix should reflect the heterogeneous and imbalance of entity pairs and improves the transform matrix into two sparse matrices corresponding to the head entity and the tail entity respectively. TransG [29] considers relations also have multiple semantics like entities. It generates multiple vectors for each relation.

Semantic information, such as types, descriptions, and other textual information, is an important supplement to structured information in KGs. DKRL [30] represents entity descriptions as vectors for tuning the entity and relation vectors. SSP [27] modifies TransH by using the topic distribution of entity descriptions to construct semantic hyperplanes. Entity descriptions are also used to derive a better initialization for training models [17]. With type information, type-constraint model [14] selects negative samples according to entity and relation types. TKRL [31] encodes type information into multiple representations in KGs with the help of hierarchical structures. It is a variant of TransR with semantic information and it is the first model introducing type information. However, TKRL also neglects the two issues mentioned above.

There are several other approaches to modeling KGs as graphs. PRA [15] and SFE [8] predict missing relations from existing paths in KGs. These approaches consider that sequences of relations in paths between two entities can comprise the relation between the two entities. RESCAL [21], PITF [6] and ARE [19] complete KGs through retrieving their adjacent matrices. These approaches need to process large adjacent matrices of entities.

3 Methodology

3.1 Overview

The goal of our model is to obtain the vector representations of entities and relations, which maximize the prediction probability over all existing triples. The prediction probability is a conditional probability because except the missing element, the rest two elements in a triple are known. Specifically, when predicting the tail entity for a triple (h, r, t), we expect to maximize the probability of t under the condition that the given triple satisfies the principle $h + r \approx t$ and the head entity and relation are h and r. We denote this conditional probability entity as $p(t \mid h, r, true)$ which means triple $(h, r, *)$ is "true". "true" represents the triple satisfies $h + r \approx t$ principle. "true" triples are also called correct triples in this paper. Maximizing this probability is the aim of the tail prediction. According to Bayes' theorem [10], $p(t \mid h, r, true)$ can be seen as a posterior probability and its correlation with the prior probability is derived as

$$p(t \mid h, r, true) = \begin{cases} \frac{p(true \mid h,r,t)\, p(t \mid h,r)}{p(true \mid h,r)} & p(t \mid h, r) \neq 0 \\ 0 & p(t \mid h, r) = 0, \end{cases} \tag{1}$$

where $p(true \mid h, r, t)$ is the probability that (h, r, t) is "true", $p(t \mid h, r)$ is the prior probability of t. To obtain the most possible entity, we can only compare probabilities of triples $(h, r, *)$. All these probabilities have the same $p(t \mid h, r)$. Thus, we can omit $p(true \mid h, r)$ in (1):

$$p(t \mid h, r, true) \propto p(true \mid h, r, t)\, p(t \mid h, r). \tag{2}$$

Similarly, the objective of the head prediction is

$$p(h \mid r, t, true) \propto p(true \mid h, r, t)\, p(h \mid r, t), \tag{3}$$

and the objective of the relation prediction is

$$p(r \mid h, t, true) \propto p(true \mid h, r, t)\, p(r \mid h, t). \tag{4}$$

All the three formulas have two components: likelihood and prior probability. $p(true \mid h, r, t)$ is the likelihood estimated by the multiple embedding representations. The other component is the prior probability estimated by the semantic similarity. TransT introduces a type-based semantic similarity to estimate the two components and optimizes the vector representations to maximize these posterior probabilities over the training set.

3.2 Type-Based Semantic Similarity

In order to estimate the likelihood and prior probability, we introduce the semantic similarity to measure the distinction of entity semantics with the type information.

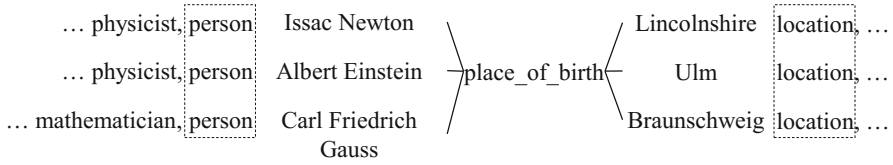

Fig. 1. The entities in the head or tail of a relation have some common types. In this example, all the head entities have "person" type and all the tail entities have "location" type. Therefore, "person" and "location" are the head and tail type of this relation, respectively. Moreover, if we relax this constraint, "physicist" type is also the head type of the relation since most head entities contain this type.

All entities appearing in the head (or tail) with the same relation have some common types. These common types determine this relation as shown in Fig. 1. There are head and tail positions for each relation. Thus, each relation r has two type sets $T_{r,head}$ for entities in the head and $T_{r,tail}$ for entities in the tail. We construct type sets of relations from these common types:

$$T_{r,head} = \bigcap_{\substack{e \in Head_r \\ \rho}} T_e \qquad T_{r,tail} = \bigcap_{\substack{e \in Tail_r \\ \rho}} T_e, \tag{5}$$

where T_e is the type sets of entity e, $Head_r$ and $Tail_r$ are the set of entities appearing respectively in the head and tail with relation r. $\bigcap_\rho$ is a special intersection which contains elements belonging to most of the type sets. This intersection can capture more type information of entities than the normal intersection. However, more information may include more noises. Thus, we balance the influence by the parameter ρ, which is the lowest frequency of types in all T_e.

With the type information of entities and relations, we denote the asymmetric semantic similarity of relations and entities as the following similarity of two sets inspired by Jaccard Index [11]:

$$s(r_{head}, h) = \frac{|T_{r,head} \cap T_h|}{|T_{r,head}|} \quad s(r_{tail}, t) = \frac{|T_{r,tail} \cap T_t|}{|T_{r,tail}|} \quad s(h, t) = \frac{|T_h \cap T_t|}{|T_h|}, \tag{6}$$

where $s(r_{head}, h)$ is the semantic similarity between the relation and the head, $s(r_{tail}, t)$ is the semantic similarity between the relation and the tail, $s(h, t)$ is the semantic similarity between the head and tail.

The type-based semantic similarity plays an important role in the following estimations especially in the prior probability estimation.

3.3 Multiple Embedding Representations

Entities with rich semantics are difficult to be accurately represented in KGC. Thus it is difficult to measure the likelihood $p(true \,|\, h, r, t)$ accurately. In this section, we introduce the multiple embedding representations to capture the entity semantics for the accurate likelihood.

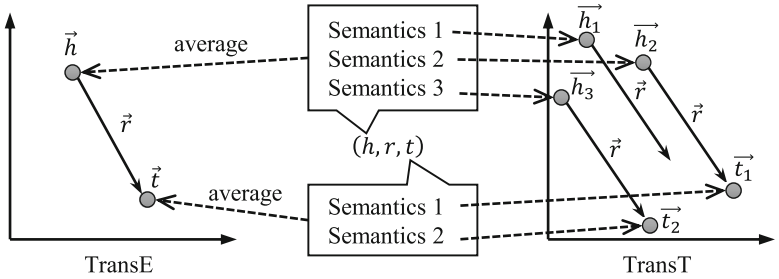

Fig. 2. TransE represents each entity as a single vector which tries to describe all semantics of the entity. Thus the vector representation is not accurate for any entity semantics. In TransT, separate representations of entity semantics describe the relationship among a triple more accurately.

As shown in Fig. 2, there is only one vector representation for one entity in previous work, e.g., TransE. To overcome this drawback, TransT represents each entity semantics as a vector and denotes each entity as a set of semantic vectors. In our approach, we embed each semantics into a vector space. We assume relations have single semantics and entities have multiple semantics. Thus, each relation is represented as a single vector. To adapt the rich entity semantics, we represent each entity as a set of semantic vectors instead of a single vector. Therefore, an entity can be viewed as a random variable of its multiple semantic vectors. Furthermore, the likelihood $p(true \mid h, r, t)$ depends on the expected probability of all possible semantic combinations of random variables h and t. This can define the likelihood of the vector representations for the triple as below

$$p(true \mid h, r, t) = \sum_{i=1}^{n_h} \sum_{j=1}^{n_t} w_{h,i} w_{t,j} p_{true}(v_{h,i}, v_r, v_{t,j}), \tag{7}$$

where n_h and n_t are the number of entity semantics of h and t; $w_h = (w_{h,1}, \ldots, w_{h,n_h})$ and $w_t = (w_{t,1}, \ldots, w_{t,n_t})$ are the distributions of random variables h and t; $v_{h,i}$, v_r, $v_{t,j}$ are the vectors of h, r, t; $p_{true}(v_{h,i}, v_r, v_{t,j})$ is the likelihood of the component with i-th semantic vector $v_{h,i}$ of h and j-th semantic vector $v_{t,j}$ of t. According to the principle $h + r \approx t$, this likelihood is determined by the difference between $h + r$ and t:

$$p_{true}(v_{h,i}, v_r, v_{t,j}) = \sigma(d(v_{h,i} + v_r, v_{t,j})), \tag{8}$$

where the distance function d measures this difference; the squashing function σ transforms values of d from 0 to $+\infty$ into probability values from 1 to 0 since the probability of a semantic combination is larger if the distance between their corresponding vectors is smaller. To satisfy the property, we set $d(x, y) = \|x - y\|_1$ (1-norm) and $\sigma(x) = e^{-x}$.

In order to capture entity semantics more accurately, we do not assign the specific semantics of entities and the size of their vector sets in advance. We

model the generating process of semantic vectors as a random process revised from Chinese restaurant process (CRP), a widely employed form for the Dirichlet process [10]. This avoids the man-made subjectivity for setting n_h and n_t.

In training process, the tail (or head) entity in each triple generates a new semantic vector with the following probability

$$p_{new,tail}(h,r,t) = \left(1 - \max_{t_i \in Semantics_t} s(t_i, r_{tail})\right) \frac{\beta e^{-\|r\|_1}}{\beta e^{-\|r\|_1} + p(true \mid h,r,t)}, \quad (9)$$

where β is the scaling parameter in CRP which controls the generation probability. The bracketed formula means t more possibly generates a new semantics when the existing semantics are more different from r; the fraction part is similar to the CRP in TransG [29], which indicates that t possibly generates a new semantics if its current semantic set cannot represent t accurately. Similarly, the new semantic vector of h can be generated with the probability $p_{new,head}(h,r,t)$.

3.4 Prior Probability Estimation

In our model, the prior probability reflects features of a KG from the perspective of semantics. We estimate the prior probabilities (2), (3) and (4) by the type-based semantic similarity.

Note that the type sets of three elements in a triple have obvious relationships. We can estimate the prior distribution of the missing element from the semantic similarity between the missing element and the others.

When we predict t in a triple (h,r,t), the entities with more common types belonging to r and h have higher probability. Therefore, we use the semantic similarity between t and its context $(*,h,r)$ to estimate t's prior probability:

$$p(t \mid h,r) \propto s(r_{tail},t)^{\lambda_{tail}}\, s(h,t)^{\lambda_{relation}}, \quad (10)$$

where $\lambda_{relation}, \lambda_{head}, \lambda_{tail} \in \{0,1\}$ are the similarity weights, because h and r have different impacts on the prior probability of t. We use these weights to select different similarity for different situation. Similarly, the prior estimation of head entity h is

$$p(h \mid r,t) \propto s(r_{head},h)^{\lambda_{head}}\, s(t,h)^{\lambda_{relation}}. \quad (11)$$

By the similar derivation, the prior estimation of relation r is

$$p(r \mid h,t) \propto s(r_{head},h)^{\lambda_{head}}\, s(r_{tail},t)^{\lambda_{tail}}. \quad (12)$$

To adapt different datasets, the parameters, $\lambda_{relation}$, λ_{head} and λ_{tail}, should be adjusted.

3.5 Objective Function with Negative Sampling

To achieve the goal of maximizing posterior probabilities, we define the objective function as the sum of prediction errors with negative sampling [18].

For a triple (h, r, t) in the training set Δ, we sample its negative triple $(h', r', t') \notin \Delta$ by replacing one element with another entity or relation. When predicting different elements of a triple, we replace the corresponding elements to obtain the negative triples. Therefore, the prediction error is denoted as a piecewise function:

$$l(h, r, t, h', r', t') = \begin{cases} -\ln p(h \mid r, t, true) + \ln p(h' \mid r, t, true) & h' \neq h \\ -\ln p(t \mid h, r, true) + \ln p(t' \mid h, r, true) & t' \neq t \\ -\ln p(r \mid h, t, true) + \ln p(r' \mid h, t, true) & r' \neq r, \end{cases} \tag{13}$$

where we measure the performance of the probability estimation by the probability difference of the training triple and its negative sample. We define the objective function as the total of prediction errors:

$$\sum_{(h,r,t)\in\Delta} \sum_{(h',r',t')\in\Delta'_{(h,r,t)}} \max\left\{0, \gamma + l(h, r, t, h', r', t')\right\}, \tag{14}$$

where $\Delta'_{(h,r,t)}$ is the negative triple set of (h, r, t).

The total posterior probabilities of predictions are maximized through the minimization of the objective function. Moreover, stochastic gradient descent is applied to optimize the objective function, and we normalize the semantic vectors of entities to avoid overfitting.

4 Experiments

In this paper, we adopt two public benchmark datasets that are the subsets of Freebase and Wordnet, FB15K [3] and WN18 [3], to evaluate our models on knowledge graph completion and triple classification [23]. As for knowledge graph completion, we divide the task into two sub-tasks: entity prediction and relation prediction. Following [3], we split datasets into train, validation and test set. The statistics of datasets are listed in Table 1.

Type information of entities in FB15K has been collected in [31]. There are 4,064 types in FB15K and the average number of types for entities is approximately 12. There is no explicit type information in WN18. Thus we construct type sets of entities from lexical categories. For example, the name of "__trade_name_NN_1" contains its lexical category "NN" (noun), we define the type of "__trade_name_NN_1" as "NN". Because each entity in Wordnet represents the exact semantics, the number of types for entities is 1. There are 4 types in WN18.

The baselines include three semantics-based models: TKRL [31] utilizes entity types; DKRL [30] and SSP [27] take advantage of entity descriptions.

4.1 Entity Prediction

Entity prediction aims at predicting the missing entity when given an entity and a relation, i.e. we predict t given $(h, r, *)$, or predict h given $(*, r, t)$. FB15K and WN18 are the benchmark dataset for this task.

Table 1. Statistics of datasets

Dataset	#Ent	#Rel	#Train	#Valid	#Test
FB15k	14,951	1,345	483,142	50,000	59,071
WN18	40,943	18	141,442	5,000	5,000

Evaluation Protocol. We adopt the same protocol used in previous studies. For each triple (h, r, t) in the test set, we replace the tail t (or the head h) with every entity in the dataset. We calculate the probabilities of all replacement triples and rank these probabilities in descending order. Two measures are considered as evaluation metrics: Mean Rank, the mean rank of original triples in the corresponding probability ranks; HITS@N, the proportion of original triples whose rank is not larger than N. In this task, we use HITS@10. This setting is called "Raw". Some of these replacement triples exist in the training, validation, or test sets, thus ranking them ahead of the original triple is acceptable. Therefore, we filter out these triple to eliminate this case. This filtering setting is called "Filter". In both settings, a higher HITS@10 and a lower Mean Rank mean better performance.

Experiment Settings. As the datasets are the same, we directly reuse the best results of several baselines from the literature [16,26,31]. We have attempted several settings on the validation dataset to get the best configuration. Under the "unif." sampling strategy [26], the optimal configurations are: learning rate $\alpha = 0.001$, vector dimension $k = 50$, margin $\gamma = 3$, CRP factor $\beta = 0.0001$, similarity weights $\lambda_{head} = \lambda_{head} = 0$, $\lambda_{relation}$ is set to 0 or 1 for different relations depending on statistical results of the training set, on WN18; $\alpha = 0.00025$, $k = 300$, $\gamma = 3.5$, $\beta = 0.0001$, $\lambda_{head} = \lambda_{tail} = 1$, $\lambda_{relation} = 0$ on FB15K. We train the model until convergence.

Results. Evaluation results on FB15K and WN18 are shown in Table 2. On FB15K, we compare impacts of multiple vectors and type information. Single or Multiple means entities are represented as single vectors or multiple vectors. Type or no type means type information is used or not. From the result, we observe that:

1. TransT significantly outperforms all baselines on WN18. On FB15K, TransT significantly outperforms all baselines with the filter setting. This demonstrates that our approach successfully utilizes the type information and multiple entity vectors can capture the different semantics of every entity more accurately than linear transformations of single entity vector.
2. Compared with baselines, TransT has the largest difference between the results of Raw and Filter settings on FB15K. This indicates that TransT ranks more correct triples ahead of the original triple. This is caused by the prior estimation of TransT. Specifically, if the predicted element is the head of the original triple, these correct triples have the same relation and tail. Thus, when we learn the prior knowledge from the training set, the head entities

Table 2. Evaluation results on entity prediction

FB15K	Mean rank		HITS@10 (%)	
	Raw	Filter	Raw	Filter
TransE	238	143	46.4	62.1
TransH	212	87	45.7	64.4
TransR	199	77	47.2	67.2
DKRL (CBOW)	236	151	38.3	51.8
DKRL (CNN)	200	113	44.3	57.6
DKRL (CNN)+TransE	181	91	49.6	67.4
TKRL (RHE)	184	68	49.2	69.4
TKRL (WHE+STC)	202	87	50.3	73.4
SSP (Std.)	**154**	77	57.1	78.6
SSP (Joint)	163	82	**57.2**	79.0
TransT (type information)	181	72	54.0	82.3
TransT (multiple vectors)	215	62	50.6	83.6
TransT (multiple+type)	199	**46**	53.3	**85.4**
WN18	Mean Rank		HITS@10 (%)	
	Raw	Filter	Raw	Filter
TransE	263	251	75.4	89.2
TransH	401	338	73.0	82.3
TransR	238	225	79.8	92.0
SSP (Std.)	204	193	81.3	91.4
SSP (Joint)	168	156	81.2	93.2
TransT	**137**	**130**	**92.7**	**97.4**

of these correct triples have higher semantic similarities to the head entity of the original triple than other triples. TransT utilizes these similarities to estimate the prior probability resulting in ranking similar entities higher. In fact, this phenomenon shows that the prior probability improves the prediction performance.

3. There is less difference between the results of Raw and Filter settings on WN18 than FB15K. The reason is that the type-based prior knowledge in WN18 is more accurate than that in FB15K. Specifically, WN18 includes 4 types with simple meanings: noun, verb, adjective and adverb. In addition, an entity in WN18 can only have one type. Thus, types in WN18 have stronger ability to distinguish different entities.
4. Both the two approaches, multiple-vector representation and type information, have their own advantages. Type information performs better in raw setting, while multiple-vector representation performs better in filter setting.

4.2 Relation Prediction

Relation prediction aims at predicting the missing relation when given two entities, i.e., we predict r given $(h, *, t)$. FB15K is the benchmark dataset for this task.

Evaluation Protocol. We adopt the same protocol used in entity prediction. For each triple (h, r, t) in the test set, we replace the relation r with every relation in the dataset. Mean Rank and HITS@1 are considered as evaluation metrics for this task.

Experiment Settings. As the datasets are the same, we directly reuse the experimental results of several baselines from the literature. We have attempted several settings on the validation dataset to get the best configuration. Under the "unif." sampling strategy, the optimal configurations are: learning rate $\alpha = 0.0001$, vector dimension $k = 300$, margin $\gamma = 3.0$, CRP factor $\beta = 0.001$, similarity weights $\lambda_{head} = \lambda_{tail} = 1$, $\lambda_{relation} = 0$.

Table 3. Evaluation results on relation prediction

Method	Mean rank		HITS@1 (%)	
	Raw	Filter	Raw	Filter
TransE	2.91	2.53	69.5	90.2
TransH	8.25	7.91	60.3	72.5
TransR	2.49	2.09	70.2	91.6
DKRL (CBOW)	2.85	2.51	65.3	82.7
DKRL (CNN)+TransE	2.41	2.03	69.8	90.8
TKRL (RHE)	2.12	1.73	71.1	92.8
TKRL (WHE+STC)	2.47	2.07	68.3	90.6
SSP (Std.)	**1.58**	1.22	69.9	89.2
SSP (Joint)	1.87	1.47	70.9	90.9
TransT	1.59	**1.19**	**72.0**	**94.1**

Results. Evaluation results on FB15K are shown in Table 3. From the result, we observe that:

1. TransT significantly outperforms all baselines. Compared with TKRL, which also utilized type information, TransT improves HITS@1 by 3.5% and Mean Rank by 0.88.
2. In the Raw setting, TransT also achieves the best performance. This result is different from the entity prediction task. The reason is more prior knowledge of relation predictions. In the entity prediction task, the prior knowledge is derived from relations. In the relation prediction task, the prior knowledge is

derived from the head and tail entities. The latter has more sources for prior estimation. Thus, TransT ranks more incorrect triples behind the original triple. This further supports the necessity of the prior probability.

4.3 Triple Classification

Triple classification aims at predicting whether a given triple is correct or incorrect, i.e., we predict the correctness of (h, r, t). FB15K is the benchmark dataset of this task.

Evaluation Protocol. We adopt the same protocol used in entity prediction. Since FB15K has no explicit negative samples, we construct negative triples following the same protocol used in [23]. For each triple (h, r, t) in the test set, if the probability of its correctness is below a threshold σ_r, the triple is incorrect; otherwise, it is correct. The thresholds $\{\sigma_r\}$ are determined on the validation dataset with negative samples.

Experiment Settings. As the datasets are the same, we directly reuse the experimental results of several baselines from the literature. We have attempted several settings on the validation dataset to get the best configuration. Under the "unif." sampling strategy, the optimal configurations are: learning rate $\alpha = 0.001$, vector dimension $k = 300$, margin $\gamma = 3.0$, CRP factor $\beta = 0.01$, similarity weights $\lambda_{head} = \lambda_{tail} = \lambda_{relation} = 0$.

Table 4. Evaluation results on triple classification

Method	Accuracy (%)
TransE	85.7
TransH	87.7
TransR	86.4
TKRL (RHE)	86.9
TKRL (WHE+STC)	88.5
TransT	91.0

Results. Evaluation results on FB15K are shown in Table 4. TransT outperforms all baselines significantly. Compared with the best result, TransT improves the accuracy by 2.5% and it is the only model whose accuracy is over 90%. This task shows the ability to discern which triples are correct.

4.4 Semantic Vector Analysis

We analyze the correlations between the semantic vector number and several statistical properties of different entities. We adopt the vector representations obtained by TransT and TransE during the entity prediction task on FB15K.

Figure 3 shows that the correlations between the number of semantic numbers and the average number of relations/types/triples for different entities. For an entity represented by more semantic vectors, it has more types and appears with more different relations and triples in the training set. Thus the entities with more semantic vectors have more complex semantics. Therefore, the result of TransT conforms to our understanding of the entity semantics.

Figure 4 shows that the prediction probabilities of several selected entities. Our approach generates at most 11 semantic vectors for entities. The entities with more semantic vectors have broader concepts. Thus, popular places and people including "Paris", "Alan Turing", have more semantic vectors than awards like "Film Award" and events like "2007 NBA draft". Compared with TransE, multiple semantic vectors improve prediction probability of most entities.

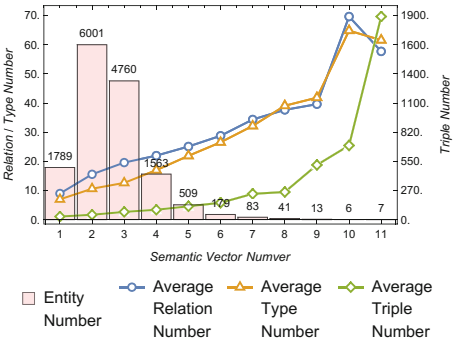

Fig. 3. The bar chart is the number of entities with different semantic numbers. The left y-axis is the number of relations or types. The right y-axis is the number of triples. The x-axis is the number of semantic vectors.

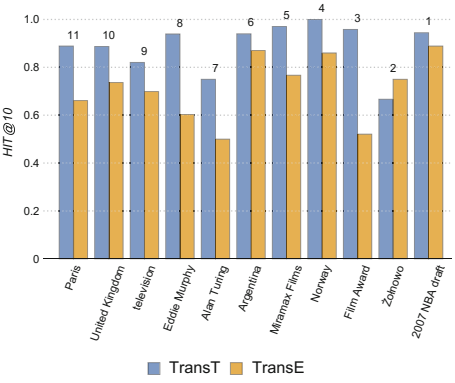

Fig. 4. HIT@10 of 11 entities with different numbers of semantic vectors. The number of semantic vectors are placed above the bars.

5 Conclusion

This paper proposes TransT, a new approach for KGC, which combines structured information and type information. With the type-based prior knowledge, TransT generates semantic vectors for entities in different contexts based on CRP and optimizes the posterior probability estimation. This approach makes full use of type information and accurately captures semantic features of entities. Extensive experiments show that TransT achieves markable improvements against the baselines.

Acknowledgments. This work is supported by Chinese National Research Fund (NSFC) Key Project No. 61532013; National China 973 Project No. 2015CB352401; NSFC No. 61572262; Shanghai Scientific Innovation Act of STCSM No. 15JC1402400; 985 Project of Shanghai Jiao Tong University No. WF220103001; China Postdoctoral Science Foundation No. 2017M610252 and China Postdoctoral Science Special Foundation No. 2017T100297.

References

1. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the web of data. J. Web Semant. **7**(3), 154–165 (2009)
2. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), pp. 1247–1250 (2008)
3. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems 26 (NIPS), pp. 2787–2795 (2013)
4. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr., E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI) (2010)
5. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 601–610 (2014)
6. Drumond, L., Rendle, S., Schmidt-Thieme, L.: Predicting RDF triples in incomplete knowledge bases with tensor factorization. In: Proceedings of the ACM Symposium on Applied Computing (SAC), pp. 326–331 (2012)
7. García-Durán, A., Bordes, A., Usunier, N.: Effective blending of two and three-way interactions for modeling multi-relational data. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI), vol. 8724, pp. 434–449. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44848-9_28
8. Gardner, M., Mitchell, T.M.: Efficient and expressive knowledge base completion using subgraph feature extraction. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1488–1498 (2015)
9. He, S., Liu, K., Ji, G., Zhao, J.: Learning to represent knowledge graphs with Gaussian embedding. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM), pp. 623–632 (2015)
10. Hjort, N.L., Holmes, C., Müller, P., Walker, S.G.: Bayesian Nonparametrics, vol. 28. Cambridge University Press, Cambridge (2010)
11. Jaccard, P.: Distribution de la Flore Alpine: dans le Bassin des dranses et dans quelques régions voisines. Rouge (1901)
12. Ji, G., Liu, K., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI), pp. 985–991 (2016)

13. Jia, Y., Wang, Y., Lin, H., Jin, X., Cheng, X.: Locally adaptive translation for knowledge graph embedding. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI), pp. 992–998 (2016)
14. Krompaß, D., Baier, S., Tresp, V.: Type-constrained representation learning in knowledge graphs. In: Arenas, M., et al. (eds.) ISWC 2015. LNCS, vol. 9366, pp. 640–655. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25007-6_37
15. Lao, N., Mitchell, T.M., Cohen, W.W.: Random walk inference and learning in a large scale knowledge base. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 529–539 (2011)
16. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2181–2187 (2015)
17. Long, T., Lowe, R., Cheung, J.C.K., Precup, D.: Leveraging lexical resources for learning entity embeddings in multi-relational data. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), Short Papers, vol. 2 (2016)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26 (NIPS), pp. 3111–3119 (2013)
19. Nickel, M., Jiang, X., Tresp, V.: Reducing the rank in relational factorization models by including observable patterns. In: Advances in Neural Information Processing Systems 27 (NIPS), pp. 1179–1187 (2014)
20. Nickel, M., Rosasco, L., Poggio, T.A.: Holographic embeddings of knowledge graphs. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI), pp. 1955–1961 (2016)
21. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning (ICML), pp. 809–816 (2011)
22. Shi, B., Weninger, T.: ProjE: Embedding projection for knowledge graph completion. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 1236–1242 (2017)
23. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems 26 (NIPS), pp. 926–934 (2013)
24. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33nd International Conference on Machine Learning (ICML), pp. 2071–2080 (2016)
25. Vrandecic, D.: Wikidata: a new platform for collaborative data collection. In: Proceedings of the 21st World Wide Web Conference (WWW), Companion Volume, pp. 1063–1064 (2012)
26. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 1112–1119 (2014)
27. Xiao, H., Huang, M., Meng, L., Zhu, X.: SSP: semantic space projection for knowledge graph embedding with text descriptions. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 3104–3110 (2017)
28. Xiao, H., Huang, M., Zhu, X.: From one point to a manifold: knowledge graph embedding for precise link prediction. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), pp. 1315–1321 (2016)

29. Xiao, H., Huang, M., Zhu, X.: TransG: a generative model for knowledge graph embedding. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers, vol. 1 (2016)
30. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI), pp. 2659–2665 (2016)
31. Xie, R., Liu, Z., Sun, M.: Representation learning of knowledge graphs with hierarchical types. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), pp. 2965–2971 (2016)