electronics

MDPI

Article

TransET: Knowledge Graph Embedding with Entity Types

Peng Wang [1,2,*], Jing Zhou [3], Yuzhang Liu [1] and Xingchen Zhou [1]

[1] School of Computer Science and Engineering, Southeast University, Nanjing 211189, China; yuzhangliu@seu.edu (Y.L.); xczhou_2021@seu.edu.cn (X.Z.)
[2] School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China
[3] Nanjing Fiberhome Starrysky Co., Ltd, Nanjing 211100, China; jingzhou0201@gmail.com
* Correspondence: pwang@seu.edu.cn

Abstract: Knowledge graph embedding aims to embed entities and relations into low-dimensional vector spaces. Most existing methods only focus on triple facts in knowledge graphs. In addition, models based on translation or distance measurement cannot fully represent complex relations. As well-constructed prior knowledge, entity types can be employed to learn the representations of entities and relations. In this paper, we propose a novel knowledge graph embedding model named TransET, which takes advantage of entity types to learn more semantic features. More specifically, circle convolution based on the embeddings of entity and entity types is utilized to map head entity and tail entity to type-specific representations, then translation-based score function is used to learn the presentation triples. We evaluated our model on real-world datasets with two benchmark tasks of link prediction and triple classification. Experimental results demonstrate that it outperforms state-of-the-art models in most cases.

Keywords: knowledge graph embedding; entity type; circular convolution

Citation: Wang, P.; Zhou, J.; Liu, Y.; Zhou X. TransET: Knowledge Graph Embedding with Entity Types. *Electronics* **2021**, *10*, 1407. https://doi.org/10.3390/electronics10121407

Academic Editor: Amir H. Gandomi

Received: 17 May 2021
Accepted: 9 June 2021
Published: 11 June 2021

1. Introduction

Knowledge graphs (KG) [1,2] are multi-relational-directed graphs and are built to model structured information about real-world entities and relations. Existing large-scale KGs usually include millions of entities as nodes and thousands of relations as edges. More specifically, factual knowledge in knowledge graphs is usually represented as a symbolic triple $(head\ entity, relation, tail\ entity)$ abridged as (h, r, t), indicating that head entity h and tail entity t are connected by the relation r.

In order to make the knowledge graph essentially computable, knowledge graph embedding is proposed to embed entities and relations into continuous low-dimensional vector spaces. To our knowledge, most existing knowledge graph embedding models only focus on observed triples to learn the embeddings of entities and relations. Specifically, TransH [3], TransR [4], TranSparse [5], etc. propose that the same entity should have different representations in different relations, but they all obscure the reasons for such differences and lack interpretability. In fact, in most KGs there are also type information of entities. Entity types, as well as constructed prior knowledge, can be incorporated to learn more precise semantic expressions. In KGs, most entities are linked to their corresponding concepts by entity types they belong to. In addition, when constructing ontologies, it is common to identify types for the domain and range of relations, respectively, which are used to constrain corresponding head entities and tail entities. These constraints can make triples more expressive in semantics. Figure 1 shows a knowledge graph with type constraints, in which a relation *birthIn* is constrained by the types of domain *person* and range *city*, analogously, relation *nationality* is constrained by *person* and *county*, *cityOf* is constrained by *city* and *country*. It is intuitive that multiple entity type information can be used for enhancing the presentation of knowledge graphs.

Knowledge graphs such as Freebase [6] and DBpedia [7] contain information about entity types. However, there are only a few knowledge graph embedding models incor-

porating entity types. Krompaß et al. [8] model entity types as kinds of constraints in different models: In RESCAL [9], a knowledge graph is divided into sub-graphs based on relations, and the entities in each sub-graph are supposed to belong to the types constrained by the domain and range of specific relation; in TransE [10] and mwNN [1], the types of entities obtained from negative sampling are also constrained by the domain and the range of specific relation. The constraints of entity types can help improve the quality of the embeddings of entities and relations. However, entity types are introduced in the form of latent constraints in the objective functions of different models. It does not reflect explicit association between entities and entity types, and thus each entity shares the same representation in different types. SSE [11] introduces Laplacian Eigenmaps [12] to make entities belong to the same types as closely as possible in the embedding space, but SSE also learns single representations for entities in different types. TKRL [13] solves this problem by using the hierarchical information of entity types to generate mapping matrices, which enable entities to have different representations in different types.

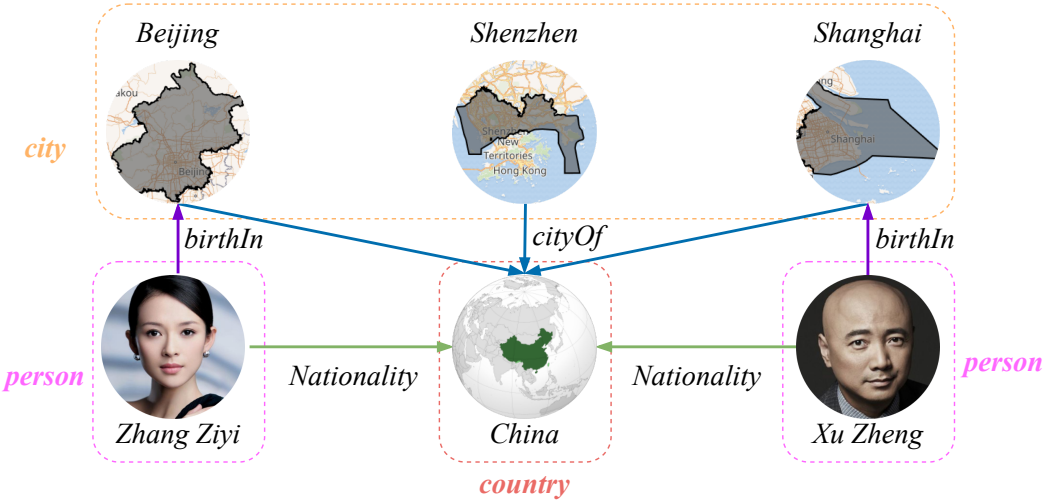


Figure 1. A konwledge graph example with type constraints.

Furthermore, knowledge graphs are multi-relational graphs. Bordes et al. [10] classify relations in knowledge graphs into four types: 1-to-1, 1-to-N, N-to-1, and N-to-N, according to the number of entities linked to relations. The last three are called complex relations. Different entities with the same complex relations tend to have similar embedded representations, which is called a complex relation problem. TransE is one of the typical models with complex relational problems. It constrains that the sum of the embeddings of head entity and relation is close to the embedding of the tail entity, resulting in similar embeddings of entities on the "N" side of a complex relation. For example, in Figure 1, given a complex relation $CityOf$, when the tail entity is *China*, the head entity can be *Beijing*, *Shanghai*, and *Shenzhen*. The embeddings of these three cities learned by TransE will be similar to each other. To overcome a complex relation problem, CTransR [4], TransD [14], and TranSparse [5] use assistant vectors or matrices to capture more complex semantics between relations and entities. In these models, entities are supposed to have different properties in different relations. However, none of the existing models deeply study what causes the difference.

To incorporate entity types to make the embeddings of entities and relations more expressive, this paper proposes a novel knowledge graph embedding model TransET, so that the restriction of head and tail entity types by relations can explain well that entities have different representations in different relations. In TransET, a reasonable explanation is provided for entities having different representations in different relations: The domains and ranges of relations constrain entities with specific types. Specifically, given a triple, circle convolution based on the embeddings of entity and entity types is utilized to map

head entity and tail entity to type-specific representations, then a translation-based score function is applied to learn the presentation of the triple. We evaluate the TransET model on benchmark datasets FB15K (https://deepai.org/dataset/fb15k-237, accessed on 15 March 2019) and DBpedia98K with link prediction and triple classification. Experimental results demonstrate that TransET incorporates entity types allowing entities to have different vector representations under different types. In addition, it significantly outperforms state-of-the-art models in most cases.

In the remainder of this paper, we first illustrate symbols used throughout the paper in Section 3.1. Then detail the proposed TransET model in Section 3. Experiments and results are reported in Section 4. Then in Section 2, we discuss related work, followed by the conclusion and future work in Section 5.

2. Related Work

Translation models view the relation between entities as a translation operation. The classical translation model TransE [10] maps entities and relations into the same semantic space, which is simple and efficient but does not handle complex relations such as 1-to-N, N-to-1, and N-to-N well. To better model complex relations, TransH [3] maps head and tail entities to corresponding relation hyperplane, respectively. TransR/CTransR [4] replaces hyperplanes with relation-based matrices that map head-tail entities into relation space. TransD [14] refines TransR/CtransR by improving the relational mapping matrix into two separate mapping matrices for head and tail entities. A similar approach is TranSparse [5], which takes the complexity of the relationship into account in the design of the relational mapping matrix. Song et al. [15] argue that the loss of the existing translation-based embedding model uses only a pair of positive and negative triples in one update of the learning parameters, which suffers from slow convergence and poor local optimality. Therefore, they proposed the N-pair translation loss that considers multiple negative triples at one update.

To capture potential semantic relationships between entities and relations, semantic matching models using similarity-based scoring functions have emerged one after another. LFM [16] embeds relations in bilinear transformation matrices acting on entities, all of which can be decomposed on the same set of low-ranked matrices. DistMult [17] restricts the above bilinear transformation matrix to diagonal arrays, further simplifying the complexity of the computation. RESCAL [9] represents the set of triples in the knowledge graph as a third-order tensor. HolE [18] introduces circular correlation operations to depict associations between entities.

In fact, there are other rich information in the knowledge graph that can help improve the quality of knowledge representation, such as entity types. Guo et al. argued that entities of the same type should be closer together in the embedding space and proposed SSE [11] to model semantic smoothness using two manifold functions. TKRL [13] uses the hierarchical type information of entities to construct a specific hierarchical category mapping matrix to map head and tail entities to relevant category space via the mapping matrix, respectively, and then uses a translation-based scoring function to measure the trustworthiness of the triples. Noticing that some relations indicate attributes of entities, KR-EAR [19] distinguishes relation types into attributes and relations. Zhang et al. [20] extended existing embedding models to learn knowledge representations with a hierarchical relation structure. MKRL [21] jointly consider triple fact, entity description, hierarchical type, and textual relation information. JECI [22] differentiates the concepts and the instances in the knowledge graph and embeds them jointly using circular convolution based on neighbor information and belonging concept information. To alleviate the problems of low generalizability and high complexity of JECI, JECI++ [23] simplifies the hierarchical concepts and takes the relations into account in neighbor information. Rahman et al. [24] map entities to the embedding space via the entity type matrix. Esma et al. [25] use the implicit type information contained in the dataset to train a joint model on the correctness and type consistency of the triplet. In addition, GREG [26] uses vector representations of

mention-level local relations to construct knowledge graphs that can represent the input document. Since each knowledge graph embedding has its own idiosyncrasy, Su et al. [27] proposed a committee model to form a committee of various knowledge graph embeddings to reflect various perspectives.

The above existing type-enhanced models ignore that the type of entities should be restricted by the corresponding relation. Therefore, in this paper, TransET designs type-based mapping function for each specific relation, and then uses a translation-based method to learn the knowledge representation. It can better explain that entities should have different representations in different relations.

3. TransET Model

3.1. Preliminaries

We first introduce the notations used in this paper. Lowercase letters in italics denote entities, relations, or types, whose bold forms denote embedding vectors.

A knowledge graph is denoted as $\mathcal{KG} = \{\mathcal{E}, \mathcal{R}, \mathcal{S}\}$, where $\mathcal{E}$ and $\mathcal{R}$ denote the set of entities and relations, respectively, and $\mathcal{S} = \{(h, r, t)\}$ represents the set of triples. Each triple (h, r, t) is composed of a head entity h, a tail entity t, and a relation r, where $h, t \in \mathcal{E}$, and $r \in \mathcal{R}$.

In addition, the set of entity types is denoted as $\mathcal{T}$. r_h denotes the type of head entity h in relation to r, and r_t denotes the type of tail entity t in relation to r, where $r_h, r_t \in \mathcal{T}$. The sets of entities belonging to type r_h and r_t are denoted as $\mathcal{E}_{r_h}$ and $\mathcal{E}_{r_t}$, respectively, where $\mathcal{E}_{r_h}, \mathcal{E}_{r_t} \in \mathcal{E}$.

3.2. TransET Model

TransET aims to incorporate entity types to embed entities and relations into a continuous vector space of which entities have different representations in different relations, and at the same time retain good interpretability.

In a triple, specific relation constrains its corresponding head entity and tail entity to specific types. On the basis of translation model and such contraints, this paper proposes a novel knowledge graph embedding model by incorporating entity types, named TransET. In TransET, all entity, relations, and types of entities are embedded to vectors with d dimensions. For example, given a triple (h, r, t), the types of head entity h and tail entity t under relation r are denoted as r_h and r_t, separately. Their embedded vectors are denoted as $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}, \boldsymbol{r}_h, \boldsymbol{r}_t \in \mathbb{R}^d$, separately.

As shown in Figure 2, in TransET, each entity can be represented as different vectors with different types. In order to get the mapped entity vector, the circular roll-up operation sequentially rotates the entity vector in a clockwise direction by several dimensions and then does the inner product operation with the type vector. Type-based mapping functions are designed to obtain the mapped vectors of entities. TransET first utilizes a mapping function to obtain vectors of the head entity h and tail entity t when they belong to specific types, denoted as $\boldsymbol{h}_\perp$ and $\boldsymbol{t}_\perp$, separately. Then, a translation operator is performed on mapped entities and relation, i.e., $\boldsymbol{h}_\perp + \boldsymbol{r} \approx \boldsymbol{t}_\perp$. Equation (1) is designed to evaluate the confidence of the triple, called the scoring function. A lower score indicates that triple (h, r, t) is an existing fact with a larger probability:

$$f(h, r, t) = ||\boldsymbol{h}_\perp + \boldsymbol{r} - \boldsymbol{t}_\perp||_{L1/L2}. \tag{1}$$

To make vectors of entities in specific types more expressive, we formalize the mapping function as a circular convolution based on vectors of entities and vectors of types. Figure 3 shows the procedure of obtaining the mapped vector of head entity $\boldsymbol{h}_\perp$, utilizing the vector of head entity $\boldsymbol{h}$ and vector of type $\boldsymbol{r}_h$. We concatenate the first dimension and the last dimension of vectors to get a reshaped vector in the form of annulus. In which, smaller circles in the same color denote the same dimensions, and the red arrows point to the first dimension of vectors. The i-th calculation, the elements in $\boldsymbol{h}$ are rotated i positions clockwise, which then is conducted as an element-wise product with $\boldsymbol{r}_h$, resulting in the

i-th element of $\boldsymbol{h}_\perp$, denoted as $[\boldsymbol{h}_\perp]_i$, formalized as Equation (2). The mapping function uses the circular convolution, which makes the type vector and entity vector communicate sufficiently to obtain feature associations between corresponding dimensions as well as non-corresponding dimensions:

$$[\boldsymbol{h}_\perp]_i = \sum_{j=0}^{d-1} [\boldsymbol{h}]_j \cdot [\boldsymbol{r}_h]_{(j+i) \bmod d}, \ i = 0, 1, \cdots, d-1. \tag{2}$$

Similar to most related works, we adopt the margin-based loss function in Equation (3) as the optimized objective:

$$L = \sum_{\xi \in \mathcal{S}} \sum_{\xi' \in \mathcal{S}'} [\gamma + f(\xi) - f(\xi')]_+ \tag{3}$$

$$[x]_+ = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{4}$$

where $[x]_+$ denotes the hinge loss function as in Equation (4). γ is the margin. ξ denotes the positive triple (h, r, t) and ξ' denotes the negative triple (h', r, t) or (h, r, t') by corrupting ξ. $\mathcal{S}'$ denotes the set of negative triples. This loss function includes both scores of positive triples and negative triples. The goal of training is minimizing the loss, which is minimizing the score of positive triples while maximizing the score of negative triples. Moreover, the score of negative triples is supposed to be γ more than the score of its corresponding positive triple.

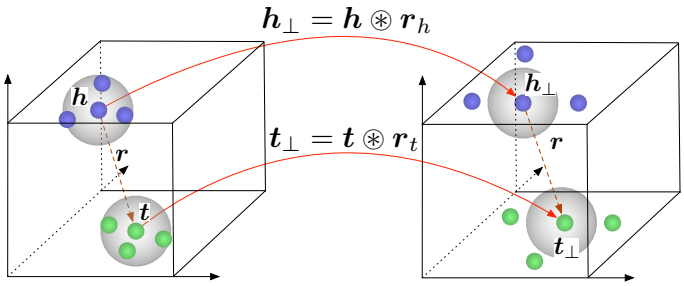


Figure 2. The principle of the TransET model.

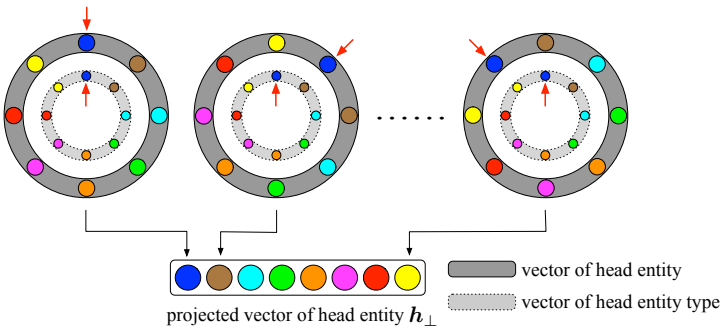


Figure 3. Illustration of the mapping function. The annulus in dark grey represents the vector of head entity, while the annulus in light gray represents the vector of head entity type.

Similar entities tend to be close to each other in the embedding space. In order to tackle this issue, Krompaß et al. [8] propose that the types of corrupting entities in negative triples are supposed to be the same as the types of corrupted entities in positive triples. For example, if (h', r, t) is generated by corrupting (h, r, t), $r_{h'} = r_h$. Such type constraint increases the difference between the entities belonging to the same type, which can help alleviate the problem of complex relations. However, such a hard type-constraint may

destroy the natural similarity between these entities. In TKRL [13], Xie et al. set a hyper-parameter relevant to the number of all entities and the entities of the same types, and utilize this hyper-parameter as the probability of corrupting positive triples with the entities whose types are the same as the corrupted entities. In TransET, this probability is denoted as ρ, an independent hyper-parameter. For example, given a triple (*Beijing*, *City Of China*), the type of head entity is constrained to be *City*. When generating a negative triple by replacing *Beijing*, we randomly choose an entity belonging to *City* with the probability ρ, e.g., *Shanghai*. We randomly choose an entity not belonging to *City* with the probability $1-\rho$, e.g., *Shakespeare*. Such a soft type-constraint can increase the difference between the entities belonging to the same type, and keep some similarity between these entities.

4. Experiments

4.1. Datasets

We adopt the following two datasets, FB15K and DBpedia98K, to evaluate the TransET model. Detailed statistics of these two datasets are shown in Table 1.

- FB15K: FB15K is a widely-used dataset in knowledge graph embedding, which contains 14,951 entities, 1345 relations, and 592,213 triples. Following Bordes et al. [10], we split the triples into the training set, validation set, and test set. The entity types are collected and processed by Xie et al. [13] through $type/instance$, $rdf\text{-}schema\#domain$, and $rdf\text{-}schema\#range$ domains in Freebase;
- DBpedia98K: DBpedia98K consisting of 98,022 entities, 294 relations, and 596,797 triples is a novel dataset we built by the following steps: (1) Randomly select some relational triples from DBpedia; (2) collect the entity types through the $rdfs:domain$ and $rdfs:range$ domains of relations contained in the triples selected; and (3) split the selected triples into thetraining set, validation set, and test set.

Table 1. Statistics of FB15K and DBpedia98K.

Dataset	FB15K	DBpedia98K
#Entity	14,951	98,022
#Relation	1345	294
#Entity Type	571	91
#Training Set	483,142	530,797
#Validation Set	50,000	60,000
#Test Set	59,071	60,000

4.2. Link Prediction

4.2.1. Design of Experiments

Link prediction is to predict the missing head entity or tail entity for an incomplete triple based on the learned embeddings of entities and relations. For each testing triple (h,r,t), we adopt the method proposed in [28] to replace h with all entities in $\mathcal{E}$ to construct negative triples. Then we use the scoring function in Equation (1) to calculate the scores for (h,r,t) and all restructured triples. After ranking these triples in ascending order based on the scores, we can get the rank of (h,r,t). The same operations are performed for t, and we can get the other rank of (h,r,t).

Following most previous works, we adopt MR (the mean of the sort positions of all correct triples) and Hits@10 (the proportion of valid test triples ranking at the top 10 prediction) metrics. A lower MR and a higher Hits@10 indicate that the embedding model has better performance. Note that, a restructured triple may have already existed in triples set $\mathcal{S}$, that is, it is positive in fact. For example, $(Shakespeare, write, Hamlet)$ is a testing triple, and *Hamlet* is replaced by *Macbeth* when constructing negative triples. Since $(Shakespeare, write, Macbeth)$ is positive in fact, the experimental results of MR will go up and Hits@10 will go down. In order to eliminate such a negative impact on evaluation, we adopt a filtering method proposed by Bordes et al. in TransE [10], that is filtering the

false negative triples from candidates triples before ranking, then we get filtered results, called "Filter" to compare with the previous results called "Raw". Hits@10 comes from the results of "Filter".

In this task, we use FB15K and DBpedia98K to train TransET model for 1000 epochs and evaluation. We adopt SGD (Stochastic Gradient Descent) to minimize the loss function. The validation set is utilized to select the learning rate η for SGD among {0.0002, 0.0003, 0.0004, 0.0005, 0.001}, the dimension of embeddings d among $\{20, 50, 100\}$, margin γ among $\{0.1, 0.3, 0.5, 1, 2\}$, and probability for soft type-constraint ρ among {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}. The optimal hyperparameters are $\eta = 0.0002$, $d = 100$, $\gamma = 2$, and $\rho = 0.3$ on FB15K, $\eta = 0.0003$, $d = 100$, $\gamma = 2$, and $\rho = 0.3$ on DBpedia98K. We choose L2 distance as the form of Equation (1).

4.2.2. Analysis of results

We choose TransE, TransH, TransR, TransD, KR-EAR [19], and TKRL as the baselines. Apart from embeddings of entities and relations, TransET has extra parameters to learn. Table 2 shows the comparison of TransET and baselines in space complexity and time complexity, where N denotes the total number and d denotes the dimension of vectors. Subscripts e, r, and t denote entity, relation, and entity type separately. TransET incorporates entity types to better learn the embeddings of entities and relations. Moreover, the number of entity types is less than the number of entities in general, that is $N_t < N_e$. Thus, TransET has the same order of magnitudes with TransE and TransH.

Table 2. Comparison of models in space complexity and time complexity.

Model	Space Complexity	Time Complexity
TransE	$O(d_eN_e + d_rN_r)(d_e = d_r)$	$O(d_e)$
TransH	$O(d_eN_e + 2d_rN_r)(d_e = d_r)$	$O(d_e)$
TransR	$O(d_eN_e + (d_ed_r + d_r)N_r)$	$O(d_ed_r)$
TransD	$O(2d_eN_e + 2d_rN_r)$	$O(max(d_e, d_r))$
KR-EAR	$O(d_eN_e + d_r^2N_r + d_t^2N_t)(d_e = d_r = d_t)$	$O(d_e^2)$
TKRL	$O(d_e^2N_e + d_rN_r + d_t^2N_t)(d_e = d_r = d_t)$	$O(d_e^2)$
TransET	$O(d_eN_e + d_rN_r + d_tN_t)(d_e = d_r = d_t)$	$O(d_e^2)$

Table 3 shows the experimental results of link prediction. On FB15K and DBpedia98K, TransET outperforms all baselines in most cases, which indicates that the entity types incorporated by TransET contain abundant semantic information, making the embeddings of entities and relations more expressive. Both TransET and TransH incorporate assistant embeddings apart from embeddings of entities and relations, however, TransET outperforms TransH. The possible reasons may cover the following aspects:

- The semantics of assistant embeddings: The assistant embeddings TransH represent the normal vectors of the relation-specific hyperplane, while in TransET represent the embeddings of entity types. The latter clarifies the semantics of the assistant embeddings, and is more targeted than the former. Thus it can better capture more precise semantic connections between entities and relations;
- The forms of mapping functions: The mapping function TransH is designed in the form of vector multiplication, while in TransET is designed in the form of circle convolution. Compared to vector multiplication, circle convolution can more directly capture the semantics between different dimensions of different vectors. For example, as shown in Figure 3, the first dimension of $\boldsymbol{h}_\perp$ is the sum of the products of the same dimension of $\boldsymbol{h}$ and $\boldsymbol{r}_h$, i.e., $[\boldsymbol{h}_\perp]_0 = \sum_{i=0}^{d-1} [\boldsymbol{h}]_i \cdot [\boldsymbol{r}_h]_i$.

Table 3. Experimental results of link prediction.

Dataset	FB15K				DBpedia98K			
Metric Setting	MR		Hits@10(%)		MR		Hits@10(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	243	125	34.9	47.1	262	148	30.2	45.1
TransH	211	84	42.5	58.5	258	119	35.6	53.4
TransR	226	78	43.8	65.5	235	83	38.8	55.6
TransD	211	67	49.4	74.2	217	65	44.1	60.4
KR-EAR	**172**	118	39.5	57.3	213	71	33.5	48.9
TKRL	202	87	**50.3**	73.4	209	63	46.4	63.4
TransET	187	**62**	50.1	**76.4**	**194**	**57**	**53.0**	**70.2**

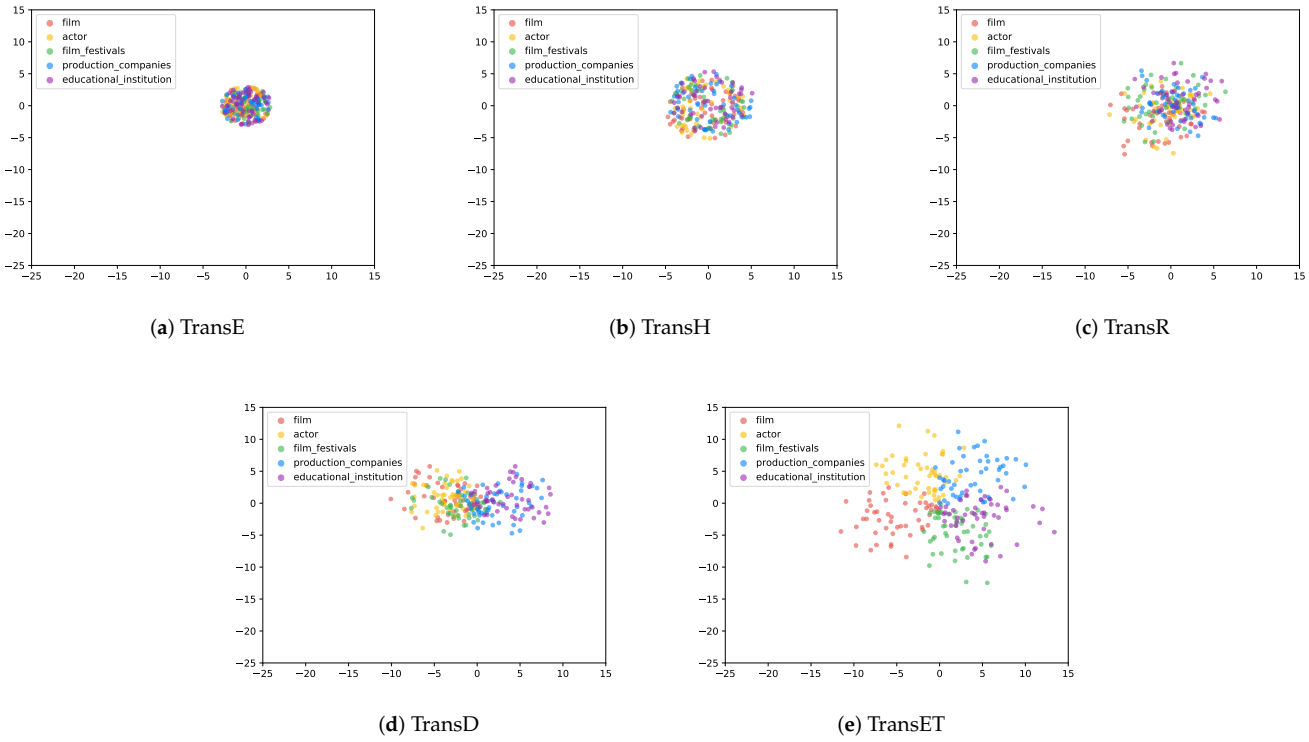
Futhermore, compared to other baselines, TransET improves the quality of embeddings of entities and relations when it has less space complexity, which highlights the superiority of TransET.

Considering the fact that TransE, TransH, TransD, and TransH are able to process complex relation problems, we conduct a further experiment to investigate the performance of TransET on four categories (i.e., 1-to-1, 1-to-N, N-to-1, and N-to-N). The dataset of relational categories is built by the approach of Bordes et al. [10]. Table 4 shows the filtered Hits@10 scores of link prediction on four types of triples based on relations. From the overall view, TransET outperforms in all cases. Especially when predicting the "1" side in 1-to-N and N-to-1 relations, TransET is 6.8% and 5.2% higher than the best baseline TransD, separately. It indicates that TransET can distinguish relevant entities and irrelevant entities more precisely. TransET also improves when predicting the "N" side in 1-to-N, N-to-1, and N-to-N relations. These experimental verifies the effectiveness of TransET in solving the complex relation problem.

Figure 4 shows the 2D visualization for the embeddings of some tail entities with the same relation */education/educational_degree/people_with_this_degree./education /education/institution* and the same head entity */m/0bkj*86 in FB15K. We project these embeddings into two dimensions using t-SNE [29]. From the figures, we can see that TransET can learn similar embedding representations for entities of the same type, while learning greater distances between representations of entities of the different types. In addition, the relative distance between these entities of TransET are obviously larger than these of baselines, which indicates that similar entities can be more easily distinguished through TransET. This is consistent with the phenomenon that TransET outperforms all baselines when predicting the tail entities in 1-to-N relations.

Table 4. Filtered Hits@10 score of link prediction on four types of triples (%).

Task	Predicting Head Entity				Predicting Tail Entity			
Type	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
TransE	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransR	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
TransD	80.7	85.8	47.1	75.6	80.0	54.5	80.7	77.9
TransET	**82.4**	**92.6**	**50.0**	**75.9**	**83.2**	**55.8**	**88.3**	**80.1**



(**a**) TransE (**b**) TransH (**c**) TransR

(**d**) TransD (**e**) TransET

Figure 4. 2D visualization for the embeddings of some tail entities with the same relation and the same head entity.

4.3. Triple Classification

Design of Experiments

Triple classification is to judge whether a given triple is correct or not. This is essentially a binary classification task, and following most related work, we use accuracy as the metric. With the exception of the hyperparameters mentioned in Section 4.2, we set a threshold δ_r for each relation r. Given a testing triple (h, r, t), if the score calculated by Equation (1) is lower than δ_r, it will be classified as a positive one, otherwise negative.

In this task, we also use the FB15K and DBpedia98K to train TransET model for 1000 epochs and evaluation, and also adopt SGD to minimize the loss function. The range of hyperparameters are the same with link prediction, and the optimal value are obtained by maximizing the accuracy on the validation set. The optimal hyperparameters are $\eta = 0.0003$, $d = 100$, $\gamma = 1$, and $\rho = 0.3$ on FB15K, $\eta = 0.0003$, $d = 100$, $\gamma = 1$, and $\rho = 0.3$ on DBpedia98K. We chose L2 distance as the form of Equation (1).

Table 5 shows the experimental results of triple classification. Similar to link prediction, TransET shows an obvious superiority among all testing models, which is consistent with the performance in link prediction. Since the entity types are incorporated and the circular convolution is utilized as the mapping function, the semantics of embeddings of entities and relations are enriched, helping to improve the accuracy of triple classification.

Table 5. Accuracy of triple classification (%).

Dataset	FB15K	DBpedia98K
TransE	77.6	70.3
TransH	77.1	69.2
TransR	79.8	74.5
TransD	84.2	78.8
KR-EAR	85.7	77.4
TKRL	88.5	80.2
TransET	**94.5**	**88.6**

5. Conclusions

In this paper, we proposed a novel knowledge graph embedding model called TransET. TransET incorporates entity types and utilizes type-based circle convolution functions to map entities. Thus type-specific representations of entities can be obtained. Then the translation model is used to learn the structure and semantics of mapped entities and relations. The experimental results showed that TransET improved the performance of link prediction and triple classification in all cases. Furthermore, entities have different embeddings in different types, which enrichs the knowledge representation.

In future, we will explore the following research. First, when learning the embeddings of entities and relations in knowledge graphs, we could incorporate more information to make embeddings more expressive. For example, entities and relations about time may be changed as time passed. Incorporating more information could make knowledge graphs more dynamic. Secondly, we coould design a general framework or algorithm for better incorporating kinds of information to lower the time and space complexities.

Author Contributions: Conceptualization, P.W. and J.Z.; methodology, P.W. and J.Z.; validation, Y.L.; writing—original draft preparation, J.Z. and Y.L.; writing—review and editing, P.W. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The work is supported by the National Key R&D Program of China (2018YFD1100302), the National Natural Science Foundation of China (No. 61972082), and All-Army Common Information System Equipment Pre-Research Project (No. 31511110310, No. 31514020501, No. 31514020503).

Data Availability Statement: FB15K dataset is available at https://www.microsoft.com/en-us/download/details.aspx?id=52312, accessed on 5 January 2021 and DBpedia98K dataset is available at https://github.com/liuyuzhangolvz/DBpedia98K, accessed on 17 January 2021.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xin, D.; Evgeniy, G.; Geremy, H.; Wilko, H.; Ni, L.; Kevin, M.; Thomas, S.; Shaohua, S.; Wei, Z. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In Proceedings of the 20th ACM SIGKDD International Conference, New York, NY, USA, 24 August 2014; pp. 601–610.
2. Aidan, H.; Eva, B.; Michael, C.; Claudia, D.; Gerard, D.; Claudio, G.; Labra, G.J.E.; Sabrina, K.; Sebastian, N.; Axel, P.; et al. Knowledge graphs. *arXiv* **2020**, arXiv:2003.02320.
3. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the 28th AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; pp. 1112–1119.
4. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2181–2187.
5. Ji, G.; Liu, K.; He, S.; Zhao, J. Knowledge graph completion with adaptive sparse transfer matrix. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, AZ USA, 12–17 February 2016.
6. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 27th ACM SIGMOD International Conference, Vancouver, BC, Canada, 9–12 June 2008.
7. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. Dbpedia: A nucleus for a web of open data. In Proceedings of the 6th International Semantic Web Conference, Busan, Korea, 11–15 November 2007.
8. Krompaß, D.; Baier, S.; Tresp, V. Type-constrained representation learning in knowledge gaphs. In Proceedings of the 14th International Semantic Web Conference, Bethlehem, PA, USA, 11–15 October 2015; pp. 640–655.
9. 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, Bellevue, WA, USA, 28 June–2 July 2011; pp. 809–816.
10. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Proceedings of the 27th Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 2787–2795.
11. Guo, S.; Wang, Q.; Wang, B.; Wang, L.; Guo, L. Semantically smooth knowledge graph embedding. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, China, 26–31 July 2015; pp. 84–94.
12. Belkin, M.; Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In Proceedings of the 15th Neural Information Processing Systems, Vancouver, BC, Canada, 9–14 December 2001; pp. 585–591.
13. Zhang, Z.; Zhuang, F.; Qu, M. Knowledge Graph embedding with hierarchical relation structure. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 3198–3207.

14. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, China, 26–31 July 2015; Volume 1, pp. 687–696.
15. Song, H.; Kim, A.; Park, S. Learning translation-based knowledge graph embeddings by N-pair translation loss. *J. Appl. Sci.* **2020**, *10*, 3964. [CrossRef]
16. Jenatton, R.; Le Roux, N.; Bordes, A.; Obozinski, G. A latent factor model for highly multi-relational data. In Proceedings of the 26th Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 3176–3184.
17. Yang, B.; Yih, W.T.; He, X.; Gao, J.; Deng, L. Embedding entities and relations for learning and inference in knowledge bases. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
18. Nickel, M.; Rosasco, L.; Poggio, T.A. Holographic embeddings of knowledge graphs. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 1955–1961.
19. Lin, Y.; Liu, Z.; Sun, M. Knowledge representation learning with entities, attributes and relations. In Proceedings of the 25th International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 2866–2872.
20. Xie, R.; Liu, Z.; Sun, M. Representation learning of knowledge graphs with hierarchical types. In Proceedings of the 25th International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 2965–2971.
21. Tang, X.; Chen, L.; Cui, J. Knowledge representation learning with entity descriptions, hierarchical types, and textual relations. *J. Inf. Process. Manag.* **2019**, *56*, 809–822. [CrossRef]
22. Zhou, J.; Wang, P.; Pan, Z.; Xu, Z. JECI: A joint knowledge graph embedding model for concepts and instances. In Proceedings of the 9th JIST Conference, Hangzhou, China, 25–27 November 2019; pp. 82–98.
23. Zhou, J.; Wang, P. JECI++: A modified joint knowledge graph embedding model for concepts and instances. *J. Big Data Res.* **2020**, *24*, 100160.
24. Rahman, M.; Takasu, A. Knowledge graph embedding via entities' type mapping matrix. In Proceedings of the 25th ICONIP, Siem Reap, Cambodia, 13–16 December 2018.
25. Balkır, E.; Naslidnyk, M.; Palfrey, D.; Mittal, A. Improving knowledge graph embeddings with inferred entity types. In Proceedings of the 32nd Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018.
26. Kim, K.; Hur, Y.; Kim, G.; Lim, H. GREG: A global level relation extraction with knowledge graph embedding. *J. Appl. Sci.* **2020**, *10*, 1181. [CrossRef]
27. Su, C.; Song, H.; Park, S. An approach to knowledge base completion by a committee-based knowledge graph embedding. *J. Appl. Sci.* **2020**, *10*, 2651.
28. Bordes, A.; Weston, J.; Collobert, R.; Bengio, Y. Learning structured embeddings of knowledge bases. In Proceedings of the 25th AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 7–8 August 2011; p. 6.
29. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Res.* **2008**, *9*, 2579–2605.