Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **5**, September **2017**

Knowledge Graph Embedding Translation Based on Constraints

Yunbing Wu[1],Jengshyang Pan[2],Ping Lu[3],Kaibiao Lin[4,*],Xiaoyan Yu[1]

[1]College of Mathematics and Computer Science
Fuzhou University
No.2. Xueyuan Road, University Town, Minhou,Fuzhou City, Fujian Province, P.C 350116 - China

[2]Fujian Provincial Key Lab of Big Data Mining and Applications
Fujian University of Technology
No.3. Xueyuan Road, University Town, Minhou, Fuzhou City, Fujian Province, P.C 350118 - China

[3]Department of Business
Xiamen University of Technology
No.600 Ligong Road, Jimei District, Xiamen, Fujian Province, P.C 361024 - China

[4]School of Computer and Information Engineering
Xiamen University of Technology
No.600 Ligong Road, Jimei District, Xiamen, Fujian Province, P.C 361024 - China
*Corresponding author: kblin@xmut.edu.cn

Received April, 2017; revised July, 2017

Abstract. *Knowledge graph reasoning is discovering new entity relations by computing and inference from existing relations. However, most reasoning models of translation embedding-based knowledge graphs have not considered the semantic-type constraints of relations in the construction of corrupted triplets. Hence, the constructed corrupted triplets may not conform to the actual semantic information and may, thus, significantly affect the prediction accuracy of the model. Therefore, we propose a constraint-based embedding model in this paper. First, the model establishes the head and tail entity set for each relationship. Then, it ensures that both the replacing head and tail entities in the corrupted triplet belong to the respective entity set so that the corrupted triplets that do not conform to the responding semantic relations are excluded. To evaluate the proposed model, we conduct link prediction and triple classification on WordNet and Freebase databases. The experimental results show that our method remarkably improves the performance compared to several state-of-the-art baselines.*
Keywords: Knowledge graph; Translation embedding; Linking predication; Triple classification

---

1. **Introduction.** With the advent of big data, knowledge graphs [1], which constitute a new knowledge representation method and data management model, have emerged as useful for natural language processing, question answering, information retrieval and other related artificial intelligence (AI) tasks. Knowledge is represented by a directed graph in the knowledge graph, where the nodes represent entities or concepts, and the directed edges represent the relations. The triplets (*head, relation, tail*) are used to describe facts. Among them, *head* and *tail* represent the head entity and tail entity, respectively, and *relation* is the relationship between the head entity and the tail entity. These triplets

are abbreviated as h, r and t, respectively. For example, the fact "*Barack Obama is the president of the United States*" is expressed by the triplet ($Barack\ Obama$, $presidentOf$, USA), where $Barack\ Obama$ is the head entity, USA is the tail entity, and $presidentOf$ is the relation between the head and tail entities. The currently established large-scale general knowledge graphs include WordNet [2]and Freebase [3], among others. Additionally, some companies have built their own knowledge graphs. For example, Google built the Google knowledge graph to improve information retrieval [4], and Microsoft established Knowledge Vault [5] to support its search engine, Satori.

Because of the restriction of the construction time and emergence of huge amounts of new facts, knowledge graphs may have imperfections. Therefore, they must be completed via knowledge graph learning and reasoning. However, the traditional reasoning algorithms of knowledge graphs do not work well in large-scale knowledge graph reasoning because of the graph structure and its poor portability. The common practice is to embed a high-dimensional knowledge graph into a low-dimensional continuous vector space [6, 7, 8, 9, 10, 11, 12] and construct an inference model by minimizing the loss function among all entities and relations. One of the most representative algorithms is the TransE model proposed by Bordes et al [9] in 2013. The TransE model is simple and has high prediction accuracy. However, it also has inadequacies for data with one-to-many, many-to-one, or many-to-many relationships. Therefore, ZhenWang et al [13], Yankai Lin et al [14] and Guoliang Ji et al [15]presented TransH, TransR and TransD to improve TransE, respectively. However, most reasoning models of translation embedding-based knowledge graphs do not consider the semantic-type constraints of relations among the entities. As a result, the constructed corrupted triplets may not conform to the actual semantic information, which can significantly affect the prediction accuracy. For example, the head entity of the relation "*born in*" usually refers to a person or animal, and the tail entity usually refers to a place or time. Thus, if the head entity or tail entity of the relation "*born in*" is beyond the above scope, the constructed triplet will be meaningless.

Inspired by TRESCAL, which was proposed by Kai-Wei Chang et al. [16],we introduce a constraint-based embedding reasoning model named TransC. The basic idea of this model is predefining the head entity set and tail entity set for each relation to ensure that the constructed corrupted triplets are consistent with the semantic relation type. Compared to TransE, TransC maintains the advantage of simple parameters and exhibits improved the prediction accuracy.

The contributions of this paper are summarized as follows:

(1)We extract the relational semantic-type constraints from a knowledge base and use them to ensure consistent constructed triplets with practical semantic meaning. By reducing meaningless triplets without semantic relations, the predicting accuracy is improved.

(2)TransC embeds the entities and relations into the same vector. It does not increase the parameter complexity of the model, and thus, the efficiency is maintained.

(3)The link prediction and triplet classification experimental results show that TransC outperforms other translation embedding-based models (e.g., TransE, TransH, TransR and TransD) in terms of its prediction accuracy.

The remainder of the paper is organized as follows: We discuss some related works in Section 2. Then, we describe our model, algorithm and model complexity in Section 3, and we present the link prediction and triplet classification experiments on two real-world knowledge graph in Section 4. Finally, we draw our conclusions and briefly describe future work directions in Section 5.

2. **Related Works.** With the construction and application of various large-scale knowledge graphs in the last few years, many knowledge graph reasoning algorithms have appeared. These algorithms can be summarized as follows. The first category includes algorithms based on tensor decomposition, which is represented by the third-order tensor decomposition algorithm RESCAL [17] proposed by Maximilian Nickel et al. in 2011. Those algorithms reduce the computational complexity using a matrix decomposition technique to compress the dimension of the knowledge graph. The second category includes learning and reasoning methods based on embedding translation and are represented by TransE [9], which was proposed by Bordes et al. in 2013. They embed the entities and relations of a high-dimensional knowledge graph in continuous low-dimensional vector spaces and then use algebraic and geometrical structures to reason in this low-dimensional vector space. The third category is the learning and reasoning method based on paths, such as the path-ranking algorithm introduced by Ni Lao et al. [18] in 2010 for information retrieval. This algorithm treats each distinct relation path in terms of its one-dimensional characteristics. Then, a large number of relation paths are used to build the feature vector for the relation classification, and a relation classifier is established to extract the links and solve the problem of path inference in information retrieval. Thus, it can effectively overcome the path length restriction of random walk. Other approaches have also been developed, such as reasoning with neural tensor networks [11], reasoning with semantic and logical rules [19, 20], and reasoning with text data bases [5, 21, 22, 23]. Among all of these algorithms, TransE proposed by Bordes et al. is simple and has the highest prediction accuracy. It is also easy to extend to large-scale knowledge graph learning and reasoning, and thus, it is very popular. This paper focuses on improving TransE to achieve TransC. We will briefly introduce the related works addressing TransE, which mainly include learning and reasoning methods based on translation and other learning and reasoning methods.

2.1. **Learning and Reasoning Methods based on Translation.** The most representative and popular model is TransE, which was proposed by Bordes et al. in 2013 [9]. This model embeds the entities and relations of a high-dimensional knowledge graph in continuous low-dimensional vector spaces. Then, algebraic and geometrical structures can be used to extract the links among the entities and relations in this low-dimensional vector space to complete the knowledge graph. The entire process is shown in Figure 1. The left part of Figure 1 presents the knowledge graph, and the right part shows the low-dimension vector space. If there is a triplet (*head*,*relation*,*tail*) in a knowledge graph, *head* and *tail* represent the head entity and tail entity of the triplet, respectively, and *relation* represents the relationship between the head and tail entities. Then, the embedded low-dimensional vectors should satisfy the following relationships: $\boldsymbol{h}+\boldsymbol{r}\approx\boldsymbol{t}$, where $\boldsymbol{h}$, $\boldsymbol{t}$ and $\boldsymbol{r}$ are the embedded low-dimensional vectors from the entity head, tail and relationship relation of the high-dimensional data, respectively. A smaller distance in formula 1 indicates that the triplet is more reasonable; otherwise, the triplet is unreasonable.

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

Because of its simplicity and few parameters, TransE can be used to learn and reason in a massive knowledge graph and to predict links and execute entity resolutions, thereby improving the knowledge graph. However, the entity and relation embedding into a low-dimensional space in TransE does not consider the links among entities and relations; therefore, it is more suitable for cases with only mono-relation or one-to-one relationships among the entities. This model is inadequate for data with one-to-many, many-to-one, and many-to-many relationships. Considering the obvious weakness of the TransE model,

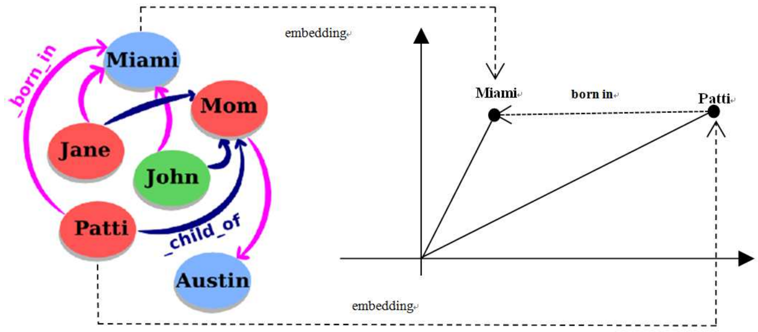

FIGURE 1. Translations of the low-dimensional embedding of the entities from a knowledge graph

ZhenWang et al [13]. designed the TransH model in 2014. Those authors believed that different representation forms should be used for different relations for each entity. Thus, different hyper planes are set for different relations in the embedded low-dimensional space, and the entities are projected onto the corresponding relational hyperplane to overcome the limitations of TransE. Hence,$\boldsymbol{h}_{\perp} = \boldsymbol{h} - w_r^T \boldsymbol{h} w_r$.The distance function of TransH is shown in formula 2.

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

Although TransH sets up different hyperplanes for the corresponding relations and improves the prediction accuracy for one-to-many, many-to-one, and many-to-many relationships, the TransE and TransH models simply embed both entities and relations in the same semantic space. In the real world, an entity may have multiple aspects, and various relations may focus on different aspects of the entities. Consequently, Yankai Lin et al. [14] built the TransR model in 2015. This model embeds the entities of a knowledge graph into an entity space and embeds the relations into a corresponding relation space. Then, the embedded entities are projected into the corresponding relation space for the distance calculation. Namely, $\boldsymbol{h}_r = \boldsymbol{h} M_r$, and $\boldsymbol{t}_r = \boldsymbol{t} M_r$,where M_r is the projection matrix for both the head and tail entities. The distance function is shown in formula 3.

$$f_r(h,t) = \| \ \boldsymbol{h}_r + \boldsymbol{r} - \boldsymbol{t}_r \|_2^2 \tag{3}$$

2.2. **Other Related Learning and Reasoning Methods.** Many learning and reasoning methods are related to TransE. Here, we list some of the methods used in the experiments described in this paper.

(1) Structured Embedding (SE): The SE model was introduced by Bordes et al. in 2011 [6]. It embeds the head and tail entities into a low-dimension vector space and sets different projection matrices for the head and tail entities according to the relation type. Thus, the distance function of SE is:

$$f_r(h,t) = \| \ \mathbf{M}_{h,r} \boldsymbol{h} - \ \mathbf{M}_{t,r} \boldsymbol{t} \| \tag{4}$$

(2) Semantic Matching Energy (SME): The SME model was introduced by Bordes et al. in 2011 [7]. It sets the weights for the embedding entities and relations and obtains the distance function by Hadamard multiplication of matrices, as shown in formula 5.

$$f_r(h,t) = (M_1 \boldsymbol{h} + M_2 \boldsymbol{r} + b_1)^T (M_3 \boldsymbol{t} + M_4 \boldsymbol{r} + b_2) \tag{5}$$

(3) Latent Factor Model (LFM): The LFM model was introduced by Jenatton et al. in 2012 [8]. It obtains a second-order correlation using a quadratic structure, as shown in

formula 6.

$$f_r(h,t) = \boldsymbol{h}^T W_r \boldsymbol{t} \tag{6}$$

3. **TransC Model and Algorithm.** The rationality of the relation semantic is not considered in most learning and reasoning models, such as TransE [9],TransH [13],TransR [14], and TransD [15]. This omission leads to substantial unreasonable triplets and significantly affects the prediction accuracy of the model. For example, the training model of TransE removes the head or tail entities of the triplets and randomly structures the corrupted triplets to avoid overfitting of the training model. No constraint exists in the construction of the corrupted triplets, and thus, the corrupted triplets may not conform to the actual semantic type. Inspired by the literature [16], we propose the TransC model. This model establishes the head entity set and tail entity set according to the relation semantic type and adds those constraints to the construction of corrupted triplets to satisfy the relation semantic constraint. For a triplet (h,r,t), head h and tail t should satisfy the relation semantic constraint of relation r. Namely, if r is the relation "*born in*", the head entity usually refers to a person or an animal, and the tail entity usually refers to a place or a time.

We define the following symbols for ease of presentation:E denotes the entire entity set, R is the entire relation set, E_{hr} (or E_{tr}) is the head (or tail) entity set that satisfies the semantic type of relation r, S is the training triplet set, S' is the corrupted triplet set, and S_r is the triplet set that satisfies the semantic type of relation r.

3.1. **TransC model.** The TransE model embeds entities and relations into a k-dimensional vector space and calculates the entity relations in that vector space to calculate the relation among the entities of the knowledge graph. The distance function is shown in formula 1. However, the TransE model does not consider the relation semantic-type constraints, and thus, the constructed corrupted triplets be inconsistent with the relation semantic type. For example, the head entity of the relation "*born in*" usually refers to a person or an animal, and the tail entity usually refers to a place or a time; if this is not the case, the corrupted triplet may not conform to the relation semantic type. The TransC model proposed in this paper can avoid this problem by adding relation semantic-type constraints to the triplet-constructing process. The distance function of TransC is shown in formula 7; the constraints $e_{hr} \in E_{hr} \subset E$ and $e_{tr} \in E_{tr} \subset E$ indicate that the head entity and tail entity should satisfy the semantic-type constraints of relation r.

$$f_r(h,t) = -\| \ \boldsymbol{e}_{hr} + \boldsymbol{r} - \ \boldsymbol{e}_{tr}\|^2 \tag{7}$$

In formula 7, the value of the distance function $f_r(h,t)$ for the triplet is small if the distance e_{hr}+r in the vector space is close to e_{tr}; i.e., the probability of the existence of the triplet (e_{hr},r,e_{tr}) may be high if $f_r(h,t)$ is small. Therefore, formula 7 can be used to infer the missing triplet in the link prediction experiment. In other words, the value of distance function $f_r(h,t)$ for a triplet that satisfies the semantic-type constraints should be lower than that of a triplet that does not satisfy the semantic-type constraints. Because the training dataset of TransC comprises the triplets in the knowledge graph, which can make the training model of TransC overfitting or not robust, we construct the corrupted triplet set, which does not belong to the training triplet set, for each training triplet in the training process to add robustness to the model. Additionally, to allow the training model to distinguish between correct and incorrect triplets, we use the margin-based boundary-adjusting machine learning method, add constraints to the corrupted triplet-constructing process to ensure that they satisfy the relation semantic type, and apply the stochastic gradient descent (SGD) to converge the training model as soon as possible during training model optimization. The optimization model of TransC is shown in formula 8.

$$L = min \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S_r} max(0, \gamma + f_r(h,t) - f_r(h',t')) \tag{8}$$

Formula 8 is subjected to the following constraint conditions:

$$s.t \begin{cases} \gamma > 0 \\ \forall h', t'; h' \in E_{hr} \wedge t' \in E_{tr}, E_{hr} \subseteq E, E_{tr} \subseteq E \\ \forall r \in R, (h', r, t) \in S_r \wedge (h, r, t') \in S_r \\ S_r \subseteq S' \\ \forall h, r, t, ||h|| \leq 1, ||r|| \leq 1, ||t|| \leq 1 \end{cases} \tag{9}$$

By adding the above constraint conditions into the training process, we avoid the construction of meaningless corrupted triplets and improve the accuracy of the training model. γ >0 is a parameter used to adjust the boundary of the training model.

3.2. **TransC Algorithm.** The main goal of TransC is the addition of relation semantic-type constraints to the randomly corrupted triplets of the TransE model to exclude meaningless corrupted triplets, which affect the prediction accuracy. Algorithm 1 presents the detailed learning algorithm. We initialize all embeddings for the entities and predicate relations with the random procedure proposed in [9] and [24]. We also normalize all embedding vectors of the entities and relations. Then, we extract all head and tail entities that satisfy the semantic-type constraints of relation r to E_{hr} and E_{tr}, respectively, and iterate the following procedure. First, we sample a small set of triplets from the training set as the training triplets of the minibatch. Then, we sample a corrupted triplet for each training triplet. For each corrupted triplet, when the head entity and the relation are fixed, the tail entity should come from E_{tr}; for the opposite case, the head entity should come from E_{hr} when the relation and tail entity are fixed. In other words, the corrupted triplets should not randomly sample; instead, they must conform to the semantic-type constraints of relation r. Finally, the parameters of TransC are updated by taking a gradient step with a constant learning rate. The algorithm is stopped based on its performance on a validation set.

The algorithm of TransC is introduced as follows.

Algorithm 1: Learning Algorithm of TransC

Input: Training triple set S, entity set E, relation set R,E_{hr} ,E_{tr} ,S_r,dimension k, γ, ϵ
/* initialization */
while $r \in R$
    $r \leftarrow$Uniform((-6)/$\sqrt{k}$,6/$\sqrt{k}$)   //embedding each relation
    $r \leftarrow r/\ ||r||$ //standardization of each embedded relation
end while
while $e \in E$
    $e \leftarrow$Uniform((-6)/$\sqrt{k}$,6/$\sqrt{k}$)   // embedding each entity
    $e \leftarrow e/\ ||e||$ // standardization of every embedded entity
end while
while ($r \in R$ and $e \in E$)
    if e is the head entity of r
        $E_{hr} \leftarrow e$
    if e is the tail entity of r
        $E_{tr} \leftarrow e$

end while
/*model training */
while less than the maximum number of iterations and energy loss $> \epsilon$
    $S_{batch} \in sample(S, m)$
    $T_{batch} \in \Phi$
    for each (h,r,t) $\in S_{batch}$ do // construct training model for each triple in S
        $h' \in E_{hr}$
        $t' \in E_{tr}$
        (h' , r , t') $\in S_r$
        (h' , r , t') $\leftarrow sample(S_{batch})$ //sample a corrupted triple, and if it does not belong to training set
        $T_{batch} \leftarrow T_{batch} \cup$ (h,r,t),(h' , r , t')
        if $\gamma + f_r(h,t) - f_r(h',t') \geq 0$ then //boundary of the training model does not meet the requirements
            updating $\sum \max(0, (\gamma + f_r(h,t) - f_r(h',t')))$ // optimize training model by SGD
        end if
    end for
end while
Output: Entity set and relation set

3.3. **Comparisons of the Parameter Complexity of the Algorithms.** To analyze the efficiency of the TransC model, we compared the parameter complexities of several typical methods, as shown in Table 1. n_e and n_r denote the number of entities and number of relations in the knowledge graph, respectively, and k is the dimension of the embedding vector space. As shown in Table 1, TransC has the same parameter complexity as TransE and TransH but lower parameter complexity than SE, SME, LFM, and TransR. SE embeds relationships into two matrices using two different projection matrices for the head and tail, which increases the computational complexity compared to TransC. The SME model captures the correlations among the entities and relations via multiple matrix products and Hadamard product; it introduces four different projection matrices, and the matrix operations increase the computational complexity. The LFM model uses second-order correlations among the entity embeddings in a quadratic form. TransR builds the entity and relation embeddings in separate entity spaces and relation spaces. It learns the embeddings by first projecting entities from the entity space to the corresponding relation space and subsequently builds the translations among the projected entities. Because TransR adds the projection transformation processes in different spaces, the complexity of the model parameters increases, which affects the running speed of the model. Therefore, the TransC model outperformed SE, SME, LFM, and TransR in terms of running time.

4. **Experiment and analysis.**

4.1. **Benchmark Dataset.** To facilitate experimental comparisons with some related works, we conducted extensive experiments on two tasks: Link Prediction and Triplet Classification. The experimental data were obtained from two general knowledge graphs: WordNet [2] and FreeBase [3]. WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each of which expresses a distinct concept. Synsets are interlinked via conceptual-semantic and lexical relations. We used two datasets from WordNetWN18 and WN11which were

Table 1. Comparisons of the parameter complexity of several algorithms

Method	#parameters
SE	$O(n_ek+2n_rk^2)$
SME	$O(n_ek+n_rk+4k^2)$
LFM	$O(n_ek+n_rk^2)$
TransE	$O(n_ek+n_rk)$
TransH	$O(n_ek+2n_rk)$
TransR	$O(n_ek+n_rk^2+n_rk)$
TrasnsD	$O(2n_ek+2n_rk)$
TransC(our)	$O(n_ek+n_rk)$

used in [9, 13, 14, 15]. WN18 contains 18 relation types, and WN11 contains 11 relation types. Freebase is an open database of the world's information with more than 23 million entities. It was built by a global community and is free for everyone to query, contribute to, and build applications on. It provides general facts of the world and contains structured information on millions of topics, such as people, places, music, film, food, science, and historical events. We used two datasets from FreebaseFB15K and FB13which were used in [9, 13, 14, 15]. The statistics of these datasets are listed in Table 2.

Table 2. Statistics of the datasets

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

4.2. **Link Prediction.** Link prediction is an important task in completing a knowledge graph. It predicts the missing entity or relation between two entities in a fragmented triplet. For example, given a fragmented triplet (h,r,?) or (?,r,t), if we aim to determine whether entity h or t has relationship r with entity t or h, we must only calculate the distance between h+r and t. In the experiment, we calculated the value of the score/distance function for each candidate triplet and ranked the candidate triplets by their scores instead of determining the best triplet. The benchmark datasets were WN18 and FB15K. The experimental process and result are presented below.

4.2.1. *Evaluation Protocol.* Here, the evaluation protocol refers to the evaluation protocols of TransE [9], TransH [13], TransR [14] and TransD [15]. For each testing triplet (h, r, t)in training set S, we corrupted it by replacing tail t (or head h) with every entity e of the knowledge graph that satisfied the semantic-type constraints of relation r and calculated a probabilistic score of this corrupted triplet (h,r,e) (or (e,r,t)) with the score function $f_r(h, t)$. By ranking these scores in ascending order, we obtained the rank of the original triplet. Two metrics are used for evaluation: the averaged rank (*Mean Rank*) and the proportion of the testing triplet ranked in the top 10% (*Hits*@10). A lower *Mean Rank* and a higher *Hits*@10 indicate better prediction accuracy. These metrics are indicative but can be awed when some corrupted triplets are actually valid (e.g., from the training set). In this case, such triplets may be ranked above the test triplet, but this problem should not be counted as an error because both triplets are true. To avoid this misleading behavior, we removed from the list of corrupted triplets all triplets that appear in the training, validation or test set (except the test triplet of interest). This step ensures that all of the corrupted triplets do not belong to the dataset. We called the original evaluation set containing the corrupted triplets the raw evaluation set and the set after the removal of the corrupted triplets the filter evaluation set.

4.2.2. *Implementation.* Because some related works used the same datasets, we directly copied the experimental results of several baselines from the literature. In TransC, we attempted several parameter combinations and selected the learning rate λ for the SGD from {0.001, 0.005, 0.01, 0.1}, margin γ from {0.25, 0.5, 0.85, 1.0, 1.5}, latent dimension k from {20, 50, 100, 120}, and batch size B from {120,480,960,1440,4800}. The optimal hyperparameters of TransC for the dataset WN18 are λ=0.001, γ=1.0, k=100, and B=1440; we limited the number of epochs to 1000. For FB15K, the optimal hyper parameters of TransC are λ=0.001, γ=0.85,k=100 and B=960; we limited the number of epochs to 500.

4.2.3. *Experimental Results.* The link prediction experiment can be divided into entity prediction and relation-type prediction. The evaluation results are reported in Table 3 and Table 4. Table 3 shows the entity prediction results for WN18 and FB15K. We observe the following:

(1) In the dataset WN18, in terms of *Mean Rank* and *Hits*@10, our TransC model outperforms all baseline methods. In particular, for *Hits*@10, our method achieved improvements of 3.6% and 10.5% compared to TransE and TransH, respectively.

(2) In the dataset FB15K, our approach shows good prediction performance for *Mean Rank* and *Hits*@10. For *Hits*@10, compared to TransE, TransH and TransR, TransC exhibited improvements of 31.5%, 14.2% and 9.9%, respectively, and a slight improvement (1.3%) compared to TransD.

Table 3. Comparison of the link prediction results

Datasets	WN18				FB15K			
Metric	*Mean rank*		*Hits*@10(%)		*Mean rank*		*Hits*@10(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
SE	1,011	985	68.5	80.5	273	162	28.8	39.8
SME	545	533	65.1	74.1	274	154	30.7	40.8
LFM	469	456	71.4	81.6	283	164	26.0	33.1
TransE	263	251	75.4	89.2	243	125	34.9	47.1
TransH	401	388	73.0	82.3	212	87	45.7	64.4
TransR	238	225	79.8	92.0	198	77	48.2	68.7
TransD	224	212	79.6	92.2	194	91	53.4	77.3
TransC(our)	218	209	80.2	92.8	192	79	54.2	78.6

For the relation-type prediction (e.g., 1-TO-1(1-1), 1-TO-MANY(1-N), MANY-TO-1(N-1), and MANY-TO-MANY(N-N)) we only used the dataset FB15K because the relation types of WN18 were too small. We followed the definition for relation type in [9]. A given relationship is 1-TO-1 if a head can appear with at most one tail, 1-TO-MANY if a head can appear with many tails, MANY-TO-1 if many heads can appear with the same tail, and MANY-TO-MANY if multiple heads can appear with multiple tails. In the FB15K dataset, the number of 1-1 relationships accounted for 24% of the total number of relations, 1-N accounted for 23%, N-1 for 29%, and N-N for 24%. The comparison of the *Hits*@10 values obtained for different types of relations is presented in Table 4:

(1) For head prediction, our TransC model outperforms TransE, TransH and TransR in 1-1/1-N/N-1/N-N. In particular, in comparison with TransE and TransH, our approach

achieved improved *Hits*@10 values by 38.4%, 24.6%, 20.1%, and 24.6% and 20.1%, 2.7%, 9.6%, and 2.7%, respectively. Compared to TransD, our method is less accurate, possibly because TransD creates dynamic mapping matrices for different relationships, and thus, it is more accurate for relation-type predictions.

(2) For tail prediction, our approach outperforms all baseline methods in 1-1, 1-N and N-N. However, for N-1, our TransC model is slightly less accurate than TransR and TransD, probably because FB15K has more N-1 relations than others; in contrast, TransR and TransD embed the entities and relationships into different corresponding spaces, and thus, they produce better prediction results in the N-1 situation.

Table 4. Results obtained for FB15K using different relation categories

Tasks	Predicting Head(*Hits*@10)				Predicting Tail(*Hits*@10)			
Relation category	1-1	1-N	N-1	N-N	1-1	1-N	N-1	N-N
SE	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
TransE	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
TransD	86.1	95.5	39.8	78.5	85.4	50.6	94.4	81.2
TransC(our)	82.1	90.3	38.2	79.3	86.5	51.2	86.7	82.5

The experimental results shown in Table 3 and Table 4 demonstrate that the TransC model significantly and consistently outperforms all baselines.

4.3. **Triplet Classification.** The aim of triplet classification is to determine whether a given triplet (h,r,t) is correct. This is a binary classification task for fact triplets. It was first explored in [11] to evaluate the NTN model and has been widely used to evaluate knowledge graph embedding [9, 13, 14, 15]. Consistent with the experiments in [9, 11, 13, 14, 15], we also used three datasets in this task: WN11, FB13 and FB15K.

4.3.1. *Evaluation protocol.* We followed the protocol used in the NTN model [11]. For triplet classification, we set a relation-specific threshold δ_r . For a triplet (h,r,t), if the dissimilarity score obtained by f_r is below δ_r , the triplet is classified as positive; otherwise, it is classified as negative. The relation-specific threshold δ_r is optimized by maximizing the classification accuracy on the validation set.

4.3.2. *Implementation.* We compared our model with several embedding models presented in [9, 11, 13, 14, 15]. In TransC, we attempted several parameter combinations; we selected the learning rate λ for the SGD from $\{0.001, 0.002, 0.01, 0.05\}$, margin γ from $\{0.5, 0.85, 1, 1.25, 1.5\}$, latent dimension k from $\{20, 50, 100, 120\}$, and batch size B from $\{480, 960, 1440, 4800\}$. The optimal hyperparameters of TransC for the dataset WN11 are λ=0.001, γ=0.85, k=100, and B=960; we limited the number of epochs to 500. For FB13, the optimal hyperparameters of TransC are λ=0.001, γ=1.0, k=100, and B=960; we limited the number of epochs to 500. For FB15K, the optimal hyperparameters of TransC are λ=0.01, γ=0.85, k=100, and B=1440; we limited the number of epochs to 500.

4.3.3. *Experimental Results.* The evaluation results are reported in Table 5. Our TransC model has significantly better accuracy for triplet classification than all of the baseline models studied, including TransE, TransH, TransR, and TransD, on WN11 and FB15K.

(1) For example, TransC is 11.7%, 8.8%, 1.7% and 1.2% more accurate than TransE, TransH, TransR and TransD, respectively, for WN11 and 9.5%, 8.5%, 4.8% and 0.7% more accurate than TransE, TransH, TransR and TransD, respectively, for FB15K.

(2) For FB13, TransC performs better than all baseline models except TransD. This finding may be attributable to the fact that there are more entity numbers in FB13 than in other datasets. Additionally, the TransD model creates a dynamic mapping matrix for each entity, and thus, it achieves better accuracy in triplet classification.

The experimental results show that the TransC model utilizes simple parameters, similar to TransE, and exhibits improved prediction accuracy.

Table 5. Triplet classification accuracies (%) of different embedding methods

Methods	WN11	FB13	FB15K
SE	53.0	75.2	-
SME	70.0	63.7	-
LFM	73.8	84.3	-
TransE	75.9	81.5	79.2
TransH	78.8	83.3	80.2
TransR	85.9	82.5	83.9
TransD	86.4	89.1	88.0
TransC(our)	87.6	86.4	88.7

5. **Conclusions and Future Work.** Learning and reasoning methods have been proposed to produce knowledge graphs, and a fast and effective reasoning algorithm is required. Considering the obvious inadequacies of the learning and reasoning methods based on translation represented by TransE in the reasoning process, in which no relation semantic-type constraints are considered in the construction of corrupted triplets, we proposed the TransC model. By adding semantic-type constraints for each relation r into the reasoning process, the corrupted head and tail entities can satisfy the relation semantic-type constraints in the TransC model. Thus, the triplets in the training process have more practical significance, and the training model has high predictive accuracy. The experimental results obtained for link prediction and triplet classification confirm the advantages of TransC, particularly its simple parameters and high predictive accuracy. Knowledge graph learning and reasoning constitutes a promising research area. However, it remains in the early stage of research and exploration, and many problems are still unresolved. For TransC, in the future, the constraint conditions should be further analyzed to expand this model's suitability for more tasks, and this model should be combined with other reasoning models to improve the accuracy of the resulting predictions.

Acknowledgment. This work is supported by the Natural Science Foundation of Fujian Province (No. 2017J01755), the Science Foundation of the Fujian Province (No.2016Y0060), Education Scientific Project of Young Teacher of Fujian Provincial (No. JAT160077), and the Research Project of Xiamen Overseas Students (No. XRS201631401). We also gratefully acknowledge the helpful comments and suggestions provided by the reviewers, which have improved the presentation of this work.

REFERENCES

[1] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, *Semantic web*, vol. 8, pp. 489-508, 2017.

[2] G. A. Miller, WordNet: a lexical database for English,*Communications of the Acm*, vol. 38, pp. 39-41, 1995.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge,*In Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, New York,USA, pp. 1247-1250,2008.

[4] S.Amit. (2012).*Introducing the Knowledge Graph:Things,Not Strings.* Available: http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html

[5] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, et al.,Knowledge vault: a web-scale approach to probabilistic knowledge fusion,*In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, USA, pp. 601-610,2014.

[6] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, Learning Structured Embeddings of Knowledge Bases, *In Proceedings of the 25th International Conference on Artificial Intelligence(AAAI)*, San Francisco,USA, pp. 301-306,2011.

[7] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, Joint learning of words and meaning representations for open-text semantic parsing, *In International Conference on Artificial Intelligence and Statistics*, La Palma, Canary Islands, pp. 127-135,2012.

[8] R. Jenatton, N. L. Roux, A. Bordes, and G. Obozinski, A latent factor model for highly multi-relational data,*In Advances in Neural Information Processing Systems*, Harrahs and Harveys, Lake Tahoe,USA, pp. 3167-3175,2012.

[9] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, Translating Embeddings for Modeling Multi-relational Data,*In Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, USA, pp. 2787-2795,2013.

[10] J. Weston, A. Bordes, O. Yakhnenko, and N. Usunier, "Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction,*In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, pp. 1366-1371,2013.

[11] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, Reasoning With Neural Tensor Networks for Knowledge Base Completion,*In Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, USA,pp. 926-934,2013.

[12] K.-W. Chang, W.-t. Yih, and C. Meek, Multi-Relational Latent Semantic Analysis, *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, pp. 1602-1612,2013.

[13] Z. Wang, J. Zhang, J. Feng, and Z. Chen, Knowledge graph embedding by translating on hyperplanes,*In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Qubec, Canada, pp. 1112-1119,2014.

[14] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, Learning entity and relation embeddings for knowledge graph completion, *In The Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin Texas, USA, pp. 2181-2187,2015.

[15] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, Knowledge Graph Embedding via Dynamic Mapping Matrix, *In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing,China, pp. 687-696,2015.

[16] K.-W. Chang, W.-t. Yih, B. Yang, and C. Meek, Typed tensor decomposition of knowledge bases for relation extraction,*In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona, Spain, pp. 1568-1579,2014.

[17] M. Nickel, V. Tresp, and H. P. Kriegel, A Three-Way Model for Collective Learning on Multi-Relational Data, *In International Conference on Machine Learning*, Washington, USA, pp. 809-816,2011.

[18] N. Lao and W. W. Cohen, Relational retrieval using a combination of path-constrained random walks, *Machine Learning*, vol. 81, pp. 53-67, 2010.

[19] S. Guo, Q. Wang, B. Wang, L. Wang, and L. Guo, Semantically smooth knowledge graph embedding,*In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing,China, pp. 84-94,2015.

[20] Q. Wang, B. Wang, and L. Guo, Knowledge base completion using embeddings and rules, *In Proceedings of the 24th International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, pp. 1859-1865,2015.

[21] Z. Wang, J. Zhang, J. Feng, and Z. Chen, Knowledge Graph and Text Jointly Embedding, *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar,pp. 1591-1601,2014.

[22] K. Toutanova and D. Chen, Observed versus latent features for knowledge base and text inference,*In Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, Beijing,China, pp. 57-66,2015.

[23] K. Toutanova, D. Chen, P. Pantel, P. Choudhury, and M. Gamon, Representing text for joint embedding of text and knowledge bases, *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing(EMNLP)*, Lisbon, Portugal,pp. 1499–1509,2015.

[24] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks,*Journal of Machine Learning Research*, vol. 9, pp. 249-256, 2015.