A Translation-Based Knowledge Graph Embedding Preserving Logical Property of Relations

Hee-Geun Yoon, Hyun-Je Song, Seong-Bae Park, Se-Young Park
School of Computer Science and Engineering
Kyungpook National University
Daegu, 41566, Korea
{hkyoon, hjsong, sbpark, sypark}@sejong.knu.ac.kr

Abstract

This paper proposes a novel translation-based knowledge graph embedding that preserves the logical properties of relations such as transitivity and symmetricity. The embedding space generated by existing translation-based embeddings do not represent transitive and symmetric relations precisely, because they ignore the role of entities in triples. Thus, we introduce a role-specific projection which maps an entity to distinct vectors according to its role in a triple. That is, a head entity is projected onto an embedding space by a head projection operator, and a tail entity is projected by a tail projection operator. This idea is applied to TransE, TransR, and TransD to produce lppTransE, lppTransR, and lppTransD, respectively. According to the experimental results on link prediction and triple classification, the proposed logical property preserving embeddings show the state-of-the-art performance at both tasks. These results prove that it is critical to preserve logical properties of relations while embedding knowledge graphs, and the proposed method does it effectively.

1 Introduction

Representing knowledge as a graph is one of the most effective ways to utilize human knowledge with a machine, and various large-scale knowledge graphs such as Freebase (Bollacker et al., 2008) and Yago (Suchanek et al., 2007) are available these days. However, the sparsity of the graphs makes it difficult to utilize them in real world applications. In spite of their huge volume, the relations among entities in the graphs are insufficient, which results in very limited inference of the knowledge of the graphs. Therefore, it is of importance to resolve such sparsity of knowledge graphs.

One of the most promising methods to complete knowledge graphs is to embed the graphs in a low-dimensional continuous vector space. This method learns a vector representation of a knowledge graph, and the plausibility of a certain knowledge within the graph is measured with algebraic operations in the vector space. Thus, new knowledge can be harvested from the space by finding knowledge instances with high plausibility.

The translation-based model among various knowledge-embedding models shows the state-of-the-art performance of knowledge graph completion (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015; Ji et al., 2015). TransE (Bordes et al., 2013) is one of the well-known translation-based approaches to this problem. When a set of knowledge triples (h, r, t) composed of a relation (r) and two entities (h and t) is given, it finds vector representations of h, t, and r by compelling the vector of t to be same with the sum of the vectors of h and r. While TransE embeds all relations in a single vector space, TransH (Wang et al., 2014) and TransR (Lin et al., 2015) assume that each relation has its own embedding space. On the other hand, Ji et al. (2015) found out that even a single relation or a single entity usually has multiple types. Thus, they have proposed TransD which allows multiple mapping matrices of entities and relations.

Even though these translation-based models achieve high performance in knowledge graph completion, they all ignore logical properties of rela-

Proceedings of NAACL-HLT 2016, pages 907–916,
San Diego, California, June 12-17, 2016. ©2016 Association for Computational Linguistics

tions. That is, transitive relations and symmetric relations lose their transitivity or symmetricity in the vector space generated by the translation-based models. As a result, the models can not complete only new knowledge with such relations, but also new knowledge with a relation affected by the relations. In most knowledge graphs, transitive or symmetric relations are common. For instance, FB15K, one of the benchmark datasets for knowledge graph completion, has a number of transitive and symmetric relations. About 20% of triples in FB15K have a transitive or a symmetric relation. Therefore, the ignorance of logical properties of relations becomes a serious problem in knowledge graph completion.

The main reason why existing translation-based embeddings can not reflect logical properties of relations is that they do not consider the role of entities. An entity should be a different vector in the embedding space according to its role. Therefore, the solution to preserve the logical properties of relations in the embedding space is to distinguish the role of entities while embedding entities and relations.

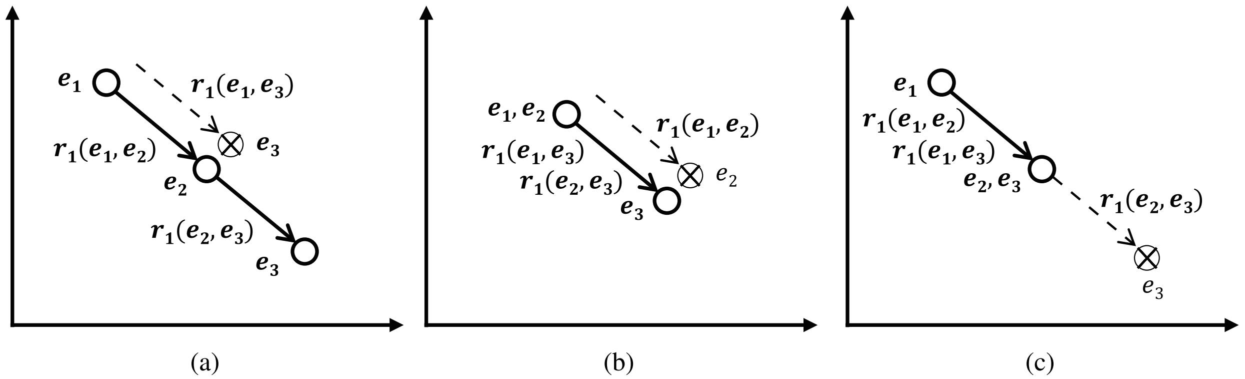
In this paper, we propose a role-specific projection to preserve logical properties of relations in an embedding space. This can be implemented by projecting a head entity onto an embedding space by a head projection operator and a tail entity by a tail projection operator. As a result, an identical entity is represented as two distinct vectors. This idea can be applied to various translation-based models including TransE, TransR, and TransD. Therefore, we also propose how to modify existing translation-based models to preserve logical properties. The effectiveness of the proposed idea is verified with two tasks of link prediction and triple classification using standard benchmark datasets of WordNet and Freebase. According to the experimental results, the logical property preserving embeddings achieve the state-of-the-art performance in both tasks.

2 Related Work

The sparsity of knowledge graphs is one of the most critical issues in utilizing them in real-world applications. Thus, there have been a number of studies on completing knowledge graphs as a solution to overcome the sparsity. Link prediction is one of the promising ways for knowledge graph completion. This task predicts new relations between entities on a knowledge graph by investigating existing relations of the graph (Nickel et al., 2015; Neelakantan and Chang, 2015). The methods used for link prediction can be categorized into three groups. One group consists of the methods based on graph features. The observable features used in these methods are the paths between entity pairs (Lao and Cohen, 2010; Lao et al., 2011) and subgraphs (Gardner and Mitchell, 2015). The other group is composed of the methods based on Markov random fields. The studies belonging to this group inference new relations by probabilistic soft logic (Pujara et al., 2013) and first-order logic (Jiang et al., 2012).

Knowledge graph embedding is another prominent method for link prediction (Bordes et al., 2011; Nickel et al., 2011; Guo et al., 2015; Neelakantan et al., 2015). It embeds entities of a knowledge graph into a continuous low dimensional space as vectors, and embeds relations as vectors or matrices. These vectors are optimized by a score function of each knowledge graph embedding model. The Semantic Matching Energy (SME) model proposed by Bordes et al. (2014) finds vector representations of entities and relations using a neural network. When a triple (h, r, t) is given, SME makes two relation-dependent embeddings of (h, r) and (r, t). Its score function for a triple is the similarity between the embeddings. Since relations are expressed as vectors instead of matrices, the complexity of SME is relatively low compared to other embedding methods. Jenatton et al. (2012) suggested Latent Factor Model (LFM). In order to capture both the first-order and the second-order interactions between two entities, LFM adopts a bilinear function as its score function. In addition, it represents a relation as a weighted sum of sparse latent factors to work with a large number of relations. Socher et al. (2013) proposed the Neural Tensor Network (NTN) model. NTN is a highly expressive embedding model which has a bilinear tensor layer instead of a standard linear neural network layer. As a result, it can process various interactions between entity vectors. However, it is difficult to process large-scale knowledge graphs with NTN due to its high complexity.

The current main stream of knowledge graph embedding is a translation-based embedding approach. The basic idea of this embedding is that entities and



(a) (b) (c)

Figure 1: An example of entity vectors trained wrong by a transitive relation.

relations are represented as vectors, and relations are treated as operators to translate entities into other positions on an embedding space. Thus, they try to find vector representations of h, t, and r so that the vector of t becomes the sum of the vectors of h and r. TransE (Bordes et al., 2013) is the simplest translation-based graph embedding. It assumes that all vectors of entities and relations lie on a single vector space. As a result, it fails in dealing with the reflexivity and the multiplicities of relations except 1-to-1. The solution to this problem is to allow the entities to play different roles according to relations, but TransE is unable to do it.

An entity plays multiple roles in TransH (Wang et al., 2014), since TransH allows entities to have multiple vector representations. In order to obtain multiple representations of an entity, TransH projects an entity vector into relation-specific hyperplanes. TransR (Lin et al., 2015) also solves the problems of TransE by introducing relation spaces. It allows an entity to have various vector representations by mapping an entity vector into relation-specific spaces rather than relation-specific hyperplanes. Although both TransH and TransR overcome the limitations of TransE, they are still not able to handle multiple types of relations which is determined by head and tail entities of each relation. For example, let us consider two triples of (*California*, *part_of*, *USA*) and (*arm*, *part_of*, *body*). Both triples have a relation *part_of* in common, but the relation should be interpreted differently in each triple. Ji et al. (2015) have proposed TransD in which a relation can have multiple relation spaces according to its entities. TransD constructs relation mapping matrices dynamically by considering entities and a relation simultaneously. For this, it introduces projection vectors for entities and relations, and then constructs the mapping matrices by multiplying these entity and relation projection vectors. As a result, every relation in TransD has multiple entity-specific spaces.

3 Loss of Logical Properties in Translation-Based Embeddings

Translation-based embeddings aim to find vector representations of knowledge graph entities in an embedding space by regarding relations as translation of entities in the space. Since they map the entities onto a vector space regardless of the role of the entities, they do not express logical properties of relations such as transitivity and symmetricity. That is, the vectors of transitive or symmetric relations do not deliver transitivity or symmetricity in the embedding spaces from translation-based embeddings.

For instance, let us consider a transitive relation. Assume that we have three triples (e_1, r_1, e_2), (e_2, r_1, e_3), and (e_1, r_1, e_3) and r_1 is a transitive relation. When the vector of r_1 is not a zero vector, there could be three types of entity vectors as shown in Figure 1. In Figure 1-(a), e_1, e_2, and e_3 are placed linearly. In this case, (e_1, r_1, e_3) can not be expressed in this figure. When e_1 and e_2 are placed at the same point like Figure 1-(b), (e_1, r_1, e_2) can not be expressed. In Figure 1-(c), (e_2, r_1, e_3) can not be expressed when e_2 and e_3 are same. In a similar way, translation-based embeddings can not express symmetric relations perfectly.

The problems caused by wrong expression of transitive and symmetric relations are two-folds.

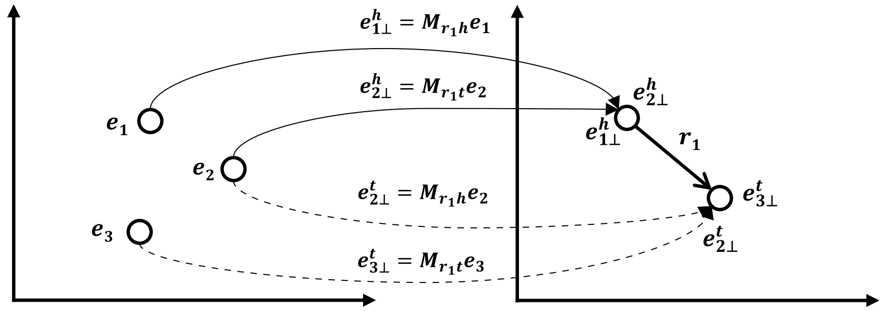

$e_{1\perp}^h = M_{r_1 h} e_1$
$e_{2\perp}^h = M_{r_1 t} e_2$
e_1
e_2
$e_{2\perp}^t = M_{r_1 h} e_2$
e_3
$e_{3\perp}^t = M_{r_1 t} e_3$
$e_{2\perp}^h$
$e_{1\perp}^h$
r_1
$e_{3\perp}^t$
$e_{2\perp}^t$

Figure 2: Simple illustration on a transitive relation with role-specific projections.

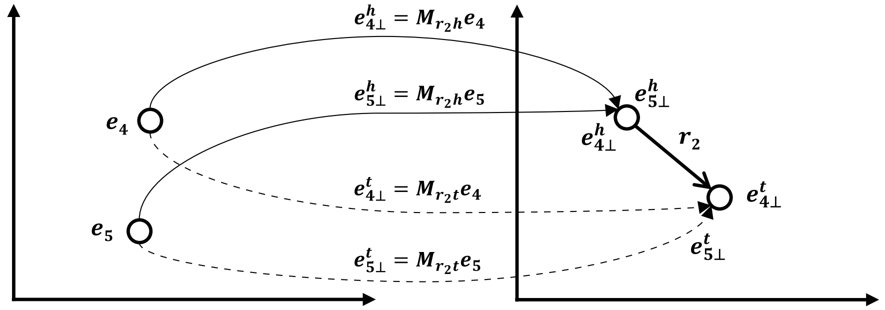

$e_{4\perp}^h = M_{r_2 h} e_4$
$e_{5\perp}^h = M_{r_2 h} e_5$
e_4
$e_{4\perp}^t = M_{r_2 t} e_4$
e_5
$e_{5\perp}^t = M_{r_2 t} e_5$
$e_{5\perp}^h$
$e_{4\perp}^h$
r_2
$e_{4\perp}^t$
$e_{5\perp}^t$

Figure 3: Simple illustration on a symmetric relation with role-specific projections.

One fold is that the relations with logical properties are common in knowledge bases. Two benchmark datasets of FB15K and WN18 in Table 2 prove it. There are 483,142 triples in FB15K, and 84,172 (= 47,841 + 36,331) triples among them have a transitive or symmetric relation. That is, the translation-based embeddings do not express triples precisely for about 17% of triples in FB15K. 22.4% of triples in another dataset WN18 also have a transitive or symmetric relation. The other is that transitive or symmetric relations do not affect the entities that are directly connected by the relations, but affect also other entities shared by non-transitive and non-symmetric relations through the entities. Therefore, it is of importance in translation-based embeddings to represent transitive and symmetric relations precisely.

4 Logical Property Preserving Embedding

4.1 Role-Specific Projection of Entity Vectors

The main reason why transitive or symmetric relations are not represented precisely by existing translation-based embeddings is that they ignore the role of entities in embedding them onto a vector space. That is, when a triple (h, r, t) is given, h and t plays different roles. However, the existing embeddings treat them equally and embed them into a space in the same way. Therefore, in order to express entities and relations more precisely, entities should be represented differently according to their role in a triple.

Figure 2 shows how entities can be represented according to their role. In this figure, solid lines represent entity mappings as head roles and dotted lines mean that entities are mapped as tails. Assume that three triples of (e_1, r_1, e_2), (e_2, r_1, e_3), and (e_1, r_1, e_3) are given with a transitive relation r_1. e_1 plays only a head role and e_3 plays only a tail role, while e_2 plays both roles. Then, the entity vectors in the entity space are mapped into the space of r_1 using two mapping matrices $\mathbf{M}_{r_1 h}$ and $\mathbf{M}_{r_1 t}$. That is, head entities are mapped by $\mathbf{M}_{r_1 h}$, while tail entities are projected by $\mathbf{M}_{r_1 t}$. Let $e_{1\perp}^h$ and $e_{2\perp}^h$ be the projected vectors of e_1 and e_2 respectively by $\mathbf{M}_{r_1 h}$, and let $e_{2\perp}^t$ and $e_{3\perp}^t$ be the projected vectors of e_2 and e_3 respectively by $\mathbf{M}_{r_1 t}$. $e_{1\perp}^h$ and $e_{2\perp}^h$ are placed at the same point in the space of r_1 from (e_2, r_1, e_3) and (e_1, r_1, e_3). Similarly, $e_{2\perp}^t$ and $e_{3\perp}^t$ are same from (e_1, r_1, e_2) and (e_1, r_1, e_3). Since e_2 is used as both a head and a tail, it is mapped differently as $e_{2\perp}^h$ and $e_{2\perp}^t$, respectively. Note that all three triples are well expressed in this space.

Symmetric relations also can be expressed precisely by logical property preserving knowledge graph embedding. Assume that two triples of (e_4, r_2, e_5) and (e_5, r_2, e_4) are given with a symmetric relation r_2. Figure 3 shows how the triples are well represented. The solid lines imply that entities are mapped by $\mathbf{M}_{r_2h}$ while the dotted lines mean that entities are mapped by $\mathbf{M}_{r_2t}$. By placing $e^h_{4\perp}$ and $e^h_{5\perp}$ at the same point and imposing $e^t_{4\perp}$ and $e^t_{5\perp}$ at the same point, r_2 is precisely expressed as a symmetric relation in the embedding space.

4.2 Realization of Logical Property Preserving Embedding

Due to the simplicity of role-specific projection of entity vectors, it can be applied to various translation-based embeddings. In this paper, we apply it to TransE, TransR, and TransD.

4.2.1 TransE

The score function of TransE is

$$f^E_r(h,t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{l_{1/2}},$$

where $\mathbf{h}$, $\mathbf{t} \in \mathbb{R}^n$, and $\mathbf{r} \in \mathbb{R}^n$ are the vectors of a head entity, a tail entity, and a relation on a single embedding space. In the logical property preserving TransE (lppTransE), $\mathbf{h}$ and $\mathbf{t}$ should be mapped differently. For this purpose, we adopt a head and a tail space mapping matrices of $\mathbf{M}_h \in \mathbb{R}^{n\times n}$ and $\mathbf{M}_t \in \mathbb{R}^{n\times n}$. As a result, the score function of lppTransE becomes

$$f^{lppE}_r(h,t) = \|\mathbf{M}_h\mathbf{h} + \mathbf{r} - \mathbf{M}_t\mathbf{t}\|_{l_{1/2}}.$$

This is similar to the score function of TransR. The difference between lppTransE and TransR is that both h and t are mapped by a single mapping matrix for r in TransR, while h is mapped by $\mathbf{M}_h$ and t is by $\mathbf{M}_t$ in lppTransE.

4.2.2 TransR

The entities in TransR are mapped into vectors in different relation space according to a relation. Thus, its score function is defined as

$$f^R_r(h,t) = \|\mathbf{M}_r\mathbf{h} + \mathbf{r} - \mathbf{M}_r\mathbf{t}\|_{l_{1/2}},$$

where $\mathbf{M}_r \in \mathbb{R}^{m\times n}$ is a mapping matrix for a relation r which is represented as $\mathbf{r} \in \mathbb{R}^m$. Thus, in the logical property preserving TransR (lppTransR), the mapping matrix of each relation is split into a head mapping matrix $\mathbf{M}_{r_h} \in \mathbb{R}^{m\times n}$ and a tail mapping matrix $\mathbf{M}_{r_t} \in \mathbb{R}^{m\times n}$. Then, the score function of lppTransR is

$$f^{lppR}_r(h,t) = \|\mathbf{M}_{r_h}\mathbf{h} + \mathbf{r} - \mathbf{M}_{r_t}\mathbf{t}\|_{l_{1/2}}.$$

With these two distinct mapping matrices, entities can have two different vector representations in the same relation space.

4.2.3 TransD

TransD maps entity vectors into different vectors in relation spaces according to entity and relation types. That is, the entity vectors are mapped by entity-relation specific mapping matrices. Thus, its score function is defined as

$$f^D_r(h,t) \;=\; \|\mathbf{M}_{rh}\mathbf{h} + \mathbf{r} - \mathbf{M}_{rt}\mathbf{t}\|_{l_{1/2}},$$

where $\mathbf{M}_{rh} \in \mathbb{R}^{m\times n}$ and $\mathbf{M}_{rt} \in \mathbb{R}^{m\times n}$ are entity-relation specific mapping matrices. These mapping matrices are computed by multiplying projection vectors of an entity and a relation as follows.

$$\begin{aligned}\mathbf{M}_{rh} &= \mathbf{r}_p\mathbf{h}_p^\intercal + \mathbf{I}^{m\times n},\\ \mathbf{M}_{rt} &= \mathbf{r}_p\mathbf{t}_p^\intercal + \mathbf{I}^{m\times n},\end{aligned}$$

where $\mathbf{h}_p, \mathbf{t}_p$ and $\mathbf{r}_p$, are the projection vectors for a head, a tail and a relation.

The logical property preserving TransD (lppTransD) divides $\mathbf{r}_p$ into two projection vectors $\mathbf{r}_{p_h}$ and $\mathbf{r}_{p_t}$ to reflect the role of entities. The mapping matrices then becomes

$$\begin{aligned}\mathbf{M}'_{rh} &= \mathbf{r}_{p_h}\mathbf{h}_p^\intercal + \mathbf{I}^{m\times n},\\ \mathbf{M}'_{rt} &= \mathbf{r}_{p_t}\mathbf{t}_p^\intercal + \mathbf{I}^{m\times n}.\end{aligned}$$

Then, its score function is

$$f^{lppD}_r(h,t) \;=\; \left\|\mathbf{M}'_{rh}\mathbf{h} + \mathbf{r} - \mathbf{M}'_{rt}\mathbf{t}\right\|_{l_{1/2}}.$$

Table 2: A simple statistics on datasets.

Dataset	#Rel	#Ent	#Train	#Valid	#Test	#Transitive	#Symmetric	Ratio
WN11	11	38,696	112,581	2,609	10,544	822	1,597	2.2%
FB13	13	75,043	316,232	5,908	23,733	262	4,152	1.4%
WN18	18	40,943	141,442	5,000	5,000	2,001	29,667	22.4%
FB15K	1,345	14,951	483,142	50,000	59,071	47,841	36,331	17.4%

Table 1: Complexity of knowledge graph embedding models.

Model	No. of parameters
TransE	$\mathcal{O}\left(N_e n + N_r n\right)$
TransR	$\mathcal{O}\left(N_e n + N_r\left(m+1\right)n\right)$
TransD	$\mathcal{O}\left(2N_e n + 2N_r m\right)$
lppTransE	$\mathcal{O}\left(N_e n + N_r n + n^2\right)$
lppTransR	$\mathcal{O}\left(N_e n + N_r\left(2m+1\right)n\right)$
lppTransD	$\mathcal{O}\left(2N_e n + 3N_r m\right)$

4.3 Training Logical Property Preserving Embeddings

Since all logical property preserving embeddings are based on the score function used by previous translation-based knowledge graph embeddings, they can be trained using a margin-based ranking loss defined as

$$L = \sum_{(h,r,t)\in P} \sum_{(h',r,t')\in N} \max(0, f_r^*(h,t)+\gamma-f_r^*(h',t')),$$

where f_r^* is the score function of a corresponding logical property preserving embedding. Here, P and N are sets of correct and incorrect triples, and γ is a margin. N is constructed by replacing a head or a tail entity in an existing triple because a knowledge graph has only correct triples. All logical property preserving embeddings are optimized by stochastic gradient descent.

The complexities of the logical property preserving embeddings are shown in Table 1. N_e and N_r in this table are the number of entities and relations, and n and m are the dimensions of entity and relation embedding spaces. Their complexity mainly depends on the number of relations. Thus, the increased complexities of the logical property preserving embeddings is not significant, when compared with TransE, TransR, and TransD.

5 Experiments

The superiority of the proposed logical property preserving embeddings is shown through two kinds of tasks. The first task is *link prediction* (Bordes et al., 2013). This task predicts the missing entity when there is a missing entity in a given triple. The other is *triple classification* (Socher et al., 2013). This task aims to decide whether a given triple is correct or not.

5.1 Data Sets

Two popular knowledge graphs of WordNet (Miller, 1995) and Freebase (Bollacker et al., 2008) are used for evaluating embeddings. WordNet provides the semantic relations among words, and there exist its two widely-used subsets which are WN11 (Socher et al., 2013) and WN18 (Bordes et al., 2014). WN18 is used for link prediction, and WN11 is adopted for triple classification. Freebase represents general facts about the world. It has two subsets of FB13 (Socher et al., 2013) and FB15K (Bordes et al., 2014). FB15K is used for both triple classification and link prediction, while FB13 is employed only for triple classification.

Table 2 summarizes a simple statistics of each dataset. #Triples is the number of training triples in each benchmark dataset. #Transitive and #Symmetric are the number of triples which have transitive and symmetric relations, respectively. Ratio denotes the proportion of triples of which relation is transitive or symmetric. As shown in this table, the triples with a transitive or symmetric relation take a large proportion in WN18 and FB15K.

5.2 Link Prediction

For the evaluation of link prediction, we followed the evaluation protocols and metrics used in previous studies (Socher et al., 2013; Bordes et al., 2013; Lin et al., 2015; Ji et al., 2015). To compare the

Table 4: Experimental results on link prediction.

Dataset	WN18				FB15K			
Metric	Mean Rank		Hits@10 (%)		Mean Rank		Hits@10 (%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	263	251	75.4	89.2	243	125	34.9	47.1
TransH (unif)	318	301	75.4	86.7	211	84	42.5	58.5
TransH (bern)	401	388	73.0	82.3	212	87	45.7	64.4
TransR (unif)	232	219	78.3	91.7	226	78	43.8	65.5
TransR (bern)	238	225	79.8	92.0	198	77	48.2	68.7
TransD (unif)	242	229	79.2	92.5	211	**67**	49.4	74.2
TransD (bern)	224	212	79.6	92.2	194	91	**53.4**	77.3
lppTransE (unif)	336	323	**77.7** (+2.3)	**89.5** (+0.3)	228	78	**47.4** (+12.5)	**72.9** (+25.8)
lppTransE (bern)	342	329	**79.5** (+4.1)	**92.7** (+3.5)	215	95	**48.9** (+14.0)	**73.1** (+26.0)
lppTransR (unif)	331	317	**79.2** (+0.9)	**92.7** (+1.0)	238	79	**47.2** (+3.4)	**74.4** (+8.9)
lppTransR (bern)	334	321	79.6 (-0.2)	**92.8** (+0.8)	219	92	**50.4** (+2.2)	**77.2** (+8.5)
lppTransD (unif)	342	328	**79.3** (+0.1)	**93.6** (+1.1)	218	69	**49.6** (+0.2)	**77.4** (+3.2)
lppTransD (bern)	283	270	**80.5** (+0.9)	**94.3** (+2.1)	195	78	53.0 (-0.4)	**78.7** (+1.4)

Table 3: Parameter values in link prediction.

Dataset	Model	α	B	γ	n, m	D.S
WN18	lppTransE	0.001	1,440	1	50	L_1
	lppTransR	0.001	1,440	1	50	L_1
	lppTransD	0.001	1,440	2	50	L_1
FB15K	lppTransE	0.001	480	1	100	L_1
	lppTransR	0.001	4,800	1	100	L_1
	lppTransD	0.0001	4,800	2	100	L_1

methods for this task, two metrics of mean rank and Hits@10 are used. The mean rank measures the average rank of all correct entities, and Hits@10 is the proportion of correct triples ranked in top 10. Since there are two evaluation settings of "raw" and "filter" in this task (Bordes et al., 2013), we report both results. In addition, we report the results for two sampling methods of "bern" and "unif" (Wang et al., 2014) as the previous studies did.

There are five parameters in the proposed property preserving embeddings. They are a learning rate α, the number of training triples in each mini-batch B[1], a margin γ, the embedding dimension for entities and relations (n and m), and a dissimilarity measure in embedding score functions (D.S). The parameter values used in our experiments are given at Table 3. The iteration number of stochastic gradient descent is 1,000.

Table 4 shows the results on link prediction. The results of previous studies are referred from their report, since the same datasets are used. The values between parentheses are the improvement over their base models. The logical property preserving embeddings outperform all other methods for both "bern" and "unif" on WN18 except lppTransR with "bern" in the raw setting. In the raw setting, lppTransE, lppTransR, and lppTransD achieve 79.5%, 79.6%, and 80.5% of Hits@10 respectively in "bern", which are 4.1%, -0.2%, and 0.9% higher than those of TransE, TransR, and TransD. The logical property preserving embeddings show even higher performance in the filter setting. Hits@10 of lppTransD in "bern" is 94.3%, while that of TransD in "unif" is 92.5% and that in "bern" is 92.2%. Thus, lppTransD improves 2.1% over TransD in "bern". Especially, this performance of lppTransD is 1.8% higher than that of TransD in "unif", the previous state-of-the-art performance.

The logical property preserving embeddings outperform their base models also on FB15K. TransE is improved most significantly with this dataset. The improvements by lppTransE in the raw setting are 12.5% in "unif" and 14.0% in "bern", while those in the filter setting are 25.8% in "unif" and 26.0% in "bern". In addition, its Hits@10 exceeds those of TransH and TransR. That is, even if TransH and TransR were proposed to tackle the problem of TransE, the proposed lppTransE solves the problem better than TransH and TransR. lppTransD achieves just a little bit lower Hits@10 than TransD in "bern", but the improvements in the filter setting are notice-

[1] α and B are related with stochastic gradient descent.

Table 6: Parameter values in triple classification.

Dataset	Model	α	B	γ	n, m	D.S
WN11	lppTransE	0.01	120	2	20	L_1
	lppTransR	0.001	120	4	20	L_1
	lppTransD	0.0001	1,000	1	100	L_2
FB13	lppTransE	0.001	30	1	100	L_1
	lppTransR	0.0001	300	1	100	L_1
	lppTransD	0.0001	300	1	100	L_2
FB15K	lppTransE	0.001	480	1	100	L_1
	lppTransR	0.001	4,800	1	100	L_1
	lppTransD	0.0001	4,800	2	100	L_1

able. Since Hits@10 of TransD in "bern" is the best performance ever reported, that of lppTransD in "bern" becomes a new state-of-the-art performance.

Table 5 exhibits Hits@10s according to mapping property of the relations of FB15K. The notable trend of this table is that the logical property preserving embeddings show much higher Hits@10 than their base models in N-to-1 and N-to-N, while their Hits@10s are similar to those of their base models in 1-to-1 and 1-to-N. This is notable with TransD and lppTransD. lppTransD improves TransD, the previous state-of-the-art method by 7.3% (N-to-1) and 3.7% (N-to-N) in predicting head, and by 0.9% (N-to-1) and 0.3% (N-to-N). Note that it is important to verify if logical property preserving embeddings achieve good performances in N-to-N, since all transitive relations and some symmetric relations are, in general, N-to-N. According to this table, Hits@10s of most logical property preserving embeddings are improved significantly in N-to-N, which proves that the proposed method solves transitivity and symmetricity problem of previous embeddings.

5.3 Triple Classification

Three datasets of WN11, FB13, and FB15K are used in this task. WN11 and FB13 have negative triples, but FB15K has only positives. Thus, we generated negative triples for FB15K by following the strategy of (Socher et al., 2013). As a result, the classification accuracies on FB15K can not be compared directly with previous studies, and the accuracies on FB15K in this table are those obtained with our dataset. The parameter values for training TransE, TransH, TransR, and TransD are borrowed from their reports, and those for logical property preserving embeddings are shown in Table 6.

Table 7: Accuracies on triple classification. (%)

Dataset	WN11	FB13	FB15K
TransE (unif)	75.9	70.9	80.3
TransE (bern)	75.9	81.5	80.8
TransH (unif)	77.7	76.5	81.9
TransH (bern)	78.8	83.3	82
TransR (unif)	85.5	74.7	82.6
TransR (bern)	85.9	82.5	82.7
TransD (unif)	85.6	85.9	84.2
TransD (bern)	86.4	**89.1**	84.8
lppTransE (unif)	81.3 (+5.4)	72.3 (+1.4)	83.2 (+2.9)
lppTransE (bern)	81.3 (+5.4)	83.4 (+1.9)	83.6 (+2.8)
lppTransR (unif)	85.5 (+0.0)	79.5 (+4.8)	83.4 (+0.8)
lppTransR (bern)	85.5 (-0.4)	83.1 (+0.6)	84.6 (+1.9)
lppTransD (unif)	86.1 (+0.5)	86.6 (+0.7)	84.7 (+0.5)
lppTransD (bern)	86.2 (-0.2)	88.6 (-0.5)	**85.3** (+0.5)

Table 7 shows the accuracies of triple classification on the three datasets. The logical property preserving embeddings in general outperform their base models. lppTransE always shows higher accuracy than TransE, lppTransR than TransR except for WN11, and lppTransD than TransD in FB15K. One thing to note is that the improvements by the logical property preserving embeddings are always observed in FB15K, while those in WN11 and FB13 are small or slightly negative. This can be explained with the number of triples with a transitive or symmetric relation in the datasets. As shown in Table 2, the triples with such a relation take just a small portion of WN11 and FB13. The ratio of those triples is less than 2.3% in these datasets. However, as noted before, more than 17% triples are such ones in FB15K. Thus, the improvement in FB15K is remarkable. The other thing to note is that lppTransD shows the best accuracy in FB15K. These results imply that the proposed logical property preserving embeddings solve the problems of existing translation-based embeddings effectively.

6 Conclusion

This paper has proposed a new translation-based knowledge graph embedding that preserves logical properties of relations. Transitivity and symmetricity are very important characteristics of relations for representing and inferring knowledge, and the triples with such a relation take a large proportion of real-world knowledge graphs. In order to preserve the logical properties in an embedding space, an entity is forced to have multiple vector represen-

Table 5: Experimental results on FB15K according to mapping properties of relations. (%)

Tasks	Predicting Head (Hits@10)				Predicting Tail (Hits@10)			
Relation Category	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
TransE	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH (unif)	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransH (bern)	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR (unif)	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
TransR (bern)	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
TransD (unif)	80.7	85.8	47.1	75.6	80.0	**54.5**	80.7	77.9
TransD (bern)	**86.1**	**95.5**	39.8	78.5	**85.4**	50.6	94.4	81.2
lppTransE (unif)	77.5	86.5	43.1	75.9	76.7	46.4	79.4	75.5
lppTransE (bern)	78.3	93.5	35.4	74.7	78.1	45.9	80.3	76.6
lppTransR (unif)	75.2	88.1	41.5	75.2	76.6	44.9	82.7	72.6
lppTransR (bern)	84.3	94.2	49.7	79.2	84.3	46.9	91.9	79.6
lppTransD (unif)	82.7	89.5	53.2	81.6	82.3	52.6	84.8	**81.5**
lppTransD (bern)	86.0	94.2	**54.4**	**82.2**	79.7	43.2	**95.3**	79.7

tations according to its role in a triple. This idea has been applied to TransE, TransR, and TransD, and they are called as lppTransE, lppTransR, and lppTransD. Their superiority was shown through two tasks of link prediction and triple classification. The logical property preserving embeddings showed the improved performance over their base models[2]. Especially, lppTransD showed the state-of-the-art performance in both tasks. These results imply that the proposed role-specific projection is plausible to preserve logical properties of relations.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. R0101-15-0054, WiseKB: Big data based self-evolving knowledge base and reasoning platform).

References

[Bollacker et al.2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

[Bordes et al.2011] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 301–306.

[Bordes et al.2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

[Bordes et al.2014] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259.

[Gardner and Mitchell2015] Matt Gardner and Tom Mitchell. 2015. Efficient and expressive knowledge base completion using subgraph feature extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1488–1498.

[Guo et al.2015] Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 2015. 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*, pages 84–94.

[Jenatton et al.2012] Rodolphe Jenatton, Nicolas L. Roux, Antoine Bordes, and Guillaume R. Obozinski. 2012. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems*, pages 3167–3175.

[2] The source codes and resources can be downloaded from `http://ml.knu.ac.kr/lppKE`.

[Ji et al.2015] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. 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*, pages 687–696.

[Jiang et al.2012] Shangpu Jiang, Daniel Lowd, and Dejing Dou. 2012. Learning to refine an automatically extracted knowledge base using markov logic. In *Proceedings of the IEEE International Conference on Data Mining*, pages 912–917.

[Lao and Cohen2010] Ni Lao and William W. Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67.

[Lao et al.2011] Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 529–539.

[Lin et al.2015] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2181–2187.

[Miller1995] George A. Miller. 1995. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

[Neelakantan and Chang2015] Arvind Neelakantan and Ming-Wei Chang. 2015. Inferring missing entity type instances for knowledge base completion: New dataset and methods. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–525.

[Neelakantan et al.2015] Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 156–166.

[Nickel et al.2011] Maximilian Nickel, Volker Tresp, and Hans peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28st International Conference on Machine Learning*, pages 809–816.

[Nickel et al.2015] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. *CoRR*, abs/1503.00759.

[Pujara et al.2013] Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. Knowledge graph identification. In *Proceedings of the 12th International Semantic Web Conference*, pages 542–557.

[Socher et al.2013] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934.

[Suchanek et al.2007] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A core of semantic knowledge unifying wordnet and wikipedia. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706.

[Wang et al.2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1112–1119.