PairRE: Knowledge Graph Embeddings via Paired Relation Vectors

Linlin Chao, Jianshan He, Taifeng Wang, Wei Chu
AntGroup
{chulin.cll,yebai.hjs}@antgroup.com
{taifeng.wang,wei.chu}@alibaba-inc.com

Abstract

Distance based knowledge graph embedding methods show promising results on link prediction task, on which two topics have been widely studied: one is the ability to handle complex relations, such as N-to-1, 1-to-N and N-to-N, the other is to encode various relation patterns, such as symmetry/antisymmetry. However, the existing methods fail to solve these two problems at the same time, which leads to unsatisfactory results. To mitigate this problem, we propose PairRE, a model with paired vectors for each relation representation. The paired vectors enable an adaptive adjustment of the margin in loss function to fit for complex relations. Besides, PairRE is capable of encoding three important relation patterns, symmetry/antisymmetry, inverse and composition. Given simple constraints on relation representations, PairRE can encode subrelation further. Experiments on link prediction benchmarks demonstrate the proposed key capabilities of PairRE. Moreover, We set a new state-of-the-art on two knowledge graph datasets of the challenging Open Graph Benchmark.

1 Introduction

Knowledge graphs store huge amounts of structured data in the form of triples, with projects such as WordNet (Miller, 1995), Freebase (Bollacker et al., 2008), YAGO (Suchanek et al., 2007) and DBpedia (Lehmann et al., 2015). They have gained widespread attraction from their successful use in tasks such as question answering (Bordes et al., 2014), semantic parsing (Berant et al., 2013), and named entity disambiguation (Zheng et al., 2012) and so on.

Since most knowledge graphs suffer from incompleteness, predicting missing links between entities has been a fundamental problem. This problem is named as link prediction or knowledge graph completion. Knowledge graph embedding methods, which embed all entities and relations into a low dimensional space, have been proposed for this problem.

Distance based embedding methods from TransE (Bordes et al., 2013) to the recent state-of-the-art RotatE (Sun et al., 2019) have shown substantial improvements on knowledge graph completion task. Two major problems have been widely studied. The first one refers to handling of 1-to-N, N-to-1, and N-to-N complex relations (Bordes et al., 2013; Lin et al., 2015). In case of the 1-to-N relations, given triples like ($StevenSpielberg$, $DirectorOf$, ?), distance based models should make all the corresponding entities about film name like $Jaws$ and $JurassicPark$ have closer distance to entity $StevenSpielberg$ after transformation via relation $DirectorOf$. The difficulty is that all these entities should have different representations. Same issue happens in cases of N-to-N and N-to-1 relations. The latter is learning and inferring relation patterns according to observed triples, as the success of knowledge graph completion heavily relies on this ability (Bordes et al., 2013; Sun et al., 2019). There are various types of relation patterns: symmetry (e.g., $IsSimilarTo$), antisymmetry (e.g., $FatherOf$), inverse (e.g., $PeopleBornHere$ and $PlaceOfBirth$), composition (e.g., my mother's father is my grandpa) and so on.

Previous methods solve these two problems separately. TransH (Wang et al., 2014), TransR (Lin et al., 2015), TransD (Ji et al., 2015) all focus on ways to solve complex relations. However, these methods can only encode symmetry/antisymmetry relations. The recent state-of-the-art RotatE shows promising results to encode symmetry/antisymmetry, inverse and composition relations. However, complex relations remain challenging to predict.

4360

Here we present PairRE, an embedding method that is capable of encoding complex relations and multiple relation patterns simultaneously. The proposed model uses two vectors for relation representation. These vectors project the corresponding head and tail entities to Euclidean space, where the distance between the projected vectors is minimized. This provides three important benefits:

- The paired relation representations enable an adaptive adjustment of the margin in loss function to fit for different complex relations;

- Semantic connection among relation vectors can be well captured, which enables the model to encode three important relation patterns, symmetry/antisymmetry, inverse and composition;

- Adding simple constraints on relation representations, PairRE can encode subrelation further.

Besides, PairRE is a highly efficient model, which contributes to large scale datasets.

We evaluate PairRE on six standard knowledge graph benchmarks. The experiment results show PairRE can achieve either state-of-the-art or highly competitive performance. Further analysis also proves that PairRE can better handle complex relations and encode symmetry/antisymmetry, inverse, composition and subrelation relations.

2 Background and Notation

Given a knowledge graph that is represented as a list of fact triples, knowledge graph embedding methods define scoring function to measure the plausibility of these triples. We denote a triple by (h, r, t), where h represents head entity, r represents relation and t represents tail entity. The column vectors of entities and relations are represented by bold lower case letters, which belong to set $\mathcal{E}$ and $\mathcal{R}$ respectively. We denote the set of all triples that are true in a world as $\mathcal{T}$. $f_r(h, t)$ represents the scoring function.

We take the definition of complex relations from (Wang et al., 2014). For each relation r, we compute average number of tails per head (tphr) and average number of heads per tail (hptr). If tphr < 1.5 and hptr < 1.5, r is treated as 1-to-1; if tphr > 1.5 and hptr > 1.5, r is treated as a N-to-N; if tphr > 1.5 and hptr < 1.5, r is treated as 1-to-N.

We focus on four important relation patterns, which includes: (1) **Symmetry/antisymmetry**. A relation r is symmetric if $\forall e_1, e_2 \in \mathcal{E}, (e_1, r, e_2) \in \mathcal{T} \iff (e_2, r, e_1) \in \mathcal{T}$ and is antisymmetric if $(e_1, r, e_2) \in \mathcal{T} \Rightarrow (e_2, r, e_1) \notin \mathcal{T}$; (2) **Inverse**. If $\forall e_1, e_2 \in \mathcal{E}, (e_1, r_1, e_2) \in \mathcal{T} \iff (e_2, r_2, e_1) \in \mathcal{T}$, then r_1 and r_2 are inverse relations; (3) **Composition**. If $\forall e_1, e_2, e_3 \in \mathcal{E}, (e_1, r_1, e_2) \in \mathcal{T} \wedge (e_2, r_2, e_3) \in \mathcal{T} \Rightarrow (e_1, r_3, e_3) \in \mathcal{T}$, then r_3 can be seen as the composition of r_1 and r_2; (4) **Subrelation** (Qu and Tang, 2019). If $\forall e_1, e_2 \in \mathcal{E}, (e_1, r_1, e_2) \in \mathcal{T} \Rightarrow (e_1, r_2, e_2) \in \mathcal{T}$, then r_2 can be seen as a subrelation of r_1.

3 Related Work

Distance based models. Distance based models measure plausibility of fact triples as distance between entities. TransE interprets relation as a translation vector r so that entities can be connected, i.e., $h + r \approx t$. TransE is efficient, though cannot model symmetry relations and have difficulty in modeling complex relations. Several models are proposed for improving TransE to deal with complex relations, including TransH, TransR, TransD, TranSparse (Ji et al., 2016) and so on. All these methods project the entities to relation specific hyperplanes or spaces first, then translate projected entities with relation vectors. By projecting entities to different spaces or hyperplanes, the ability to handle complex relations is improved. However, with the added projection parameters, these models are unable to encode inverse and composition relations.

The recent state-of-the-art, RotatE, which can encode symmetry/antisymmetry, inverse and composition relation patterns, utilizes rotation based translational method in a complex space. Although expressiveness for different relation patterns, complex relations remain challenging. GC-OTE (Tang et al., 2020) proposes to improve complex relation modeling ability of RotatE by introducing graph context to entity embedding. However, the calculation of graph contexts for head and tail entities is time consuming, which is inefficient for large scale knowledge graphs, e.g. ogbl-wikikg (Hu et al., 2020).

Another related work is SE (Bordes et al., 2011), which utilizes two separate relation matrices to project head and tail entities. As pointed out by (Sun et al., 2019), this model is not able to encode symmetry/antisymmetry, inverse and composition

Method	Score Function	Performance of complex relations	Relation Patterns				
			Sym	$Asym$	Inv	$Comp$	Sub
TransE	$-\|\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}\|$	Low	✗	✓	✓	✓	✗
TransR	$-\|\boldsymbol{M_r h}+\boldsymbol{r}-\boldsymbol{M_r t}\|$	High	✓	✓	✗	✗	✗
RotatE	$-\|\boldsymbol{h}\circ\boldsymbol{r}-\boldsymbol{t}\|$	Low	✓	✓	✓	✓	✗
PairRE	$-\|\boldsymbol{h}\circ\boldsymbol{r}^H-\boldsymbol{t}\circ\boldsymbol{r}^T\|$	High	✓	✓	✓	✓	✓*

Table 1: Comparison between PairRE and some distance based embedding methods. Sym, $Asym$, Inv, $Comp$ and Sub are abbreviations for symmetry, antisymmetry, inverse and subrelation respectively. ✓* means the model can have the specific capability with some constraints.

relations.

Table 1 shows comparison between our method and some representative distance based methods. As the table shows, our model is the most expressive one, with the ability to handle complex relations and encode four key relation patterns.

Semantic matching models. Semantic matching models exploit similarity-based scoring functions, which can be divided into bilinear models and neural network based models. As the models have been developed, such as RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2014), HolE (Nickel et al., 2016), ComplEx (Trouillon et al., 2016) and QuatE (Zhang et al., 2019), the key relation encoding abilities are enriched. However, all these models have the flaw in encoding composition relations (Sun et al., 2019).

RESCAL, ComplEx and SimplE (Kazemi and Poole, 2018) are all proved to be fully expressive when embedding dimensions fulfill some requirements (Wang et al., 2018; Trouillon et al., 2016; Kazemi and Poole, 2018). The fully expressiveness means these models can express all the ground truth which exists in the data, including complex relations. However, these requirements are hardly fulfilled in practical use. It is proved by (Wang et al., 2018) that, to achieve complete expressiveness, the embedding dimension should be greater than $N/32$, where N is the number of entities in dataset.

Neural networks based methods, e.g., convolution neural networks (Dettmers et al., 2018), graph convolutional networks (Schlichtkrull et al., 2018) show promising performances. However, they are difficult to analyze as they work as a black box.

Encoding Subrelation. Existing methods encode subrelation by utilizing first order logic rules. One way is to augment knowledge graphs with grounding of rules, including subrelation rules (Guo et al., 2018; Qu and Tang, 2019). The other way is adding constraints on entity and relation representations, e.g., ComplEx-NNE-AER and

SimplE$^+$. The second way enriches the models' expressiveness with relatively low cost. In this paper, we show that PairRE can encode subrelation with constraints on relation representations while keeping the ability to encode symmetry/antisymmetry, inverse and composition relations.

4 Methodology

To overcome the problem of modeling 1-to-N/N-to-1/N-to-N complex relations and enrich the capabilities for different relation patterns, we propose a model with paired vectors for each relation. Given a training triple (h, r, t), our model learns vector embeddings of entities and relation in real space. Specially, PairRE takes relation embedding as paired vectors, which is represented as $[\boldsymbol{r}^H, \boldsymbol{r}^T]$. $\boldsymbol{r}^H$ and $\boldsymbol{r}^T$ project head entity h and tail entity t to Euclidean space respectively. The projection operation is the Hadamard product[1] between these two vectors. PairRE then computes distance of the two projected vectors as plausibility of the triple . We want that $\boldsymbol{h}\circ\boldsymbol{r}^H \approx \boldsymbol{t}\circ\boldsymbol{r}^T$ when (h, r, t) holds, while $\boldsymbol{h}\circ\boldsymbol{r}^H$ should be far away from $\boldsymbol{t}\circ\boldsymbol{r}^T$ otherwise. In this paper, we take the L_1-norm to measure the distance.

In order to remove scaling freedoms, we also add constraint on embeddings similar to previous distance based models (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015). And the constraint is only added on entity embeddings. We want relation embeddings to capture semantic connection among relation vectors (e.g., $PeopleBornHere$ and $PlaceOfBirth$) and complex characteristic (e.g., 1-N) easily and sufficiently. For entity embedding, the L_2-norm is set to be 1.

The scoring function is defined as follows:

$$f_r(\boldsymbol{h}, \boldsymbol{t}) = -\|\boldsymbol{h}\circ\boldsymbol{r}^H - \boldsymbol{t}\circ\boldsymbol{r}^T\|, \qquad (1)$$

where $\boldsymbol{h}, \boldsymbol{r}^H, \boldsymbol{r}^T, \boldsymbol{t} \in \mathbb{R}^d$ and $\|\boldsymbol{h}\|^2 = \|\boldsymbol{t}\|^2 = 1$. The model parameters are, all the entities' embed-

---

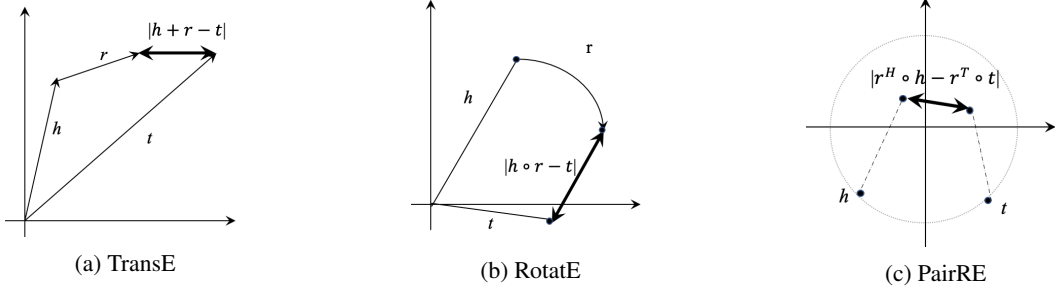[1] Hadamard product means entry-wise product.

Figure 1: Illustration of TransE, RotatE and PairRE when the entities stay in a plane. For PairRE, all entities are on the unit circle. The relation vectors project entities to different locations.

dings, $\{e_j\}_{j=1}^{\mathcal{E}}$ and all the relations' embeddings, $\{r_j\}_{j=1}^{\mathcal{R}}$.

Illustration of the proposed PairRE is shown in Figure 1. Compared to TransE/RotatE, PairRE enables an entity to have distributed representations when involved in different relations. We also find the paired relation vectors enable an adaptive adjustment of the margin in loss function, which alleviates the modeling problem for complex relations.

Let's take a 1-to-N relation as an example. We set the embedding dimension to one and remove the constraint on entity embeddings for better illustration. Given triples $(h, r, ?)$, where the correct tail entities belong to set $S = \{t_1, t_2, ..., t_N\}$, PairRE predicts tail entities by letting

$$\|h \circ r^H - t_i \circ r^T\| < \gamma,$$

where γ is a fixed margin for distance based embedding models and $t_i \in S$. The value of t_i should stay in the following range:

$$t_i \in \begin{cases} ((h \circ r^H - \gamma)/r^T, (h \circ r^H + \gamma)/r^T), \text{if } r^T > 0, \\ ((h \circ r^H + \gamma)/r^T, (h \circ r^H - \gamma)/r^T), \text{if } r^T < 0, \\ (-\infty, +\infty), \text{otherwise.} \end{cases}$$

The above analysis shows PairRE can adjust the value of r^T to fit the entities in S. The larger the size of S, the smaller the absolute value r^T. While models like TransE or RotatE have a fixed margin for all complex relation types. When the size of S is large enough, these models will be difficult to fit the data. For N-to-1 relations, PairRE can also adjust the value of r^H adaptively to fit the data.

Meanwhile, not adding a relation specific translational vector enables the model to encode several key relation patterns. We show these capabilities below.

Proposition 1. *PairRE can encode symmetry/antisymmetry relation pattern.*

Proof. If $(e_1, r_1, e_2) \in \mathcal{T}$ and $(e_2, r_1, e_1) \in \mathcal{T}$, we have

$$e_1 \circ r_1^H = e_2 \circ r_1^T \wedge e_2 \circ r_1^H = e_1 \circ r_1^T \\ \Rightarrow r_1^{H2} = r_1^{T2} \quad (2)$$

if $(e_1, r_1, e_2) \in \mathcal{T}$ and $(e_2, r_1, e_1) \notin \mathcal{T}$, we have

$$e_1 \circ r_1^H = e_2 \circ r_1^T \wedge e_2 \circ r_1^H \neq e_1 \circ r_1^T \\ \Rightarrow r_1^{H2} \neq r_1^{T2} \quad (3)$$

$\square$

Proposition 2. *PairRE can encode inverse relation pattern.*

Proof. If $(e_1, r_1, e_2) \in \mathcal{T}$ and $(e_2, r_2, e_1) \in \mathcal{T}$, we have

$$e_1 \circ r_1^H = e_2 \circ r_1^T \wedge e_2 \circ r_2^H = e_1 \circ r_2^T \\ \Rightarrow r_1^H \circ r_2^H = r_1^T \circ r_2^T \quad (4)$$

$\square$

Proposition 3. *PairRE can encode composition relation pattern.*

Proof. If $(e_1, r_1, e_2) \in \mathcal{T}$, $(e_2, r_2, e_3) \in \mathcal{T}$ and $(e_1, r_3, e_3) \in \mathcal{T}$, we have

$$e_1 \circ r_1^H = e_2 \circ r_1^T \wedge e_2 \circ r_2^H = e_3 \circ r_2^T \wedge \\ e_1 \circ r_3^H = e_3 \circ r_3^T \quad (5) \\ \Rightarrow r_1^T \circ r_2^T \circ r_3^H = r_1^H \circ r_2^H \circ r_3^T$$

$\square$

Moreover, with some constraint, PairRE can also encode subrelations. For a subrelation pair, $\forall h, t \in \mathcal{E} : (h, r_1, t) \rightarrow (h, r_2, t)$, it suggests triple (h, r_2, t) should be always more plausible than triple (h, r_1, t). In order to encode this pattern, PairRE should have the capability to enforce $f_{r_2}(h, r_2, t) \geq f_{r_1}(h, r_1, t)$.

Proposition 4. *PairRE can encode subrelation relation pattern using inequality constraint.*

Proof. Assume a subrelation pair r_1 and r_2 that $\forall h, t \in \mathcal{E}$: $(h, r_1, t) \rightarrow (h, r_2, t)$. We impose the following constraints:

$$\frac{\boldsymbol{r}_{2,i}^H}{\boldsymbol{r}_{1,i}^H} = \frac{\boldsymbol{r}_{2,i}^T}{\boldsymbol{r}_{1,i}^T} = \boldsymbol{\alpha}_i, |\boldsymbol{\alpha}_i| \le 1, \quad (6)$$

where $\alpha \in \mathbb{R}^d$. Then we can get

$$\begin{aligned}
&f_{r_2}(h, t) - f_{r_1}(h, t) \\
&= ||\boldsymbol{h} \circ \boldsymbol{r}_1^H - \boldsymbol{t} \circ \boldsymbol{r}_1^T|| - ||\boldsymbol{h} \circ \boldsymbol{r}_2^H - \boldsymbol{t} \circ \boldsymbol{r}_2^T|| \\
&= ||\boldsymbol{h} \circ \boldsymbol{r}_1^H - \boldsymbol{t} \circ \boldsymbol{r}_1^T|| - ||\boldsymbol{\alpha} \circ (\boldsymbol{h} \circ \boldsymbol{r}_1^H - \boldsymbol{t} \circ \boldsymbol{r}_1^T)|| \\
&\ge 0.
\end{aligned} \quad (7)$$

When the constraints are satisfied, PairRE forces triple (h, r_2, t) to be more plausible than triple (h, r_1, t). $\square$

Optimization. To optimize the model, we utilize the self-adversarial negative sampling loss (Sun et al., 2019) as objective for training:

$$\begin{aligned}
L = &-\log \sigma(\gamma - f_r(\boldsymbol{h}, \boldsymbol{t})) \\
&- \sum_{i=1}^{n} p(h_i', r, t_i') \log \sigma(f_r(\boldsymbol{h_i'}, \boldsymbol{t_i'}) - \gamma),
\end{aligned} \quad (8)$$

where γ is a fixed margin and σ is the sigmoid function. (h_i', r, t_i') is the ith negative triple and $p(h_i', r, t_i')$ represents the weight of this negative sample. $p(h_i', r, t_i')$ is defined as follows:

$$p((h_i', r, t_i')|(h, r, t)) = \frac{\exp f_r(h_i', t_i')}{\sum_j \exp f_r(h_j', t_j')}. \quad (9)$$

5 Experimental results

5.1 Experimental setup

We evaluate the proposed method on link prediction tasks. At first, we validate the ability to deal with complex relations and symmetry/antisymmetry, inverse and composition relations on four benchmarks. Then we validate our model on two subrelation specific benchmarks. Statistics of these benchmarks are shown in Table 2.

ogbl-wikikg2[2] (Hu et al., 2020) is extracted from Wikidata knowledge base (Vrandečić and Krötzsch, 2014). One of the main challenges for this dataset is complex relations. **ogbl-biokg**

Dataset	$|\mathcal{R}|$	$|\mathcal{E}|$	Train	Valid	Test
ogbl-wikikg2	535	2,500k	16,109k	429k	598k
ogbl-biokg	51	94k	4,763k	163k	163k
FB15k	13k	15k	483k	50k	59k
FB15k-237	237	15k	272k	18k	20k
DB100k	470	100k	598k	50k	50k
Sports	4	1039	1312	-	307

Table 2: Number of entities, relations, and observed triples in each split for the six benchmarks.

(Hu et al., 2020) contains data from a large number of biomedical data repositories. One of the main challenges for this dataset is symmetry relations. **FB15k** (Bordes et al., 2013) contains triples from Freebase. The main relation patterns are inverse and symmetry/antisymmetry. **FB15k-237** (Toutanova and Chen, 2015) is a subset of FB15k, with inverse relations removed. The main relation patterns are antisymmetry and composition. **DB100k** (Ding et al., 2018) is a subset of DBpedia. The main relation patterns are composition, inverse and subrelation. **Sports** (Wang et al., 2015) is a subset of NELL (Mitchell et al., 2018). The main relation patterns are antisymmetry and subrelation.

Evaluation protocol. Following the state-of-the-art methods, we measure the quality of the ranking of each test triple among all possible head entity and tail entity substitutions: (h', r, t) and (h, r, t'), $\forall h', \forall t' \in \mathcal{E}$. Three evaluation metrics, including Mean Rank(MR), Mean Reciprocal Rank (MRR) and Hit ratio with cut-off values $n = 1, 3, 10$, are utilized. MR measures the average rank of all correct entities. MRR is the average inverse rank for correct entities with higher value representing better performance. Hit@n measures the percentage of correct entities in the top n predictions. The rankings of triples are computed after removing all the other observed triples that appear in either training, validation or test set. For experiments on ogbl-wikikg2 and ogbl-biokg, we follow the evaluation protocol of these two benchmarks (Hu et al., 2020).

Implementation. We utilize the official implementations of benchmarks ogbl-wikikg2 and ogbl-biokg (Hu et al., 2020) for the corresponding experiments[3]. Only the hypeparameter γ and embedding dimension are tuned. The other settings are kept the same with baselines. For the rest experiments, we implement our models based on the implementation of RotatE (Sun et al., 2019). All hypeparam-

---

[2]ogbl-wikikg2 fixes a bug in test/validation negative samples from original ogbl-wikikg.

[3]Our code is available at:
https://github.com/alipay/KnowledgeGraphEmbeddingsViaPairedRelationVectors_PairRE

-	ogbl-wikikg2			ogbl-biokg		
Model	#Dim	Test MRR	Valid MRR	#Dim	Test MRR	Valid MRR
TransE	100	0.2622 ± 0.0045	0.2465 ± 0.0020	-	-	-
DistMult	100	0.3447 ± 0.0082	0.3150 ± 0.0088	-	-	-
ComplEx	50	0.3804 ± 0.0022	0.3534 ± 0.0052	-	-	-
RotatE	50	0.2530 ± 0.0034	0.2250 ± 0.0035	-	-	-
PairRE	100	$\mathbf{0.4849} \pm 0.0029$	$\mathbf{0.4941} \pm 0.0035$	-	-	-
TransE	500†	0.4256 ± 0.0030	0.4272 ± 0.0030	2000	0.7452 ± 0.0004	0.7456 ± 0.0003
DistMult	500†	0.3729 ± 0.0045	0.3506 ± 0.0042	2000	0.8043 ± 0.0003	0.8055 ± 0.0003
ComplEx	250†	0.4027 ± 0.0027	0.3759 ± 0.0016	1000	0.8095 ± 0.0007	0.8105 ± 0.0001
RotatE	250†	0.4332 ± 0.0025	0.4353 ± 0.0028	1000	0.7989 ± 0.0004	0.7997 ± 0.0002
PairRE	200	$\mathbf{0.5208} \pm 0.0027$	$\mathbf{0.5423} \pm 0.0020$	2000	$\mathbf{0.8164} \pm 0.0005$	$\mathbf{0.8172} \pm 0.0005$

Table 3: Link prediction results on ogbl-wikikg2 and ogbl-biokg. Best results are in bold. All the results except PairRE are from (Hu et al., 2020). † requires a GPU with 48GB memory. PairRE runs on a GPU with 16GB memory.

-	FB15k					FB15k-237				
Model	MR	MRR	Hit@10	Hit@3	Hit@1	MR	MRR	Hit@10	Hit@3	Hit@1
TransE†	-	0.463	0.749	0.578	0.297	357	0.294	0.465	-	-
DistMult◇	42	0.798	0.893	-	-	254	0.241	0.419	0.263	0.155
HolE	-	0.524	0.739	0.759	0.599	-	-	-	-	-
ConvE	51	0.657	0.831	0.723	0.558	244	0.325	0.501	0.356	0.237
ComplEx	-	0.692	0.840	0.759	0.599	339	0.247	0.428	0.275	0.158
SimplE	-	0.727	0.838	0.773	0.660	-	-	-	-	-
RotatE	40	0.797	0.884	0.830	0.746	177	0.338	0.533	0.375	0.241
SeeK	-	**0.825**	0.886	0.841	**0.792**	-	-	-	-	-
OTE	-	-	-	-	-	-	0.351	0.537	0.388	0.258
GC-OTE	-	-	-	-	-	-	**0.361**	**0.550**	**0.396**	**0.267**
PairRE	**37.7**	0.811	**0.896**	**0.845**	0.765	**160**	0.351	0.544	0.387	0.256
	± 0.4979	± 0.00077	± 0.00071	± 0.0011	± 0.0012	± 0.9949	± 0.00066	± 0.00093	± 0.00079	± 0.00097

Table 4: Link prediction results on FB15k and FB15k-237. Results of [†] are taken from (Nickel et al., 2016); Results of [◇] are taken from (Kadlec et al., 2017). Other results are taken from the corresponding papers. GC-OTE adds graph context to OTE (Tang et al., 2020).

Subrelation
(h, CoachesTeam, t) → (h, PersonBelongsToOrganization, t)
(h, AthleteLedSportsTeam, t) → (h, AthletePlaysForTeam, t)

Table 5: The added subrelation rules for Sports dataset.

Model	MRR	hit@1
SimplE	0.230	0.184
SimplE^{+}	0.404	0.349
PairRE	0.468 ± 0.003	0.416 ± 0.005
PairRE+Rule	$\mathbf{0.475} \pm 0.003$	$\mathbf{0.432} \pm 0.004$

Table 6: Link prediction results on Sports dataset. Other results are taken from (Fatemi et al., 2019).

Model	MRR	Hit@10	Hit@3	Hit@1
TransE	0.111	0.270	0.164	0.016
DistMult	0.233	0.448	0.301	0.115
HolE	0.260	0.411	0.309	0.182
ComplEx	0.242	0.440	0.312	0.126
SeeK	0.338	0.467	0.370	0.268
ComplEx-NNE	0.298	0.426	0.330	0.229
ComplEx-NNE-AER	0.306	0.418	0.334	0.244
PairRE	0.412	**0.600**	0.472	0.309
	± 0.0015	± 0.0006	± 0.0015	± 0.0027
PairRE+rule	**0.419**	0.599	**0.475**	**0.321**
	± 0.0010	± 0.0008	± 0.0008	± 0.0016

Table 7: Link prediction results on DB100k. All the results are taken from the corresponding papers.

eters except γ and embedding dimension are kept the same with RotatE.

5.2 Main results

Comparisons for ogbl-wikikg2 and ogbl-biokg are shown in Table 3. On these two large scale datasets, PairRE achieves state-of-the-art performances. For ogbl-wikikg2 dataset, PairRE performs best on both limited embedding dimension and increased embedding dimension. With the same number of parameters to ComplEx (dimension 100), PairRE

improves Test MRR close to 10%. With increased dimension, all models are able to achieve higher MRR on validation and test sets. Due to the limitation of hardware, we only increase embedding dimension to 200 for PairRE. PairRE also outperforms all baselines and improves Test MRR 8.7%. **Based on performances of baselines, the performance of PairRE may be improved further if embedding dimension is increased to 500**. Under the same experiment setting and the same number of parameters, PairRE also outperforms all baselines on ogbl-biokg dataset. It improves Test

MRR by 0.69%, which proves the superior ability to encode symmetry relations.

Comparisons for FB15k and FB15k-237 datasets are shown in Table 4. Since our model shares the same hyper-parameter settings and implementation with RotatE, comparing with this state-of-the-art model is fair to show the advantage and disadvantage of the proposed model. Besides, the comparisons also include several leading methods, such as TransE (Bordes et al., 2013), DistMult (Yang et al., 2014), HolE (Nickel et al., 2016), ConvE (Dettmers et al., 2018), ComplEx (Trouillon et al., 2016), SimplE (Kazemi and Poole, 2018), SeeK (Xu et al., 2020) and OTE (Tang et al., 2020). Compared with RotatE, PairRE shows clear improvements on FB15k and FB15k-237 for all evaluation metrics. For MRR metric, the improvements are 1.4% and 1.3% respectively. Compared with the other leading methods, PairRE also shows highly competitive performances. All these comparisons prove the effectiveness of PairRE to encode inverse and composition relations.

5.3 Further experiments on subrelation

We further compare our method with two of the leading methods ComplEx-NNE-AER and SimplE$^+$, which focus on encoding subrelation. These two methods add subrelation rules to semantic matching models. We utilize these rules as constraints on relation representations for PairRE. Two ways are validated. We first test the performance of weight tying for subrelation rules on Sports dataset. The rules ($r_1 \longrightarrow r_2$) are added as follows:

$$
\begin{aligned}
\boldsymbol{r}_2^H &= \boldsymbol{r}_1^H \circ cosine(\boldsymbol{\theta}), \\
\boldsymbol{r}_2^T &= \boldsymbol{r}_1^T \circ cosine(\boldsymbol{\theta}),
\end{aligned}
\tag{10}
$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$. The added rules are shown in Table 5. The experiments results in Table 6 show effectiveness of the proposed method.

Weight tying on relation representation is a way to incorporate hard rules. The soft rules can also be incorporated into PairRE by approximate entailment constraints on relation representations. In this section, we add the same rules from ComplEx-NNE-AER, which includes subrelation and inverse rules. We denote by $r_1 \xrightarrow{\lambda} r_2$ the approximate entailment between relations r_1 and r_2, with confidence level λ. The objective for training is then

changed to:

$$
\begin{aligned}
L_{rule} = L + \mu \sum_{\tau_{subrelation}} \lambda \mathbf{1}^T (\boldsymbol{r}_1^H \circ \boldsymbol{r}_2^T - \boldsymbol{r}_1^T \circ \boldsymbol{r}_2^H)^2 \\
+ \mu \sum_{\tau_{inverse}} \lambda \mathbf{1}^T (\boldsymbol{r}_1^H \circ \boldsymbol{r}_2^H - \boldsymbol{r}_1^T \circ \boldsymbol{r}_2^T)^2,
\end{aligned}
\tag{11}
$$

where L is calculated from Equation 8, μ is loss weight for added constraints, $\tau_{subrelation}$ and $\tau_{inverse}$ are the sets of subrelation rules and inverse rules respectively. Following (Ding et al., 2018), we take the corresponding two relations from subrelation rules as equivalence. Because $\tau_{subrelation}$ contains both rule $r_1 \rightarrow r_2$ and rule $r_2 \rightarrow r_1$.

We validate our method on DB100k dataset. The results are shown in Table 7. We can see PairRE outperforms the recent state-of-the-art SeeK and ComplEx based models with large margins on all evaluation metrics. With added constraints, the performance of PairRE is improved further. The improvements for the added rules are 0.7%, 1.2% for MRR and Hit@1 metrics respectively.

5.4 Model analysis

Analysis on complex relations

We analyze the performances of PairRE for complex relations. The results of PairRE on different relation categories on FB15k and ogbl-wikikg2 are summarized into Table 8. We can see PairRE performs quite well on N-to-N and N-to-1 relations. It has a significant lead over baselines. We also notice that performance of 1-to-N relations on ogbl-wikikg2 dataset is not as strong as the other relation categories. One of the reasons is that only 2.2% of test triples belong to the 1-to-N relation category.

In order to further test the performance of paired relation vectors, we change the relation vector in RotatE to paired vectors. In the modified RotatE model, both head and tail entities are rotated with different angles based on the paired
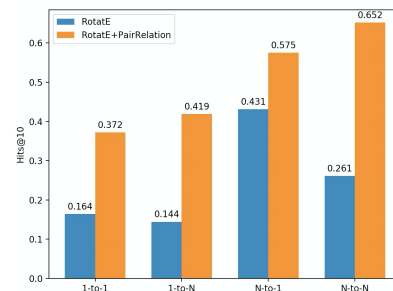

Figure 2: Performance comparison between RotatE and RotatE+PairRelation on ogbl-wikikg2 dataset.

-	FB15k(Hits@10)				ogbl-wikikg2(Hits@10)			
Model	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
KGE2E_KL(He et al., 2015)	0.925	0.813	0.802	0.715	-	-	-	-
TransE	0.887	0.822	0.766	0.895	0.074	0.063	0.400	0.220
ComplEx	**0.939**	0.896	0.822	0.902	**0.394**	**0.278**	0.483	0.504
RotatE	0.923	0.840	0.782	0.908	0.164	0.144	0.431	0.261
PairRE	0.785	**0.899**	**0.872**	**0.940**	0.262	0.270	**0.594**	**0.587**

Table 8: Experimental results on FB15k and ogbl-wikikg2 by relation category. Results on FB15k are taken from RotatE (Sun et al., 2019). The embedding dimensions for models on ogbl-wikikg2 are same to the experiments in Table 3, which is 100 for real space models and 50 for complex value based models.
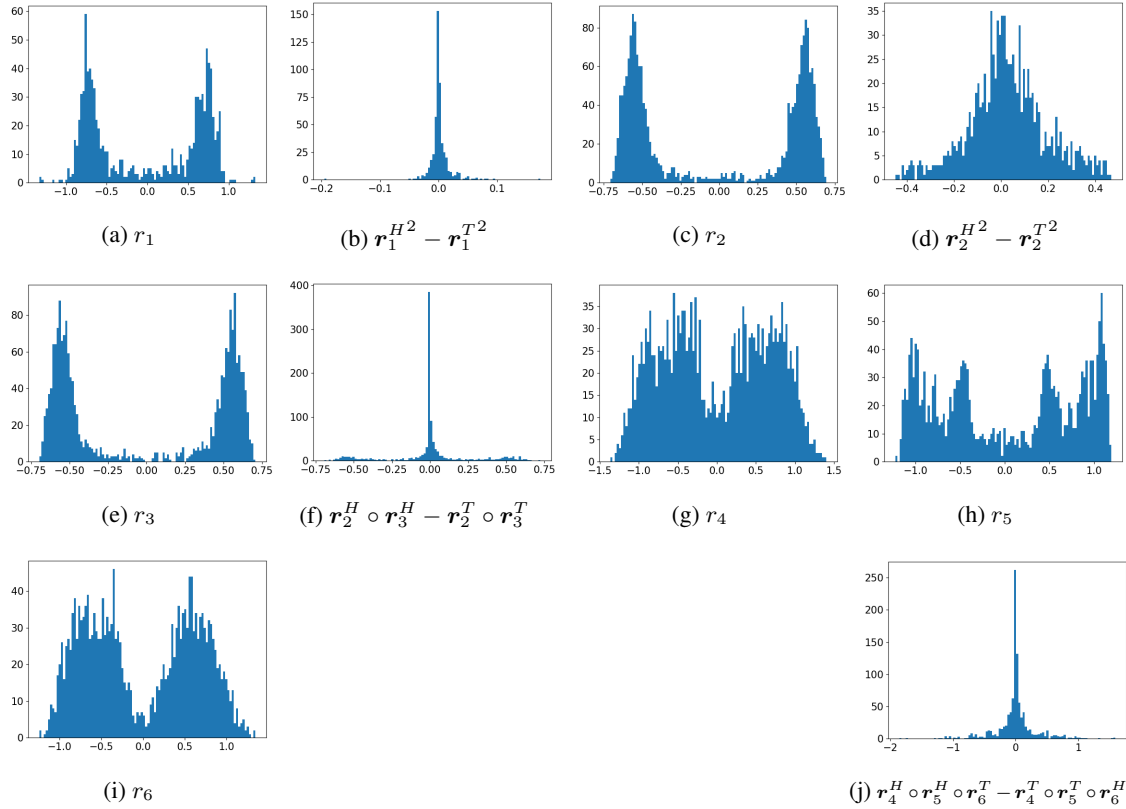

Figure 3: Histograms of relation embeddings for different relation patterns. r_1 is relation *spouse*. r_2 is relation $/broadcast/tv\_station/owner$. r_3 is relation $/broadcast/tv\_station\_owner/tv\_stations$. r_4 is relation $/location/administrative\_division/capital/location/administrative\_division\text{-}\_capital\_relationship/capital$. r_5 is relation $/location/hud\_county\_place/place$. r_6 is relation $base/areas/schema/administrative\_area/capital$.

relation vectors. This model can also be seen as complex value based PairRE. We name this model as RotatE+PairRelation. The experiment results are shown in Figure 2. With the same embedding dimension (50 in the experiments), RotatE+PairRelation improves performance of RotatE with 20.8%, 27.5%, 14.4% and 39.1% on 1-to-1, 1-to-N, N-to-1 and N-to-N relation categories respectively. These significant improvements prove the superior ability of paired relation vectors to handle complex relations.

Analysis on relation patterns

To further verify the learned relation patterns, we visualize some examples. Histograms of the learned relation embeddings are shown in Figure 3 .

Symmetry/AntiSymmetry. Figure 3a shows a symmetry relation *spouse* from DB100k. The embedding dimension is 500. For PairRE, symmetry relation pattern can be encoded when embedding r satisfies $r^{H2} = r^{T2}$. Figure 3b shows most of the paired elements in r^H and r^T have the same absolute value. Figure 3c shows a antisymmetry relation $tv\_station\_owner$, where most of the paired

elements do not have the same absolute value as shown in Figure 3d.

Inverse. Figure 3c and Figure 3e show an example of inverse relations from FB15k. As the histogram in Figure 3f shows these two inverse relations $tv\_station\_owner$ (r_2) and $tv\_station\_owner\_tv\_stations$ (r_3) close to satisfy $\boldsymbol{r}_3^H \circ \boldsymbol{r}_2^H = \boldsymbol{r}_3^T \circ \boldsymbol{r}_2^T$.

Composition. Figures 3g, 3h, 3i show an example of composition relation pattern from FB15k, where the third relation r_6 can be seen as the composition of the first relation r_4 and the second relation r_5. As Figure 3j shows these three relations close to satisfy $\boldsymbol{r}_4^H \circ \boldsymbol{r}_5^H \circ \boldsymbol{r}_6^T - \boldsymbol{r}_4^T \circ \boldsymbol{r}_5^T \circ \boldsymbol{r}_6^H$.

6 Conclusion

To better handle complex relations and tackle more relation patterns, we proposed PairRE, which represents each relation with paired vectors. With a slight increase in complexity, PairRE can solve the aforementioned two problems efficiently. Beyond the symmetry/antisymmetry, inverse and composition relations, PairRE can further encode subrelation with simple constraint on relation representations. On large scale benchmark ogbl-wikikg2 an ogbl-biokg, PairRE outperforms all the state-of-the-art baselines. Experiments on other well designed benchmarks also demonstrate the effectiveness of the focused key abilities.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

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. AcM.

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.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Conference on artificial intelligence*, CONF.

Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 165–180. Springer.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Boyang Ding, Quan Wang, Bin Wang, and Li Guo. 2018. Improving knowledge graph embedding using simple constraints. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 110–121, Melbourne, Australia. Association for Computational Linguistics.

Bahare Fatemi, Siamak Ravanbakhsh, and David Poole. 2019. Improved knowledge graph embedding using background taxonomic information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3526–3533.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2018. Knowledge graph embedding with iterative guidance from soft rules. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 623–632.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*.

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 (Volume 1: Long Papers)*, pages 687–696.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *AAAI*, volume 16, pages 985–991.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge base completion: Baselines strike back. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 69–74, Vancouver, Canada. Association for Computational Linguistics.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems*, pages 4284–4295.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhanava Dalvi, Matt Gardner, Bryan Kisiel, et al. 2018. Never-ending learning. *Communications of the ACM*, 61(5):103–115.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Thirtieth Aaai conference on artificial intelligence*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *ICML*, volume 11, pages 809–816.

Meng Qu and Jian Tang. 2019. Probabilistic logic neural networks for reasoning. In *Advances in Neural Information Processing Systems*, pages 7710–7720.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Yun Tang, Jing Huang, Guangtao Wang, Xiaodong He, and Bowen Zhou. 2020. Orthogonal relation transforms with graph context modeling for knowledge graph embedding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2713–2722, Online. Association for Computational Linguistics.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Quan Wang, Bin Wang, and Li Guo. 2015. Knowledge base completion using embeddings and rules. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Yanjie Wang, Rainer Gemulla, and Hui Li. 2018. On multi-relational link prediction with bilinear models.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*.

Wentao Xu, Shun Zheng, Liang He, Bin Shao, Jian Yin, and Tie-Yan Liu. 2020. SEEK: Segmented embedding of knowledge graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3888–3897, Online. Association for Computational Linguistics.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embedding. *arXiv preprint arXiv:1904.10281*.

Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y Chang, and Xiaoyan Zhu. 2012. Entity disambiguation with freebase. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 82–89.