# TripleRE: Knowledge Graph Embeddings via triple Relation Vectors

Long Yu[1]*, ZhiCong Luo[1,2], Deng Lin[1], HongZhu Li[1], HuanYong Liu[1], and YaFeng Deng[1]

[1] Qihoo360, China
[2] Xi'an Jiaotong University, China
{yulong,luozhicong,lihongzhu,lindeng,liuhuanyong,dengyafeng}@360. cn
zhicong_luo@stu. xjtu. edu. cn

**Abstract.** Knowledge representation is a classic problem in Knowledge graphs. Distance-based models have made great progress. The most significant recent developments in this direction have been those of Rotate[1] and PairRE[2], which focus on express relationships as projections of nodes. However TransX series Model(TransE[3], TransH[4], TransR[5]) expresses relationships as translations of nodes. To date, the problem of the Combination of Projection and translation has received scant attention in the research literature. Hence, we propose **TripleRE**, a method that models relationships by projections and translations. Compared with the other knowledge representation model, we achieve the best results on the ogbl-wikikg2 dataset.

**Keywords:** Knowledge Graph Embeddings · Distance base Model

## 1 Introduction

Knowledge representation(KR) is an important research branch of knowledge graph, which plays a essential role in the life cycle of downstream tasks, such as semantic parsing[6], named entity disambiguation[7], question answering[8], and etc. The previous research has established two main directions: translation distance model and bilinear model, mainly focusing on modeling knowledge triples with scoring functions. The Translation distance model expresses relationships as projections or translations of nodes. The bilinear model uses matrix decomposition to model triples. Our work mainly lies in the optimization of the Translation distance model. One major theoretical issue that has dominated the field for many years concerns how to model the complex relation. Rotate[1] expresses the relationship as the projection of the head node and expands it in the complex vector space. It can model symmetric, asymmetric, inverse and combination relationships. PairRE[2] divides the relationship into rh and rt, where rh

* Corresponding author
First author and second author contributed equally to this work and should be considered co-first authors

represents the projection of the head node, and rt represents the projection of the tail node, In addition to modeling the above relationships, it can also model sub-relationships. As the sota of the translation distance model, pairRE[2] removes the expansion of the complex vector space, is more concise than Rotate[1]. However, pairRE[2] still only regards the relationship as the projection of the node. We believe that both pairRE[2] and Rotate[1] does not take account of the relationship can learn as the translation part of the node. Our work is equivalent to making a complement on this basis. On the other hand, pairRE[2] can be regarded as our special case. We split the relationship into three parts. The projection part is the same as PairRE[2]. The translation part is learned by a separate parameter. In order to better learn translation features, we provide two TripleRE solutions. In addition to Nodepiece and RP techniques, our TripleRE achieve the best results. It is worth noting that our **TripleRE + NodePiece** solution achieves the best results on the ogbl-wikikg2 dataset.

## 2 Related Work

The knowledge graph is composed of entities and relationships, usually expressed in the form of triples, [head (head entity), relation (relationship of entities), tail (tail entity)], abbreviated as (h, r, t). The task of knowledge representation learning is to learn distributed representations of h, r, and t (also known as the embedding representation of the knowledge graph). The elements in the knowledge graph are embedded as dense low-dimensional vectors while retaining the original structure and connections. The embedded entities and relationships can complete a variety of knowledge graph tasks, such as semantic parsing, named entity disambiguation and question answering.

**1) Translation distance model** TransE[3] uses the translation invariance of the word vector embedding space found to express the relationship as the translation of the entity vector, thereby opening the door to the translation distance model. The model itself has the advantages of simple principles and fewer parameters, but it also cannot handle complex relationships and symmetry. Relationship and inverse relationship modeling and other issues. The key to the translation distance model is to choose an appropriate scoring function. A better scoring function will have a better performance in modeling complex relationships such as 1-N, N-1, N-N and relationship patterns such as symmetric/non-relationship, inverse relationship, combination relationship, and sub-relationship. The TransX series(TransE[3], TransH[4], TransD[9], TransR[5]) has made up for the shortcomings of TransE's inability to express complex relationships and symmetrical relationships, but there are still shortcomings such as the model is too complex and the expression of the relationship model is insufficient. RotatE[1] takes inspiration from Euler's formula and uses the rotation of the vector to express the relationship. At the same time, it uses complex embedding to model the inverse relationship. The model has achieved excellent results. PairRE[2] uses two-stage vectors to express the relationship to model the sub-relationship.

**2) Bilinear model** The bilinear model is also known as the semantic matching model. It uses similarity-based scores to measure the possibility of the fact that the triples are true by matching the latent semantics of entities and relationships in the embedding vector space. RESCAL[10] uses a vector to represent the embedding of the head entity and the tail entity, and the relationship is expressed as a matrix to model the interaction of the three. DisMult[11] imposes constraints on the relationship matrix and simplifies the calculation. ComplEX[12] embeds the head and tail entities and relationships into the complex space, so that it can better model the antisymmetric relationship.

**3) Others** Recently, an AutoML based model, AutoSF[13], has emerged. Through AutoML, it uses a certain search algorithm to search the score function with the best performance in the search space, and has achieved good results. Based on AutoSF[13], other scholars have proposed the method of combining NodePiece[14]with it. Each entity node is uniquely represented by anchor entity nodes and context relations, and a vocabulary is constructed for model training, which not only greatly reduces the amount of parameters, but also improves the effect of the model.

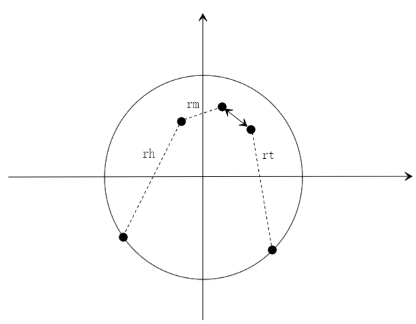


**Fig. 1.** Illustration of TripleREv1. For TripleREv1, all entities are on the unit circle. rh and rt project entities to different locations, rm is response for the translation part.

## 3 Methodology

We have proposed two plans for tripleRE. We call them v1 and v2 respectively, where v1 is the version we have submitted to the ogb benchmark.Illustration of the proposed TripleREv1 is shown in Figure 1.

**Loss function** In the KR task, the goal is to embed knowledge graph triples into a low-dimensional vector space. The loss we use is close to transE's loss[3]. loss functions can be written as the following formula:

$$\mathcal{L} = -(\bar{\sigma}(S) + \bar{\sigma}(S'))/2 \tag{1}$$

where S means positive score, S' means negative score. $\bar{\sigma}$ means take the average of $\sigma$. $\sigma$ means sigmoid function.

**TripleREv1 score function** pairRE split relation into $r_h$ and $r_t$. $r_h$ means the projection of head node. $r_t$ means the projection of tail node. we split relation into three parts, $r_h$, $r_m$ and $r_t$. $r_h$ and $r_t$ are the same as pairRE. $r_m$ is mainly responsible for the translation of the node. Score functions can be written as the following formula:

$$f_r(h,t) = -\|h \circ r^h - t \circ r^t + r^m\| \tag{2}$$

**TripleREv2 score function**: We try to add additional variables to rh and rt, so that our method can better learn the representation of triples.We added two parameters u and e, where u is a constant. In our experiment, we set it to 1, and e is the unit vector. The specific formula is as follows:

$$f_r(h,t) = -\|h \circ (r^h + u * e) - t \circ (r^t + u * e) + r^m\| \tag{3}$$

**TripleREv3 score function**: Inspired by InterHT[15], we found that the transformation of entities can learn more entity information, so we tried to add entity-based transformation representation in TripleRE, and we also divided the entity into three parts, namely $h_1$, $h_2$, $h_3$ and $t_1$, $t_2$, $t_3$, the specific formula is as follows:

$$f_r(h,t) = -\|h1 * t2 - t1 * h2 + h3 \circ (r1 + u * e) - t3 \circ (r2 + u * e) + r3\| \tag{4}$$

**More train step** We found that our model is not prone to overfitting, so we lengthened the training step and still gained benefits.

### 3.1 Implementation Detail

Specifically, we set learning rate 0.0005, step 700 thousand, the other Hyperparameters are the same as pairRE. pairRE need double relation dimension. In our score function, We expand the dimension of the relationship to three times.The implementation of TripleREv1 are shown in Figure 2.

```
re_head, re_mid, re_tail = torch.chunk(relation, 3, dim=2)

head = F.normalize(head, 2, -1)
tail = F.normalize(tail, 2, -1)

score = head * re_head - tail * re_tail + re_mid
score = self.gamma.item() - torch.norm(score, p=1, dim=2)
return score
```

**Fig. 2.** The implementation of score function.

### 3.2 Main Results

**General Performance** Table 1 shows model performance of each KR model.We can see that our method achieves the best results.

**Table 1.** On ogbl-wikikg2[16], our model achieved the best performance.

| | ogbl-wikikg2 | | ogbl-biokg | |
|---|---|---|---|---|
| Model | test MRR | valid MRR | test MRR | valid MRR |
| ComplEx-RP (50dim) | 0.6392 ± 0.0045 | 0.6561 ± 0.0070 | **0.8492 ± 0.0002** | **0.8497 ± 0.0002** |
| NodePiece + AutoSF | 0.5703 ± 0.0035 | 0.5806 ± 0.0047 | - | - |
| AutoSF | 0.5458 ± 0.0052 | 0.5510 ± 0.0063 | 0.8309 ± 0.0008 | 0.8317 ± 0.0007 |
| PairRE (200dim) | 0.5208 ± 0.0027 | 0.5423 ± 0.0020 | 0.8164 ± 0.0005 | 0.8172 ± 0.0005 |
| RotatE (250dim) | 0.4332 ± 0.0025 | 0.4353 ± 0.0028 | 0.7989 ± 0.0004 | 0.7997 ± 0.0002 |
| TransE (500dim) | 0.4256 ± 0.0030 | 0.4272 ± 0.0030 | 0.7452 ± 0.0004 | 0.7456 ± 0.0003 |
| ComplEx (250dim) | 0.4027 ± 0.0027 | 0.3759 ± 0.0016 | 0.8095 ± 0.0007 | 0.8105 ± 0.0001 |
| DistMult (500dim) | 0.3729 ± 0.0045 | 0.3506 ± 0.0042 | 0.8043 ± 0.0003 | 0.8055 ± 0.0003 |
| TripleRE (200dim) | 0.5794 ± 0.0020 | 0.6045 ± 0.0024 | - | - |
| TripleREv2 (200dim) | 0.6045 ± 0.0017 | 0.6117 ± 0.0008 | 0.8348 ± 0.0007 | 0.8360 ± 0.0006 |
| TripleREv2 + NodePiece | **0.6582 ± 0.0020** | **0.6616 ± 0.0018** | - | - |
| TripleREv3 + NodePiece | **0.6866 ± 0.0014** | **0.6955 ± 0.0008** | - | - |

## 4 Conclusions and Future Work

Our work shows that the distance-based knowledge representation model can also learn very competitive knowledge representation vectors. In order to enrich the expression of relationships and deal with complex relationships and multiple relationship patterns, we propose TripleRE, which represents relationships as two projections and one translation. At the same time, we seek a better expression of the relationship between the projection of the entity nodes at both ends and the translation after the projection of the head node. In large scale benchmark ogbl-wikikg2,We have achieved relatively good results. Our follow-up work will focus on how to better apply the knowledge representation vector to tasks downstream of the knowledge graph.

## References


1. Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.
2. Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. Pairre: Knowledge graph embeddings via paired relation vectors. *arXiv preprint arXiv:2011.03798*, 2020.
3. Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.
4. Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

5. Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.
6. Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544, 2013.
7. Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y Chang, and Xiaoyan Zhu. Entity disambiguation with freebase. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 82–89. IEEE, 2012.
8. Antoine Bordes, Jason Weston, and Nicolas Usunier. Open question answering with weakly supervised embedding models. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 165–180. Springer, 2014.
9. Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun 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 (Volume 1: Long Papers)*, pages 687–696, 2015.
10. Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, 2011.
11. Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.
12. Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.
13. Yongqi Zhang, Quanming Yao, Wenyuan Dai, and Lei Chen. Autosf: Searching scoring functions for knowledge graph embedding. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 433–444. IEEE, 2020.
14. Mikhail Galkin, Jiapeng Wu, Etienne Denis, and William L Hamilton. Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. *arXiv preprint arXiv:2106.12144*, 2021.
15. Baoxin Wang, Qingye Meng, Ziyue Wang, Dayong Wu, Wanxiang Che, Shijin Wang, Zhigang Chen, and Cong Liu. Interht: Knowledge graph embeddings by interaction between head and tail entities, 2022.
16. Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.