TECHNICAL REPORT OF WIKIKG90M-LSC

TECHNICAL REPORT

Weihua Peng
Harbin Institute of Technology
Shenzhen, China
pengwh.hit@gmail.com

Donghai Bian
University of Science and Technology of China
HeFei, China
bian2015@mail.ustc.edu.cn

Yanhui Huang
Southeast University
Nanjing, China
huangyanhui@seu.edu.cn

Guangzhi Sheng
University of Science and Technology of China
Suzhou, China
sgz@mail.ustc.edu.cn

Jian Sun
Tsinghua Unversity
Beijing, China
sunj18@mails.tsinghua.edu.cn

ABSTRACT

We describe our winning solution to the KDD Cup 2021 Open Benchmark Challenge. We mainly explore the following three implementation strategies in our proposed solution: a) more powerful representation vector learning, b) the complementarity between different models, c) statistical analysis based on data set. The optimal technical solution based on the three strategies gets great achievement in the test set: 11% higher than the highest official baseline (In the case of using only the model, we can improve the performance by 8% compared to the offical baseline. The third strategy may be caused by the bias in the evaluation dataset which may has limited usage in practice if the generation of evaluation dataset did include some bias which does not exist in real life. Details will be discussed laterly).

Keywords Link Property Prediction · Open Graph Benchmark · Knowledge Graph Completion

1 Introduction

Knowledge Graph (KG), as a special kind of graph structure with entities as nodes and relations as edges, is important to both data mining and machine learning, and has inspired various downstream applications, e.g., structured search question answering and recommendation. In KGs, each edge is represented as a triplet with form (head entity, relation, tail entity), denoted as (h, r, t). A fundamental issue is how to quantize the plausibility of triplets (h, r, t)s. KG embedding (KGE) has recently emerged and been developed as a promising method serving this purpose. Basically, given a set of observed triplets, KGE attempts to learn low-dimensional vector representations of entities and relations so that the plausibility of triplets can be quantized. Scoring function (SF), which returns a score for (h, r, t) based on the embeddings, is used to measure the plausibility. Generally, SF is designed and chosen by humans and it has significant effects on embeddings' quality. In this competition, we used multiple models for data representation learning, and effectively improved the generalization ability of the model through ensemble; Notice that when analysing the data set, we find that there may be some bias in the evaluation dataset ,which is instructional for models. Details is provided in Appendix.

2 Related Work

Distance based models (TDMs), Distance based models measure plausibility of fact triples as distance between entities. TransEBordes et al. [2013] interprets relation as a translation vector r so that entities can be connected, i.e., h + r ≈ t. TransE is efficient, though cannot model symmetry relations and have difficult in modeling complex relations. Several models are proposed for improving TransE to deal with complex relations, including TransHWang et al. [2014], TransRLin et al. [2015], TransDJi et al. [2015], TranSparseJi 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, TDMs are not fully expressive and their empirical performance is inferior to other models.

Bilinear models (BLMs) , RESCALNickel et al. [2011], ComplExTrouillon et al. [2016] , DistMultYang et al. [2014] and SimplEKazemi and Poole [2018] are all proved to be fully expressive when embedding dimensions fulfill some requirements. The fully expressiveness means these models can express all the ground truth existed in the data, including complex relations.

3 Approach

3.1 more powerful vector learning

Since the size of the model and the method of training can have a huge impact on the final performance, the experiments from multiple aspects are conducted to determine which type of method is more suitable for large-scale data sets. Specifically,

1. different representation dimension of entities ranging from 200 to 768
2. different number of negative samples ranging from 100 to 1000
3. different layer of Multilayer Perceptron (MLP) ranging from 1 to 3

3.2 the complementarity between different models

The knowledge learning by different model can be different, thus, multiple models can consist an system with the expert function which can greatly improve the performance of the model system. Motivated by this commonsense, we explored different performance of different models and then tried to linearly combine these models into a system. The following methods are what we have explored:

1. Models: TransE/ ComplEx/ DistMult/ SimplE/rotate[5]/ PairRE[8]/ AutoSF[9]. The score functions and the identifiable relationships are shown by Table 1.
2. Lost functions: Logsigmoid / Hinge/ Logistic / Focal Loss
3. Model combinations: weighted search based on grid search

Method	Score function	Relation patterns			
		Sym	Asym	Inv	Comp
TransE	$-\|\|h+r-t\|\|$	0	1	1	1
CompIEx	$h \times r \times \bar{t}$	1	1	1	1
DistMult	$h \times r \times t$	1	1	1	1
SimplE	$^{1/2}(h_h \times r \times t_t) +^{1/2} (h_t \times r_{-1} \times t_h)$	1	1	1	1
RotatE	$-\|\|h \circ r - t\|\|$	1	1	1	1
AutoSF	Combination of the above methods	1	1	1	1
PairRE	$\|\|h \circ r^H - t \circ r^T\|\|$	1	1	1	1

Table 1: Comparison of modeling capabilities of different scoring functions

3.3 statistical analysis based on data set

In addition to the optimization of the model and the experiments, we also conduct analyze about the data set, including the following aspects:

1. In the training set, there are 8Kw+ different heads while 2Kw+ different tails. The super node exists since there is a node which appears 36424411 times. The frequency of the top5 most frequently appeared super nodes with their corresponding entity are shown by Table 2.
2. Among the 1001 candidates in the validation set and the test set, except for the positive sample to be predicted, the other negative samples should be randomly selected. Therefore, the task does not belong to the fine-grained relationship classification.
3. According to the frequency of positive samples that appear in each relationship of the verification set, the relations with higher occurrences (such as 814), the frequency of positive samples is greater than 10, and the frequency of negative samples is less than 10. This is consistent with the above analysis(Please note that this is very unreasonable. Why is the frequency of positive cases is higher than negative cases? There may be two reasons here: First, the data set is a very coarse-grained edge relationship prediction, leading to the existence of super nodes; second, in the evaluation set, the organizer of the competition only gave 1001 candidate sets, among which 1000 negative cases should be randomly assigned).

Tail Entity Id	Freq
2529820	36424411
7186206	10628235
38242992	8739764
2024616	8085978
53132316	4965251

Table 2: The frequency of the tail node, in the training set

Based on the above analysis, we propose the following strategies:

1. For each relationship in the validation set/test set, count the frequency of its candidate entities.
2. If the frequency is higher than 5, remain the entity and the related frequency.
3. Sort in reverse order by frequency.
4. Select the first 700 entities as candidates of positive samples (if the number of entities greater than 5 is less than 700, all are selected) (noted as **Can**).
5. Prediction:
   (a) For each relationship, intersect the 1001 entities in its candidates with the **Can** set obtained in the Step 4.
   (b) If the size of the intersection is larger than 10, then all of the 10 entities will be sorted according to the frequency of occurrence and the result of the reverse order as the prediction.
   (c) If the size of the intersection is less than 10 but larger than 0, the remaining entities will be filled in according to the score predicted by the model.
   (d) If the intersection is empty, the prediction score of the model is used directly for sorting.

4 Experiments

4.1 Data set

The Open Graph Benchmark (OGB)Hu et al. [2020] is a collection of realistic, large-scale, and diverse benchmark datasets for machine learning on graphs. OGB Large-Scale Challenge (OGB-LSC)Weihua Hu [2021] encourages engineers to develop state-of-the-art graph ML models for modern massive datasets. Specifically, WikiKG90M-LSC is a knowledge graph, and the task is to impute missing triplets (link prediction). For more information about the data set, please refer to Table 3.

Dataset	Number of entities	Number of relationships	Number of edges
Train	87,143,637	1315	504,220,369
Val	-	855	1,700,584
Test	-	831	1,359,303

Table 3: Basic information of evaluation data

4.2 Experimental configuration

1. Hardware: 96 Intel(R) Xeon(R) Gold 6271C CPU@2.60GHz; memory 380G
2. Hyperparameters of the model: batch_size = [1000, 2000], lr = [0.05, 0.1, 0.25, 0.5], hidden_dim = [512, 768], neg_sample_num = [100, 200, 1000, 2000]; For **shallow**, max_step = 1000W, each MRR score are evaluated by 10% of the validation set for each 5w steps. For **concat**, max_step = 100W, each MRR score are evaluated by 10% of the validation set for each 2.5w steps and reserve the model with optimal performance.

4.3 Performance

4.3.1 More powerful vector learning

We conduct experiments from multiple aspects including dimension of entity, the number of negative samples and the layer number of MLP to determine the optimal method. See Table 4, 5, 6 for details.

Model	Dim	Batch_size	Neg_sample_num	Max_step	Optimal_step	Mrr(val)
TransE-shallow	200	1000	1000	1000W	900W	0.83
TransE-shallow	768	1000	1000	1000W	890W	0.88

Table 4: The impact of different dimensions of entity on the learning results: the higher the dimension, the better the prediction performance. Here we take transE as an example.

Model	Dim	Batch_size	Neg_sample_num	Max_step	Optimal_step	Mrr(val)
TransE-shallow	768	1000	100	1000W	830W	0.85
TransE-shallow	768	1000	200	1000W	790W	0.86
transE-shallow	768	1000	1000	1000W	890W	0.88

Table 5: The influence of different negative samples on the learning results: the more negative samples, the prediction performance. Here we take transE as an example.

4.3.2 the complementarity between different models

In section 4.3.1, the optimal setting are obtained: For shallow, entity_dim=768; for concat, entity_dim=512; a 2-layer of MLP is used; batch_size=2000 and the number of negative samples are 2000.

We tried to use 4 types of Loss to train the model including Logsigmoid Loss, Hinge Loss, Logistic Loss and Focal Loss. But the performance of Logistic loss and Focal Loss are not so great, and there is no advantage in model complementarity, thus they are not taken into consideration. In Table 7, we show the backbone model that we submit as the final model. It can be observed that the complementarity between TransE and CompIEX is clear. Details are shown in Table 9.

In addition to the official model, we also tried the two models with the highest MRR scores on the OGB list, AutoSF and PairRE. The corresponding effects are as follows. Overall, the performance is poor. See Table 8 for details

Model	Dim	Batch_size	Neg_samplenum	MLP_layernum	Max_step	Optimal_step	Mrr(val)
ComplEx	512	2000	2000	1	100W	27.5W	0.871
ComplEx	512	2000	2000	2	100W	35W	0.885
ComplEx	512	2000	2000	3	100W	45W	0.884

Table 6: For concat training method, the best performance is obtained when the layer of MLP is 2.

Model_id	Model	Dim	lr	gamma	seed	adv	pw	Optimal_step	Loss	Mrr(val)
A	TransE-shallow	768	0.1	10	0	True		890W	Log	0.881
B	ComplEx-concat	512	0.1	10	0	True		35W	Log	0.885
C	TransE-shallow	768	0.1	10	1	True		790W	Log	0.869
D	ComplEx-concat	512	0.1	10	1	True		15W	Log	0.876
E	ComplEx-concat	512	0.1	10	9	True		22.5W	Log	0.878
F	ComplEx-concat	512	0.1	10	77	True		30W	Log	0.881
G	DistMult-concat	512	0.1	10	13	True		47.5W	Log	0.885
H	DistMult-concat	512	0.1	20	47		True	65W	Hinge	0.871
I	SimplE-concat	512	0.1	10	77	True		40W	Log	0.884

Table 7: The submitted model

Table 8: the score of autoSF and pairRE where batch_size=2000, neg_sample_num=2000.

Model	Dim	lr	gamma	seed	Regularization_coef	adv	Optimal_Step	Loss	Mrr(val)
autoSF-shallow	768	0.05/0.1/0.25/0.5	50	0	1e-6	True	95W	Log	0.77
pairRE- shallow	512	0.1	10	0	1e-9	True	100W	Log	0.78
autoSF-concat	512	0.1	10	0	1e-6	True	25W	Log	< 0.2

In the initial experiment, we mainly used TransE and CompIEX. This is the verification of the complementarity of these two models. As can be seen from Table 9, the two models have good complementarity.

Model_id	Model_weight	Ensemble method	Mrr(val)	Mrr(test)
A	W1=0.881	A * W1 + B * W2	0.93	0.94
B	W2=0.885			

Table 9: Simple model complementarity verification, using the predicted MRR scores as weights

Subsequently, we follow the same idea to train another 7 models (see Table 7), and used the grid search to find the optimal linear weighted complementary model. After about 1000 evaluations, we selected the weight showed in Table 10

Model_id	Model_weight	Ensemble method	Mrr(val)
A	W1=1.0	A * W1 + B * W2 + C * W3 + D * W4 + E * W5 + F * W6 + G * W7 + H * W8 + I * W9	0.94
B	W2=0.3		
C	W3 =0.4		
D	W4 =0.3		
E	W5 =0.3		
F	W6 =0.1		
G	W7 =0.3		
H	W8 =0.8		
I	W9 =0.1		

Table 10: Using the model trained in Table 7 as the backbone, the weight values obtained using the grid search method

Model_id_list	Model_weight	Ensemble method	Mrr(val)	Mrr(test)
A, B, C D, E, F G, H, I	w1=1.0 w2=0.3 w3=0.4 w4=0.3 w5=0.3 w6=0.1 w7=0.3 w8=0.8 w9=0.1	A * w1 + B * w2 + C * w3 + D * w4 + E * w5 + F * w6 + G * w7 + H * w8 + I * w9 + Strategy	0.9781	0.9712

Table 11: Results on the full test set of the submitted model combination

4.3.3 statistical analysis based on data set

To illustrate the existence of data bias in the evaluation data set of the competition, we conducted the experimental comparison in Appendix A. It can be found that in this competition, inherent data bias does exist in the data set, in the prediction of some data, even simple statistical information can still get a higher MRR score, This means that no matter how it is used, as long as it is related to the frequency of occurrence of the entities in the test set, data biases can be exploited.

Finally, we use the construction strategy mentioned in Section 3.3, and combine the output with the final result of Table 10. In the end, the model has 3 points of improvement. See Table 11 for details.

5 Conclusion and Discussion

In our solution, three processes, model representation learning, evaluation of complementarity between each model, and analysis of the data set, play a big role in the final result of the game. The third strategy is very trick, and we are also very surprised by the existence of the data bias because the evaluation set can be hacked without touching the test set at training, but when deploying the link prediction method in practice, the small candidate sets will not be given. We are hornored to help the organizer find such problems.

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*, pages 1–9, 2013.

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.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

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.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, 2011.

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.

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.

Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. *arXiv preprint arXiv:1802.04868*, 2018.

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.

Hongyu Ren Maho Nakata Yuxiao Dong Jure Leskovec Weihua Hu, Matthias Fey. Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430*, 2021.

A Appendix

In this section, we focus on the problem of data bias and its impact. We hope that our discussion can help the event organizer to minimize this data bias in future competitions. We found it accidentally which may be suitable in the competition scenario but not in real life.

To illustrate the problem of data bias, we provide a detailed experimental comparison in Table 12. We only use rules based on our construction plan in Section 3.3 (without the model) to obtain the MRR score on the val set. We extracted the rules from the val set and the test set. At the same time, according to whether the rules and the candidate set of the data to be predicted have an intersection, 4 sets of comparative experiments are given. In the case that there must be an intersection during the evaluation process, our rule's MRR score in the val set can reach 0.99, regardless of whether the rule comes from the val set or the test set, which is enough to show that the evaluation set of this competition has a large data bias.

We are also very surprised by this, because the participants did not know the existence of the data bias. Therefore, if the contestant unintentionally uses this type of information (For example, some rules of manual (or automatic) mining), this makes them think that the model is working, but this may not be the case in reality.

val_dataset num	strategy	prediction process	MRR
1700584	rule(by val)	Use the strategy construction method in Section 3.3, but there are two differences: 1. If the size of the intersection is less than 10 but larger than 0, the rest will be randomly filled with the index in the candidate. 2. If the intersection is empty, The prediction will be filled randomly with the index in the candidate.	0.77
1326175	rule(by val)	Use the strategy construction method in Section 3.3, but there are three differences: 1. If the size of the intersection is less than 10 but larger than 0, the rest will be randomly filled with the index in the candidate. 2. If the intersection is empty, remove this data. 3. Only 1326175 pieces of data remain in the val set.	0.99
1700584	rule(by test)	Use the strategy construction method in Section 3.3, but there are two differences: 1. If the size of the intersection is less than 10 but larger than 0, the rest will be randomly filled with the index in the candidate. 2. If the intersection is empty, The prediction will be filled randomly with the index in the candidate.	0.43
742755	rule(by test)	Use the strategy construction method in Section 3.3, but there are three differences: 1. If the size of the intersection is less than 10 but larger than 0, the rest will be randomly filled with the index in the candidate. 2. If the intersection is empty, remove this data. 3. Only 742755 pieces of data remain in the val set.	0.99

Table 12: The effect comparison on the validation set only uses the method based on the data bias in the evaluation data set. (by val) indicates that the rule comes from the validation set, and (by test) indicates that the rule comes from the test set.