TIEG-Youpu's Solution for NeurIPS 2022 WikiKG90Mv2-LSC

Feng Nie
TIEG-Youpu
Tencent, Shenzhen
China
jannie@tencent.com

Zhixiu Ye
TIEG-Youpu
Tencent, Shenzhen
China
zhixiuye@tencent.com

Sifa Xie
TIEG-Youpu
Tencent, Shenzhen
China
sifaxie@tencent.com

Shuang Wu
TIEG-Youpu
Tencent, Shenzhen
China
seungwu@tencent.com

Xin Yuan
TIEG-Youpu
Tencent, Shenzhen
China
bartyuan@tencent.com

Liang Yao
TIEG-Youpu
Tencent, Shenzhen
China
dryao@tencent.com

Jiazhen Peng
TIEG-Youpu
Tencent, Shenzhen
China
brucejzpeng@tencent.com

Xu Cheng
TIEG-Youpu
Tencent, Shenzhen
China
alexcheng@tencent.com

Abstract

WikiKG90Mv2 in NeurIPS 2022 is a large encyclopedic knowledge graph. Embedding knowledge graphs into continuous vector spaces is important for many practical applications, such as knowledge acquisition, question answering, and recommendation systems. Compared to existing knowledge graphs, WikiKG90Mv2 is a large scale knowledge graph, which is composed of more than 90 millions of entities. Both efficiency and accuracy should be considered when building graph embedding models for knowledge graph at scale. To this end, we follow the retrieve then re-rank pipeline, and make novel modifications in both retrieval and re-ranking stage. Specifically, we propose a priority infilling retrieval model to obtain candidates that are structurally and semantically similar. Then we propose an ensemble based re-ranking model with neighbor enhanced representations to produce final link prediction results among retrieved candidates. Experimental results show that our proposed method outperforms existing baseline methods and improves MRR of validation set from 0.2342 to 0.2839.

1 Introduction

Web-scale knowledge graphs (KGs) is important in both data mining and machine learning [1, 4], and plays an important role in various downstream applications such as question answering, knowledge acquisition, and recommendation systems. A typical knowledge graph is composed of entities and various relational edges, where each edge is represented as a triplet of the form $(head\ entity, relation, tail\ entity)$ ((h, r, t) for short). Despite KGs contain rich structural information, they often suffer from knowledge incompleteness as world knowledge is updating rapidly[4]. Therefore, prediction over missing facts becomes a crucial task, also named as knowledge graph completion. Figure 1 depicts an example.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

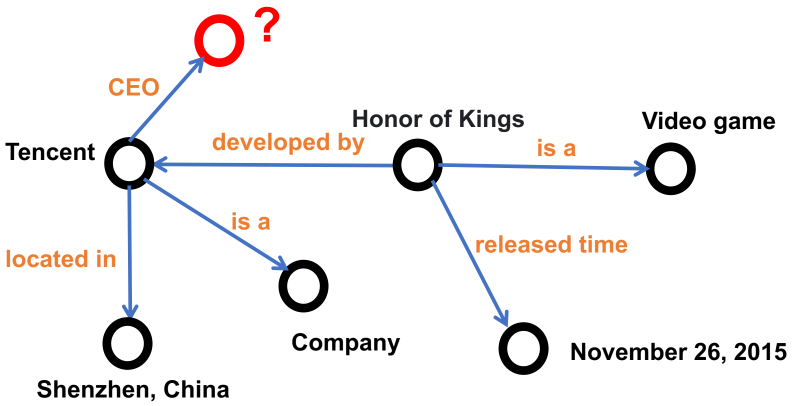

Figure 1: An example of link prediction in knowledge graph.

The 2022 NeurIPS releases the WikiKG90Mv2-LSC task, which focuses on correctly predicting missing facts in a large-scale KG. It is composed of more than 91 millions entities, a thousand relations, and 600 million triples. Directly building a complex graph embedding system and computing similarities with 91 millions of entities is time consuming. To this end, we follow the common strategy in large scale recommendation systems, by building a retrieval then re-ranking paradigm for link prediction. We make some novel modifications in both retrieval and re-ranking models. To retrieve most relevant candidates, we propose to leverage both structure and semantic information provided in knowledge graphs. Specifically, we train multiple PIE models [3] with different strategies and additionally define 11 structural paths to fully leverage graph structure. Moreover, to incorporate semantic information, we directly use the original textual representations [5] provided by 2022 NeurIPS and apply k nearest neighbor search to generate semantically relevant entities. For re-ranking models, we train three types of knowledge graph embedding models TransE[2], CompIEx [7], NOTE[6, 8] with different node presentations (i.e., randomly initialized embeddings, text feature embeddings, and graph-structure enhanced embeddings). Then we aggregate these results with a two step ensemble strategy. We conduct experiments on WikiKG90Mv2 dataset and improves MRR@10 of validation set from 0.2342 to 0.2839. Our team also achieves 0.2309 MRR in test-challenge dataset.

2 Methodology

Our proposed method is composed of two major components, a retrieval model and a re-ranking model. We will first introduce the retrieval model and then the re-ranking model in the following subsections.

2.1 Retrieval Model

Our retrieval model is composed of two steps. First, we design multiple retrieval models to exploit both structure or semantic information provided by KGs. Then, a priority infilling method is proposed to ensemble candidates retrieved by different retrieval models.

2.2 Structural & Semantic Enhanced Retrieval

PIE Retrieval: we follow the official baseline method [3] and apply fine grained typing aware inference (PIE for short) to generate structurally similar candidates. This method utilize neighborhood and relational types to search most relevant entities. Specifically, given an entity e and a relation r, the candidates are selected based on the posterior distribution as follows

$$p(e|r) = \frac{p(e)p(r|e)}{p(r)} \propto p(e)p(r|e), e \in N(e_q)$$

where $N(e_q)$ is the neighborhood entities of query entity e_q, $p(e)$ and $p(r)$ are prior distributions over entities and relations respectively (i.e., degrees). The entity typing model $p(r|e)$ is optimized with a self-supervised task by randomly masking several relations, and inference masked relations with remaining observed triplets in KG, more details can be find in [3].

To enable retrieve diverse candidates, we train two PIE models with different size of sampled neighbors (we find $N \in \{6, 10\}$ achieves similar recall accuracy during experiments). Moreover, retrieval is more difficult for frequent relations, we therefore apply an up-sampling strategy by assigning higher weights for frequent relations samples. The weights are set accordingly where higher weights for relations with lower recall performance[1].

Path Based Retrieval: Despite PIE model is able to leverage graph structure information, we find that it performs poor in some infrequent relations. Therefore, we propose to directly leverage the structure information in KGs by defining several walking paths from head entities and relations. We reserve candidates for tail entities with highest walking probabilities. We follow the walking path settings in NOTE [8], and define 11 head to tail paths (e.g., TH,, HT, TH-HT, HT-HT, TH-HT) and relation to tail paths (RT, RH, RT-HR-RT, RT-TR-RT, RH-HR-RT, RH-TR-RT). Take HT, TH-HT and RT-HR-RT as example, we define $F(h, r, t)$ to measure the co-occurrence possibility of a triplet (h, r, t) as

$$F_{HT}(h, r, t) = \frac{count(h, *, t)}{count(h, *, *)}$$

$$F_{TH-HT}(h, r, t) = \sum_{e_1} F_{TH}(e_1, *, h) \cdot F_{HT}(e_1, *, t)$$

$$F_{RT-HR-RT}(h, r, t) = \sum_{e_1, r_1} F_{RT}(*, r, e_1) \cdot F_{HR}(e_1, r_1, *) \cdot F_{RT}(*, r_1, t)$$

where $*$ denotes all possible entities or relations in KGs, $count(h, *, *)$ is the count of head entity h in the whole OGB training dataset, and $F(h, *, t) = \sum_r F(h, r, t)$. Finally, given (h, r), we can retrieve 11 lists of tail entities sorted by $F(h, r, t)$ functions and we set 20,000 as the upper bound of the length of each list.

Semantic Embedding Retrieval: To incorporate semantic information of entities, we compute semantic similarity using text feature embedding produced by MPNet[5] as following

$$F(h, r, t) = dist(\mathbf{e}_r^{MPNet}, \mathbf{e}_t^{MPNet})$$

where $dist(.)$ denotes euclidean distance. We apply k nearest search with product quantization version of FIASS to enable fast retrieval. We only set k to 1000 according to validation set.

2.3 Priority Infilling Ensemble

In order to ensemble results of recall methods mentioned above, we design a priority infilling ensemble method. First of all, we calculate the accuracy of each recall models m as

$$accuracy(m) = \frac{|S_{dev} \cap S_m|}{|S_m|}$$

where S_{dev} is the set of (h, r, t) in OGB valid set, S_m is the set of all triplets (h, r, t) retrieved by model m and $|\cdot|$ is the size of a set. We then set the priority of each model according to $accuracy(m)$. Then we aggregate results of different models with priority from high to low until yielding N candidates for each query (h, r).

2.4 Re-ranking Model

Our re-ranking model is composed of three steps. First, we propose a neighbor enhanced entity representation to aggregate first-order neighbor information directly for embedding initialization. Then, we apply several knowledge graph embedding methods to predict missing facts. Finally, we use an ensemble method to select most important graph embedding models for final predictions.

[1] Weights can be found in release code `https://github.com/CoderMusou/NeurIPS_2022_WikiKG90Mv2_TIEG-Youpu`

Neighbor Enhanced Entity Representations Conventional choice for entity embedding initialization can be randomly initialized embeddings, and semantic embeddings (i.e., official text feature embedding with MPNet[5]). However, structure information is ignored with above methods. To address this issue, we propose a neighbor enhanced semantic embedding method to directly leverage graph structure into initialization stage. Specifically, each entity embedding $\mathbf{e}_x^{struct}$ is represented by aggregating its first order neighbors entities as following.

$$\mathbf{e}_x^{struct} = \sum_{e_t \in N(e_x)} \mathbf{e}_t^{MPNet} \tag{1}$$

where $N(e_x)$ is the first order neighborhood entities of query entity e_x. In this way, three types of entity embedding are produced for graph embedding models to continue training.

Graph Embedding Models For graph embedding models, we adopt advance algorithms in different domains to encode entities and relations, including TransE, NOTE and ComplEx. With these graph embedding methods, the model is capable of inference over various kind of relations, such as 1-to-1, N-to-1, 1-to-N, reflective, inverse, symmetric and asymmetric relations.

TransE: Bordes et al. [2] interprets relation as a translation vector $\mathbf{r}$, so that entities can be connected with simple translation, i.e., $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. TransE is capable of capturing relation composition, but has difficulty in learning symmetric and asymmetric relations.

ComplEx: Trouillon et al. [7] embeds entities and relations into complex domain. ComplEx is simple and efficient in capturing symmetric, and asymmetric relations by following:

$$f(h, r, t) = Re(< \mathbf{w}_r, \mathbf{e}_h, \overline{\mathbf{e}_t} >)$$

where $<, >$ denotes hermitian product, $\mathbf{e}_h = Re(\mathbf{e}_h) + iIm(\mathbf{e}_h)$ is a vector that contains both real vector components and imaginary vector components. $\overline{\mathbf{e}_h} = Re(\mathbf{e}_h) - iIm(\mathbf{e}_h)$ refers to conjugate of vector $\mathbf{e}_h$. $Re(.)$ denotes taking the real vector component.

NOTE: is a normalized version of OTE model[6]. OTE models relations as group-based orthogonal transform embedding, which is able to model symmetric, inverse and compositional relations by simple transposing. The scoring function is defined as

$$f((h, r), t) = \sum_{i=1}^{K} (||\mathbf{s}_r^h(i) \phi(\mathbf{M}_r(i)) \mathbf{e}_h(i) - \mathbf{e}_t(i)||) \tag{2}$$

$$f(h, (r, t)) = \sum_{i=1}^{K} (||\mathbf{s}_r^t(i) \phi(\mathbf{M}_r(i))^T \mathbf{e}_t(i) - \mathbf{e}_h(i)||) \tag{3}$$

where $\mathbf{s}_r^h(i) = \frac{diag(exp(\mathbf{s}_r(i))}{||diag(exp(\mathbf{s}_r(i))||}$ and $\mathbf{s}_r^t(i) = \frac{diag(exp(-\mathbf{s}_r(i))}{||diag(exp(-\mathbf{s}_r(i))||}$ are the weights of relation matrix, ϕ is the Gram-Schmidt process.

Model Selection In this competition, we train TransE, ComplEx and NOTE with different entity embeddings (i.e., randomly initialized embeddings, text feature embeddings, graph structure enhanced embeddings) and hyper parameters. In order to combine results produced by multiple knowledge graph embedding models, we merge these results with a two-step ensemble strategy. First, we apply a greedy search to decide whether current model is qualified or not for final prediction. With the first filtering step, only 6 models are selected to produce the link prediction results. Then, we apply grid search to learn importance of each model.

3 Experiments

3.1 Experimental Details

WikiKG90Mv2 dataset contains three time-stamps: May,17th, June 7th, and June 28th of 2021 for training, validation and testing respectively. We only use training dataset to train recall and re-ranking models, and select hyper parameters based on validation set. For PIE recall model, we use the following hyper parameters for model training, where batch size is 512, context hops is 3, learning rate is 2e-3, hidden dimension is 1024, margin and gamma are set to 3, and number of sampled neighbors is 10.

Method	Main Parameter Settings	Recall @20000	Acc.
Structure Based Retrieval Models			
PIE_6	neighbor samples = 6	0.6008	**4.18e-5**
PIE_10	neighbor samples = 10	0.6044	3.73e-5
PIE_10_upsample	neighbor samples = 10 w/ up sampling	**0.6044**	3.79e-5
Rule-1	HT	0.0568	**5.46e-3**
Rule-2	TH	0.0488	7.42e-4
Rule-3	RT	0.582	6.43e-5
Rule-4	RH	0.0244	1.66e-6
Rule-5	TH-TH	0.0332	8.76e-5
Rule-6	HT-HT	0.1962	5.50e-4
Rule-7	TH-HT	0.0958	5.11e-4
Rule-8	RT-TR-RT	**0.6229**	3.32e-05
Rule-9	RT-HR-RT	0.3308	1.75e-05
Rule-10	RH-HR-RT	0.5641	2.99e-05
Rule-11	RH-TR-RT	0.2126	1.13e-05
Semantic Based Retrieval Models			
MPNet Recall	FIASS w/ product quantize centroids = 64, code size = 64	0.2394	1.197e-05
Ensemble Based Methods			
direct structural ensemble	majority voting	0.7013	-
structural priority infilling	ensemble structural results based on model priority	0.7136	-
structural and semantic priority infilling	ensemble all results based on model priority	**0.7361**	-

Table 1: Recall results based with structural and semantic enhanced retrieval models

Method	Main Parameter Settings	Validation MRR@10
TransE-0	-	**0.214**
TransE-1	batch size = 20480, negative sample size = 20480	0.2094
TransE-2	batch size = 20480, negative sample size = 20480, MPNet and randomly initialized embeddings w/ MLP layer	0.2114
TransE-3	neighbor enhanced embeddings and randomly initialized embeddings w/ MLP layer	0.1877
ComplEx	-	0.1649
NOTE-0	-	0.1561
NOTE-1	negative sample size = 1200	0.1648
NOTE-2	neighbor enhanced embeddings and randomly initialized embeddings w/ MLP layer	0.1654
NOTE-3	negative sample size = 1200, neighbor enhanced embeddings and randomly initialized embeddings w/ MLP layer	0.1592
Direct ensemble	grid search on above all models	0.28
Ensemble with Model Selection	first select best models (TransE-0,TransE-1,TransE-2,ComplEx, OTE-0, OTE-2) then grid search on model weights	**0.2839**

Table 2: Results of different graph embedding models using generated candidates on validation set.

For graph embedding models, we use the following hyper parameters as default NOTE setting, where batch size is 1000, hidden dimension is 200, orthogonal vector size is 20, learning rate is 0.1, regularization coefficient is 1e-9, negative sample size is 1000, learning rate for entity encoder is 4e-5, learning rate decay step is 2000. For TransE and ComplEx, which are simpler than NOTE, we therefore set a larger batch size (16384), hidden dimension (600) and negative sampling size (16384). During experiments, we find that NOTE achieves better results using the combination of text feature/structure enhanced embedding and randomly initialized embeddings, while TransE and ComplEx are more suitable with randomly initialized embeddings. Note that we use 4 A100 GPUs to train each graph embedding model.

3.2 Experimental Results

Retrieval Results: Table 1 reports retrieval results of different methods. Note that we only reserve 6 rule based recall models with higher accuracy than PIE's and limit the candidate size to 20,000 according to validation results. Table 1 show that structural enhanced retrieval models generally achieves better result than semantic based retrieval models. Combination of results produced by PIE models and rule based models can lead to huge improvements. When incorporating semantic retrieval methods, the results can be improved further.

Re-ranking Results: Table 2 presents the MRR@10 of different graph embedding methods based on generated candidates. TransE performs best compared to some state-of-art graph embedding models such as ComplEx and NOTE on WikiKG90Mv2 surprisingly. The result also shows the necessity of ensemble of these different graph embedding models. Combinations of these models leads to significant improvement, which indicates that different graph embedding models are good at prediction on different types of relations. Moreover, the results can be further improved with appropriate model selection strategy.

4 Conclusion

In this paper, we present our solution for link prediction task on WikiKG90Mv2 dataset. Our proposed method follows the retrieval and re-ranking paradigm, and makes some novel modifications in both retrieval and re-ranking step. Specifically, for candidate retrieval, we propose to leverage structural and semantic information during retrieval to select relevant candidates. Moreover, we propose a priority infilling ensemble technique to merge candidate results produced by different retrieval models. For re-ranking step, we first enhance the original node representation by aggregating first order neighbors and then train multiple state-of-art graph embedding models including TransE, ComplEx and NOTE. Then we ensemble these results with a model selection strategy and grid search. The experimental results show effectiveness of our proposed method. In the future, we will consider improving the efficiency of candidate retrieval models.

References

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-102-6.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[3] Linlin Chao, Xiexiong Lin, Taifeng Wang, and Wei Chu. Pie: a parameter and inference efficient solution for large scale knowledge graph embedding reasoning, 2022.

[4] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610. ACM, 2014.

[5] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/c3a690be93aa602ee2dc0ccab5b7b67e-Paper.pdf`.

[6] Yun Tang, Jing Huang, Guangtao Wang, Xiaodong He, and Bowen Zhou. 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, July 2020. Association for Computational Linguistics.

[7] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2071–2080, New York, New York, USA, 20–22 Jun 2016. PMLR.

[8] Hui Zhong Huijuan Wang Siming Dai Zhengjie Huang Yunsheng Shi Shikun Feng Weiyue Su, Zeyang Fang. Note: Solution for kdd-cup 2021 wikikg90m-lsc. 2021.