Knowledge Graph Completion Model Based on Con-distillation

Min Xia*, Tao Zhang[a], Zhonghai Wu[b]

School of Software and Microelectronics Peking University, Beijing, China

*mxia@ss.pku.edu.cn; [a]2053175467@qq.com; [b]wuzh@pku.edu.cn

Abstract. To improve the ability of the knowledge graph completion model to predict long-tail entities and keep its prediction indices for popular entities, this paper proposes a knowledge graph completion model based on co-knowledge distillation. Overall, the model consists of three parts as shown below. In detail, the first part is the neighborhood-information-based transformer model (NT), which aims to learn the general representation of the current knowledge graph. In this regard, NT takes the first-order subgraph of the knowledge graph as input to learn the representation of entities from neighboring nodes, subsequently learning the task target of link prediction through masking training. The second part is the relational-path-based bidirectional encoder representations from transformers (BERT) model, hereinafter the PB model, which expands the retrieval range for long-tail entity information by learning multiple relational paths of triples, thereby improving the characterization ability of long-tail entities. Lastly, the proposed model enables the two foregoing models to learn from each other in a lightweight co-knowledge distillation way, so that their prediction ability for popular entities and long-tail entities can be improved simultaneously.

Keywords: Transformer; Knowledge Graph; Bidirectional Encoder Representation from Transformers

1 INTRODUCTION

Knowledge graph completion (KGC) exposes two crucial defects, one of which is that the existing completion algorithms generally fail to take long-tail entities into consideration. In other terms, the model can solely learn a limited degree of feature information for nodes with few connected edges. Another defect can be summarized as the dependence of numerous models on inductive learning algorithms. Hence, in cases where the data of the test set and the data of the training set are different in distribution, the model is typically difficult to exert a relatively excellent effect. Recently, extensive research has revealed that the transformer is superior to other models in the fields of natural language processing and graph structure data. Leveraging the stacking of self-attention mechanisms, the transformer can mine some laws that are difficult to perceive by human beings in massive data, with minimal consideration for the inherent distributional characteristics of the data. At this point, bidirectional encoder representation

A. Haldorai et al. (eds.), *Proceedings of the 2024 3rd International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID 2024)*, Atlantis Highlights in Intelligent Systems 11,
https://doi.org/10.2991/978-94-6463-490-7_61

from transformers (BERT), built upon the transformer encoder as its primary architecture, can obtain information beyond the confines of the training dataset through pre-training. Given this, this paper designs a KGC algorithm of two-way distillation (i.e., con-distillation) based on the transformer and BERT models. Among them, the transformer model primarily learns the neighborhood information of entities, thereby capturing the direct semantics of triples, whereas the BERT model learns the semantic representation and relationship path information of entities, employing the advantages of the pre-training model to mine the semantic information of long-tailed entities. Simply put, both of them are capable of learning from each other through con-distillation, thus enhancing their respective characterization ability.

2 RELATED RESEARCH

The knowledge graph embedding (KGE) serves as a primary method to achieve the KGC at present. By embedding the entities and relationships within the knowledge graph triplet into the vector space R^d, it can be utilized to investigate the relationships between vectors belonging to the same triplet. Regarding the KGC, the KGE-related research focus can be summarized as which vector embedding method can fulfill the matching of the same entity in different triples. In this connection, the reliability of the prediction vector is evaluated by the score function, which is generally the operational expression of the embedding vector of the triple entities h and t as well as relationship r, with the specific calculation method determined by the embedding space and the model selected. Additionally, the long tail effect on data sets is also prevalent in the field of machine learning. Consequently, these samples with few features are usually difficult to be captured by the model. Moreover, the accumulation of such samples greatly interferes with the effect of the model. The KGC is facing such a challenge. GMatching [1] marks the initial work to address the related problems of one-time KGC. First of all, a neighbor encoder is proposed to generate better entity embedding by leveraging local graph structure. Likewise, FSRL [2] adopted the same idea, extending GMatching to small sample data. Furthermore, based on the heterogeneous graph structure and attention mechanism, a relation-aware heterogeneous neighbor encoder is introduced to enhance entity embedding, facilitating the model to encode the different influences of diverse neighbors on task relationships. FAAN [3] proposed an adaptive attention neighbor encoder to model the entity embedding with one-hop entity neighbors, following TransE to model task relations embedding as transformations between head and tail entity embeddings. Moreover, GEN [4] studied an off-map KGC scenario to predict the relationship between invisible entities. REFORM [5] proposed an error perception module to control the negative impact exerted by errors on KGC. Slightly different from the original KGC, it predicts and queries the missing relationship categories of entity pairs from a minority of relationship categories.

Besides, a plurality of scholars employ language models to investigate the capture of long-tail entities. InductiveE [3], for instance, proposed a common-sense KG link prediction method, which can address invisible entities by using text entity descriptions, thereby realizing inductive learning by directly constructing representations from entity

descriptions. KG-BERT [6] was developed to tackle this challenge. By transforming the three head entities, relation and tail entities into text sequences, it further takes triple prediction as a downstream text classification task. Notably, BERT[9] [10] can be fine-tuned according to a given training triplet. On the same note, BERTRL [7] proposed a method with triple prediction as the downstream text classification task of BERT, which uses a single triple and possible paths connecting two entities to fine-tune BERT, thus realizing explicit reasoning.

3 MODEL DESIGN

To improve the ability to predict the missing tail entities, this research designed the overall model framework as outlined in Figure 1. Specifically, the first part is based on Transformer, while the second part is based on BERT's missing entity prediction model. Through extracting the related paths of target entities, a group of related long paths is further determined.

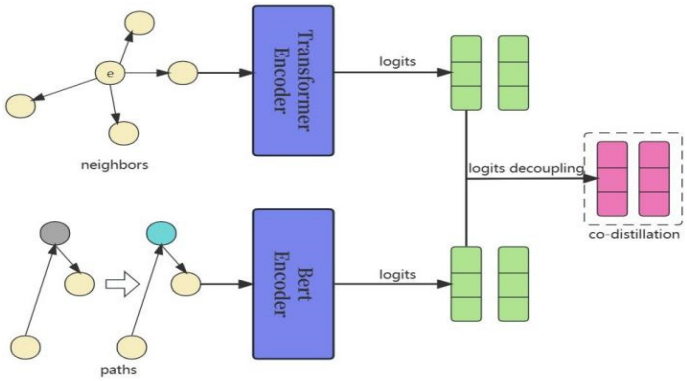

Fig. 1. General Framework of Con-distillation

3.1 Embedded Model Based on Neighborhood Information

Transformer has made great progress within the KGC field. Currently, a mainstream method to address KGC-related problems with the transformer is to treat graph data as a token sequence with position coding, thereby masking the influence of other neighboring entities. Regarding link prediction, however, the neighboring entities of one node usually contain the prediction of the important information of the next node, provided that the relationship between entities should be taken into account. Hence, a KGC model based on neighborhood information, namely the neighborhood-information-based transformer model (NT model), is proposed with neighborhood subgraphs as the input of the transformer. Figure 2 depicts the transformer encoder of the NT model, which primarily comprises a multi-head attention layer as well as a mixture-of-expert (MoE) [8] feedforward neural network (FNN) layer controlled by sparse gates. With the collected subgraph around the central node as input, this network randomly hides a

peripheral node or relation in each iteration, which is replaced by [MASK], subsequently training the target and peripheral triples simultaneously. For the convenience of the transformer to fully mine the information of the relation r, this research adopts a method similar to KGE to embed the relation r and entity in the same dimension d when processing the input of the NT model, both of which are denoted as x_e. Concurrently, a certain dimension is selected as a mark bit to distinguish entities from relations. Through multi-layer perceptron, the subgraph generates a set of embedded vectors, thus calculating the attention coefficient within the multi-head attention structure, as shown in Equations (1):

$$\text{Attention} = \text{softmax}\left(Q^{\top}K/\sqrt{d_h}\right)V \tag{1}$$

where $Q = \mathbf{x}_e\mathbf{W}_Q$; $\{K,V\} = \|_{n\in\mathcal{N}(e)}\ \mathbf{x}_n\{\mathbf{W}_K,\mathbf{W}_V\}$; attention matrix $\mathbf{W}_{\{Q,K,V\}} \in \mathbb{R}^{d\times d_h}$; $\|$ stands for join operation; $d_h = d/H$; and, H denotes the number of heads of self-attention.

In this foundation, the vector x_e^n output by the n-th attention network is input into the FNN, which is employed to calculate the projected attention-weighted output of the matrix, as shown in Equations (2):

$$\text{FFN} = \sigma(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{2}$$

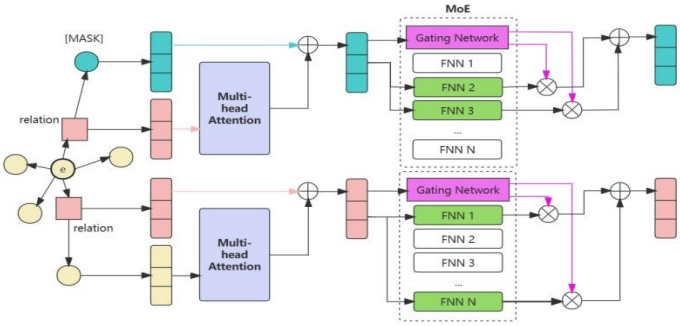

Fig. 2. Internal Structure of NT-Transformer

The training of NT consists of two parts, one of which is the learning of neighborhood subgraphs, while the other is the link prediction. As shown in Figure 3, which illustrates the training framework of the NT model, this model is based on the idea of recursive entity reconstruction, enabling the model to observe more predictions while maintaining robustness to the incompleteness and sparsity of graphs. With the prediction of triple (h, r, ?) as an example, this research randomly selects triples in the training set and replaces entity t with a placeholder [MASK] to determine the representation of t from the incomplete triples (h, r,[MASK]), intending to enable the model to acquire the ability to predict missing entities during the training process. By inputting the randomly initialized input embedding $\left(\mathrm{E}_h^0, \mathrm{E}_r^0, \mathrm{E}_{[\mathrm{MASK}]}^0\right)$ into the Transformer, the output representation of [MASK] is further given by as shown in Equations (3):

$$E^{n}_{[MASK]} = Transformer\left(E^{0}_{h}, E^{0}_{r}, E^{0}_{[MASK]}\right) \tag{3}$$

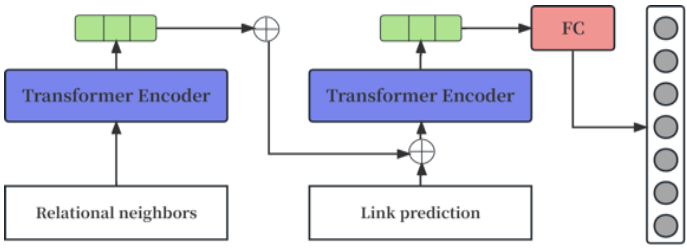

Fig. 3. Architecture of KGC Model Based on Neighborhood Information

3.2 KGC Model Based on Relational Paths

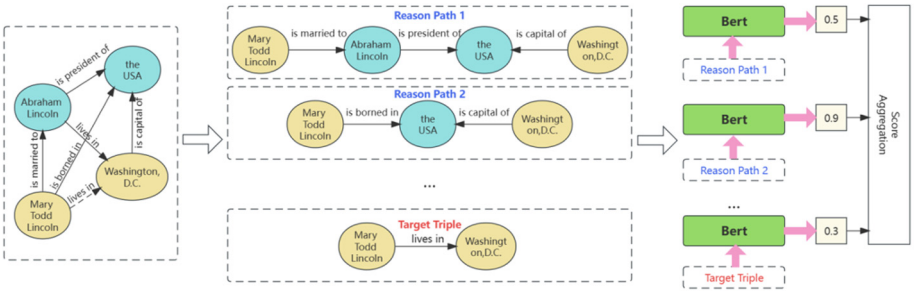

Fig. 4. Training Principle of PB Model

Figure 4 depicts the training principle of BERT model based on relational paths. This research adopts the method of sampling to obtain the multi-segment path $path_{(h,?,t)}$ from the head entity h to the tail entity t within the knowledge graph. To extend the prediction to entities, this research regards the relation r across entities as entities. In this case, the triplet is represented as (e_h, e_r, e_t). When the path is used as the input of the model, the path can be expressed as $(e_1, e_2, \ldots, e_n)$.

PB model employs the form of the relational path in terms of prompt message. In other words, regarding each entity or relation, it obtains a fixed-length path $path_{(e_0,e_n)}$ by random walk sampling, thereby realizing the advanced training of the prompt path. Given that link prediction encompasses the prediction of entities and relations, the PB model adopts the following path selection strategies when addressing the foregoing two problems:

For one thing, regarding entity prediction, taking （h, r, ?） as an example, the PB model initiates random walk sampling from the current entity h until the maximum path length. In cases where the path arrives at r, the sampling is terminated in advance.

For another, regarding relation prediction, the PB model implements bidirectional random walk sampling from h and t until 1/2 of the maximum path length. In cases of two nodes encountering, the sampling is terminated in advance.

Given that each path is input as linear knowledge in this research, each pair of target paths and prompt path acquires a separate score. It is therefore imperative to aggregate all instances within a group. As such, the prediction of missing entities can be determined as shown in Equations (4):

$$\mathrm{E}_{[\mathrm{MASK}]} = \max\left(\mathrm{BERT}\left(path_{(e_0,e_n)}\right)\right) \tag{4}$$

where $BERT$ denotes the whole process of prediction of [MASK] by the BERT model, encompassing word vector embedding using pre-trained parameters as well as the transformer encoder. To obtain the predicted entity, the name of the entity needs to be added to the BERT vocabulary as a new marker. Given the absence of external knowledge, the random initial embedding of new markers greatly affects the training effect, leading to a rather slow model convergence. Consequently, this paper applies the method proposed by Lv et al., which obtains the ranking of candidate entities from other models and inputs the top X entities into the proposed model, thereby recalculating the score and reordering. The PB model framework is shown in Figure 5. Eventually, the loss function $\mathcal{L}_{\mathrm{Path}}$ of the PB model is defined as shown in Equations (5):

$$\mathcal{L}_{\mathrm{Path}} = \sum_{(h,r,t)\in\mathcal{T}} \left(\mathrm{CrossEntropy}(\mathbf{P}_h^T, \mathbf{L}_h) + \mathrm{CrossEntropy}(\mathbf{P}_t^T, \mathbf{L}_t)\right) \tag{5}$$

where $\mathbf{P}_h^T$ and L_h represent the logit and label of the entity, whereas $\mathbf{P}_t^T$ and L_t respectively represent the predicted logit and label of the predicted entity.

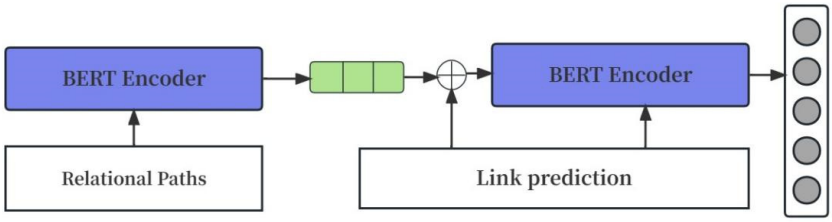

Fig. 5. Framework of PB Model

3.3 Co-knowledge Distillation

Upon obtaining the predicted logits $\mathbf{P}^T$ and $\mathbf{P}^S$ of the two models, this research arranges all entities from high to low according to $\mathbf{P}^S$ in the case of NT as a teacher model, selecting the half with the higher score as the entity. Following this, the logits of prediction entities selected from $\mathbf{P}^T$ and $\mathbf{P}^S$ are denoted by $\mathbf{P}^{T_1}$ and $\mathbf{P}^{S_1}$ respectively. Notably, the decoupling loss functions of $\mathbf{P}^{T_1}$ and $\mathbf{P}^{S_1}$ are determined as shown in Equations (6):

$$\mathcal{L}_{KD}(\mathbf{P}^{S_1}, \mathbf{P}^{T_1}) = \alpha \mathrm{KL}\left(\mathbf{P}_b^{S_1} \parallel \mathbf{P}_b^{T_1}\right) + \beta \mathrm{KL}\left(\widehat{\mathbf{P}}^{S_1} \parallel \widehat{\mathbf{P}}^{T_1}\right) \tag{6}$$

where $\mathbf{P}_b^{S_1}$ and $\mathbf{P}_b^{T_1}$ respectively denote the binary classification probability of the target entity by the student and teacher models; $\widehat{\mathbf{P}}^{S_1}$ and $\widehat{\mathbf{P}}^{T_1}$ denote the probability of

excluding the target entity from $\mathbf{P}_b^{S_1}$ and $\mathbf{P}_b^{T_1}$, respectively; and, α, coupled with β, represents the hyperparameters of DKD. In the case where PB serves as a teacher model, the same method can be utilized to determine $\mathbf{P}_b^{S_2}$ and $\mathbf{P}_b^{T_2}$. The total loss function of the two models is the sum of the distillation part and cross-entropy, as shown in Equations (7):

$$\begin{aligned} \mathcal{L}_{\mathrm{PB}} &= \mathrm{A}\mathcal{L}_{KD}\left(\mathrm{P}_t^{S_1}, \mathrm{P}_t^{T_1}\right) + (1-\mathrm{A})\mathrm{CrossEntropy}(\mathbf{P}_t^{T}, \mathbf{L}_t) \\ \mathcal{L}_{\mathrm{NT}} &= \mathrm{B}\mathcal{L}_{KD}\left(\mathbf{P}_t^{T_2}, \mathbf{P}_t^{S_2}\right) + (1-\mathrm{B})\mathrm{CrossEntropy}(\mathbf{P}_t^{S}, \mathbf{L}_t) \end{aligned} \tag{7}$$

where A and B represent the parameters used by the two models to balance their respective terms, thereby realizing the joint optimization of the two models. To put it another way, each batch is input into two models simultaneously, with their losses being calculated and updated as per the same data. The pseudo-code of the algorithm during the model training process is presented as follows:

Algorithm 1: Pseudo-code for training NT-PB algorithm

Input: Triple set $\mathcal{S}(h,r,t)$, entity adjacency matrix $\mathcal{M}Neighbor$, sampling path set $\mathcal{S}path$;
Output: List of candidate entities missing triplet $\mathscr{L}is$.
1: Initializing network parameters and entity embedding;
2: Adding the missing marker [MASK] and segmentation marker to the training set;
3: for $epoch \leftarrow 1$ to max_epoch do;
4: for $step \leftarrow 1$ to max_step do;
5: Extracting a batch of training batches $batch$;
6: for (h, r, $[MASK]$) in $batch$ do;
7: Calculating the logits and losses of NT according to Equation (3.13);
8: Calculating the logits and losses of PB according to Equation (3.15);
9: Calculating the total loss according to Equation (3.17);
10: end for;
11: Updating model parameters by gradient descent method;
12: end for;
13: Validating the validation set;
14: if fitting with the validation set then;
15: break;
16: end if;
17: end for.

Ultimately, the mutual learning between NT and PB is realized by the con-distillation module. Given that NT is prone to obtain the structural information in KG, this research takes NT as the teacher model and PB as the student model in the first stage of training, so that PB can quickly acquire the structural knowledge that BERT is difficult to learn. Subsequently, in the case that the loss of the PB model tends to be stable, the roles of the two models are further reversed. In other words, NT is selected as the student model and PB as the teacher model, enabling NT to learn more about the representation of long-tail entities.

4 EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Evaluation Indices

Major evaluation indices of link prediction encompass the mean ranking (MR), mean reciprocal ranking (MRR), and hit@n. Here, MR represents the average of the rankings of all correctly predicted examples, which is determined as shown in Equations (8):

$$MR = \frac{1}{|S|}\sum_{i=1}^{|S|} rank_i \tag{8}$$

where |S| represents the total number of predicted triples, while $rank_i$ represents the ranking of the predicted sequence where the correct answer of a triplet is located. A higher ranking of correct prediction implies a smaller MR value. Consequently, a smaller MR value indicates a higher accuracy of the prediction results of the model.

MRR serves as an internationally common mechanism for evaluating search algorithms, which primarily matches the results according to the search ranking with scores of 1, 1/2, ..., 1/n. Drawing upon this idea, the KGC model takes the average of the reciprocal of all correct examples. In this regard, MRR is given by as shown in Equations (9):

$$MRR = \frac{1}{|S|}\sum_{i=1}^{|S|} \frac{1}{rank_i} \tag{9}$$

The symbols involved in the above equation are the same as those in the equation for calculating MR. Given that the ranking is reciprocal, a larger value of this index indicates a superior evaluation effect.

In addition, hits@n represents the average proportion of triples ranked below the threshold n in link prediction, which is given by as shown in Equations (10):

$$hits@n = \frac{1}{|S|}\sum_{i=1}^{|S|} \mathbb{I}(rank_i \le n) \tag{10}$$

Typically, n is taken as 1, 3, or 10. In the above equation, $\mathbb{I}$ denotes the index function. The function value is 1 if the condition is true; otherwise, it is 0. A larger value of this index indicates a superior evaluation effect.

4.2 Datasets

In terms of general link prediction tasks, most researchers select FB15K-237 and WN18rr datasets, which are subsets extracted from the knowledge graphs Freebase and WordNet respectively, removing the redundancy relations of the original version. Their triple entities are all identified by the entity ID, which has no practical significance. Moreover, the implementation of KGE tasks usually does not need to determine the specific data meaning. NELL is a system that constantly extracts facts from the network. Likewise, NELL-995 is a subset of the dataset constructed by NELL's high confidence facts, which is characterized by relatively sparse relations and is helpful to validate the prediction ability of the model for long-tail entities.

This research employed the subsets generated by FB15K-237, WN18rr, and NELL-995 realized by Teru et al. to implement further experiments. Each subset consisted of the Train-Graph and the Ind-Test-Graph. The former was used for training, while the latter provided an incomplete graph for relation prediction. Of note, the Train-Graph encompasses all relations present in the Ind-Test-Graph, with non-overlapping entity sets. Detailed information regarding the datasets is illustrated in Table 1.

Table 1. Statistical Information of Datasets

Datasets	Subsets	Quantity of relations	Quantity of nodes	Quantity of links
FB15k-237	Train	180	1594	5223
	Train-1000	180	923	1027
	Train-2000	180	1280	2008
NELL-995	Train	88	2564	10063
	Train-1000	88	1086	1020
	Train-2000	88	2173	2011
WN18RR	Train	9	2564	6670
	Train-1000	9	893	1001
	Train-2000	9	1970	2002

4.3 Link Prediction

Table 2 presents the experimental results of link prediction of the model under FB15k-237 and WN18RR datasets. As can be seen from Table 2, the overall level of NT model indices on the FB15k-237 dataset is equivalent to HittER, the most advanced KGE model at present. The foregoing two models, based on the transformer architecture, both utilized neighborhood information. The NT model exhibited poor performance on the WN18RR dataset, which was slightly inferior to HittER and ATTH. The preliminary analysis of this research indicates that this phenomenon can be attributed to the more hierarchical structure of the WN18RR dataset, while HittER benefits from its hierarchical structure to capture such patterns. Furthermore, ATTH's score function is based on hyperbolic space, which possesses natural advantages in hierarchical data mining.

Table 2. Experimental Results of FB15k-237 and WN18rr

Models	**FB15k237**				**WN18rr**			
	MR	**MRR**	**hits@ 1**	**hits@ 10**	**MR**	**MRR**	**hits@ 1**	**hits@ 10**
TransE	357	0.294	-	0.465	3384	0.226	-	0.501
RotatE	**177**	0.338	0.241	0.533	**3340**	0.476	0.428	0.571
ComplEx	339	0.247	0.158	0.428	5261	0.440	0.410	0.510
CompGCN	244	0.325	0.237	0.501	4187	0.430	0.400	0.520
ATTH	187	0.344	0.209	0.529	4655	0.471	0.383	0.545
HIttER	184	0.349	0.279	0.560	4830	0.465	0.415	0.568

KG_BERT	164	**0.**261	0.261	0.539	3434	0.437	0.455	0.580
NTco−distillation	188	0.278	0.278	0.550	3809	0.490	0.450	0.581
PBco−distillation	176	**0.285**	**0.285**	**0.571**	3881	**0.500**	0.475	**0.586**

PB model demonstrated excellent performance in this experiment. In comparison with the KG-BERT model, which is also a pre-training model, the indices of the PB model were improved by 32% (MRR), 28% (hits@1), and 37%(hits@10), respectively. Given the limited additional prompt messages provided by the KG-BERT model, it can be considered that the additional information is of great significance in improving the effect of the pre-training model. Furthermore, the experimental results in Table 2 reveal that the PB model possesses obvious advantages over other typical KGE models. This may be attributed to the prompting function of the relational path and the rich knowledge contained in the pre-training model. Moreover, the introduction of con-distillation enables the PB model to learn additional domain information from NT.

This experiment investigated the training efficiency of several models. Given the difference in loss function selected by each model, this experiment mainly selected hit@1 as the reference index. Figure 6 depicts the change trend of hit@1 values of several models with iteration times under the same environment. As can be seen intuitively from Figure 6, several KGE models are superior to the pre-training model in terms of training duration and convergence speed. Particularly, the training speed of the proposed model significantly outperforms other pre-training models.

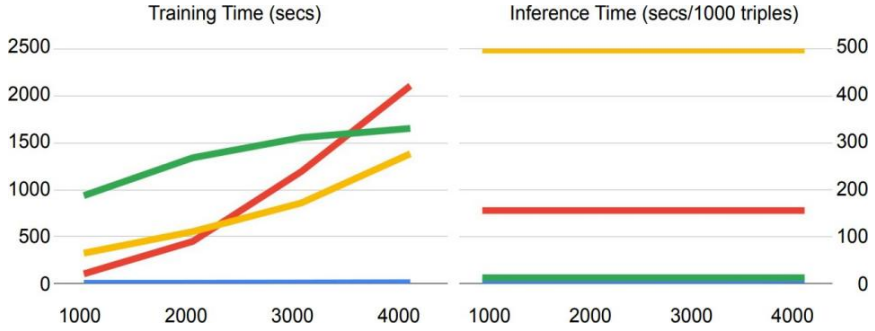

Fig. 6. Comparison of Training Time of Different Models

This research further analyzes the correlation between the prediction results of NT and PB models. By extracting the correctly predicted entities of the two models from the FB15K-237 dataset, this research investigated the coincidence degree of the two models within the prediction results, thereby exploring the necessity of conducting knowledge distillation between the two models. The specific experimental results are depicted in Figure 7.

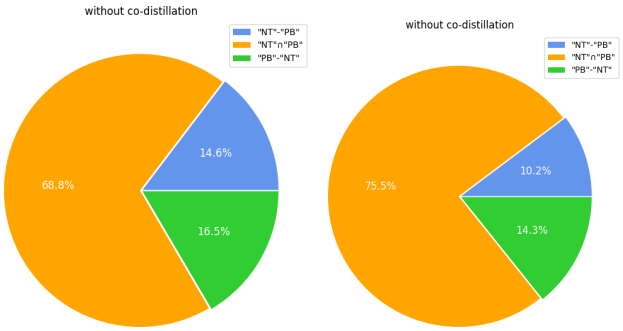

Fig. 7. Analysis and Statistics of the Coincidence Degree of Prediction Results of NT and PB Models

This experiment determined the coincidence degree regarding the hits@10 value of each entity of the two models under the conditions of canceling and adding the con-distillation module. More specifically, the NT model correctly predicted 21,438 entities, whereas the PB model correctly predicted 21,929 entities without employing the con-distillation module. Among them, as illustrated in the orange area in Figure 7, the total number of overlapping entities is 17,682, indicating that the two models learn features that the other may not learn. It implies that the two models are complementary to each other to some extent.

In the case of introducing the con-distillation module, on the other hand, the NT model correctly predicted 22,709 entities, while the PB model correctly predicted 23,071 entities, with a total of 20,038 overlapping entities. Furthermore, the hits@10 values of the two models were significantly improved, with the proportion of overlapping parts increasing from 68.8% to 75.5%. This demonstrates that the introduction of a con-distillation module enables an effective knowledge transfer between the two models.

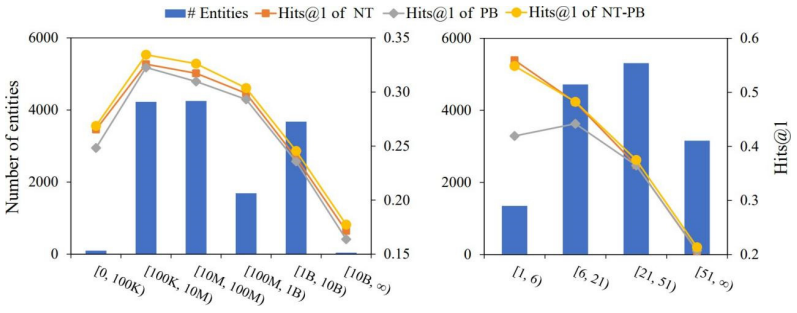

Fig. 8. Group-based Comparison of the Values of Hits@1 of Diverse Models

In addition, Figure 8 presents the analysis of the results of this research on long-tail and popular entities. This research divides the entities of the FB15K-237 dataset into several groups as per the number of edges of each entity. Specifically, the effect of the PB model far exceeds that of the NT model regarding the prediction of long-tail entities.

By contrast, the effect of the NT model is slightly superior to that of the PB model regarding the prediction of popular entities. Despite the close scores of the two models on the prediction of popular entities, the large number of triples related to popular entities leads to the complementarity that cannot be ignored. Moreover, numerous triples can solely be correctly predicted by the NT model. Simply put, the foregoing analysis validates the complementarity of NT and PB models regarding the prediction of popular and long-tail entities.

5 CONCLUSIONS

Regarding neighborhood-information-based knowledge graph completion, this research proposes an NT model for reconstructing and predicting missing entities based on relational neighbors of incomplete triples. The proposed model takes advantage of the transformer's self-attention mechanism to realize the simultaneous training of the prediction of missing entities or relations as well as neighborhood entity information. Meanwhile, an expert network with sparse gate control is introduced to reduce the parameters of the proposed model. Moreover, regarding the KG embedding based on relational paths, this research employs entity names and relational paths to detect entities, thereby generating the representation of missing entities by using BERT's robust pretraining background.

REFERENCES

1. Xiong W, Yu M, Chang S, One-Shot Relational Learning for Knowledge Graphs Conference on Empirical Methods in Natural Language Processing. 2018.
2. Zhang C, Yao H, Huang C, Few-Shot Knowledge Graph Completion//AAAI Conference on Artificial Intelligence. 2019.
3. Sheng J, Guo S, Chen Z, Adaptive Attentional Network for Few-Shot Knowledge Graph Completion //Conference on Empirical Methods in Natural Language Processing. 2020.
4. Baek J, Lee D B, Hwang S J. Learning to Extrapolate Knowledge: Transductive Few-shot Out-ofGraph Link Prediction. ArXiv, 2020, abs/2006.06648.
5. Wang S, Huang X, Chen C, REFORM: Error-Aware Few-Shot Knowledge Graph Completion. Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021.
6. Yao L, Mao C, Luo Y. KG-BERT: BERT for Knowledge Graph Completion. ArXiv, 2019, abs/1909.03193.
7. Zha H, Chen Z, Yan X. Inductive Relation Prediction by BERT//AAAI Conference on Artificial Intelligence. 2021.
8. Rokach L. Pattern Classification Using Ensemble Methods//Series in Machine Perception and Artificial Intelligence. 2009.
9. Araci D. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. ArXiv, 2019, abs/1908.10063.
10. Brandes N, Ofer D, Peleg Y, ProteinBERT: a universal deep-learning model of protein sequence and function. Bioinformatics, 2021, 38: 2102 -2110.