HIGH TECHNOLOGY LETTERS | Vol. 29 No. 3 | Sep. 2023 | pp. 269-278

doi:10.3772/j.issn.1006-6748.2023.03.005

Semantic-aware graph convolution network on multi-hop paths for link prediction①

PENG Fei (彭 斐), CHEN Shudong②, QI Donglin, YU Yong, TONG Da

(Institute of Microelectronics of the Chinese Academy of Sciences, Beijing 100029, P. R. China)
(University of Chinese Academy of Sciences, Beijing 101408, P. R. China)

Abstract

Knowledge graph (KG) link prediction aims to address the problem of missing multiple valid triples in KGs. Existing approaches either struggle to efficiently model the message passing process of multi-hop paths or lack transparency of model prediction principles. In this paper, a new graph convolutional network path semantic-aware graph convolution network (PSGCN) is proposed to achieve modeling the semantic information of multi-hop paths. PSGCN first uses a random walk strategy to obtain all k-hop paths in KGs, then captures the semantics of the paths by Word2Sec and long short-term memory (LSTM) models, and finally converts them into a potential representation for the graph convolution network (GCN) messaging process. PSGCN combines path-based inference methods and graph neural networks to achieve better interpretability and scalability. In addition, to ensure the robustness of the model, the value of the path threshold K is experimented on the FB15K-237 and WN18RR datasets, and the final results prove the effectiveness of the model.

Key words: knowledge graph (KG), link prediction, graph convolution network (GCN), knowledge graph completion (KGC), multi-hop paths, semantic information

0 Introduction

Nowadays knowledge graphs (KGs) are widely used in social media, e-commerce, healthcare and finance[1], such as YAGO[2], DBpedia, Freebase[3], and the analysis of knowledge graphs has become a hot topic of current research. KGs often represent facts by different triples, where each triple is of the form (head entity, relation, tail entity), or a simple form (h, r, t). However, due to the overabundance of factual information in reality, the triples in KGs are usually incomplete[4-5], which brings a great challenge to the study of knowledge graphs, leading to the study of link prediction for knowledge graphs. Link prediction algorithms can automatically complement KGs, which is very important to improve the quality of KGs.

In recent years, graph neural networks (GNNs) have received a lot of attention with the increasing research on KGs. For example, graph convolutional neural networks (GCNs)[6] is the most classical graph neural network model, and the main idea of GCNs is to map nodes into a low-dimensional representation and then update node information accordingly by aggregating the neighborhood information used for message passing, and thus gradually iterate to obtain the final result[7]. However, although GNNs make full use of the structural information of the knowledge graph, they ignore the relational paths between nodes, and thus the inference results of the model lack transparency, i.e., GNNs cannot provide explicit relational paths for model behavior interpretation. In contrast, the path-based inference approach extracts relational paths directly from KGs and models them interpretably. However, since the number of paths in the graph may increase incrementally with the length of the path, these models are difficult to achieve scalability, i.e., it is very difficult for the models to obtain good computational results when the number of paths is too large.

In this paper, a path semantic-aware graph convolutional network (PSGCN) is proposed for knowledge graph link prediction using semantic information of relational paths. Specifically, the corresponding k-hop path sets are obtained using each node in the graph as the head node, then the set of word vectors of the path sets are obtained using the Word2Sec model[8-10], and

① Supported by the National Natural Science Foundation of China (No. 61876144).
② To whom correspondence should be addressed. E-mail: chenshudong@ime.ac.cn.
Received on Nov. 16, 2022

the semantic information of the k-hop paths are obtained using the long short-term memory (LSTM) model[11-13], and finally the semantic information is used as the weights of the paths for the message passing of the model. In addition, to further enhance the robustness of the model, this work considers the effect of the value of parameter k (i. e., path length) of the k-hop path on the effect of the model, and finally chooses the optimal k value. Thus, the proposed model PSGCN inherits the scalability of GNN by preserving the message passing formula, while increasing the transparency of the mode's inference results by aggregating the semantic information of multiple k-hop paths of each PSGCN combines the advantages of both approaches node, i. e., path-based model and GNN model, making the model simultaneously interpretable and scalable, and solving the current problems of knowledge graph link prediction.

The main contributions of this work are as follows.

(1) The node and relationship contents in KGs are encoded as semantic vectors to facilitate better access to the semantic information in the KGs.

(2) The multi-hop path information of each node is added to the aggregated content of the graph neural network, which enriches the corresponding network structure and makes the model have some interpretability based on the inheritance scalability.

(3) The experimental results on two datasets show that PSGCN combines the advantages of both approaches and outperforms the experimental results compared with GNN-based inference models and path-based inference methods.

1 Related work

There are two common methods for link prediction, and the first one is to model the relational paths in the graph directly, i. e., to consider the implicit information of the relational paths based on the embedding model. Among them, path ranking algorithm (PRA)[14-15] is a typical algorithm that considers a path as 1 if it exists between h and t and 0 otherwise. Since the judgment logic of PRA[14-15] considers only the information of each path itself, the PSTransE[16], PTransE[17], RTransE[18] models are enriched with additional judgment conditions, which consider that the more information in the head entity that is passed through the path to the tail entity, the more reliable the path is. However, the semantic information of the paths is also an equally important judgment factor, for example, as shown in Fig. 1, Harry Potter novels→created→J. K. Rowling→career→British author and Harry Potter novels→created→British author are two different paths, but they convey close semantic information.

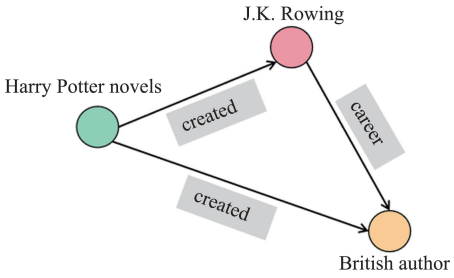

Fig. 1 An example of two different paths that convey the same semantics.

Therefore, it is important to add multi-hop path semantic information into the judgment condition of confidence of that path, which is not considered in the above models. Moreover, path-based models are difficult to achieve scalability due to the limitation of path length, so some models resort to using only one-hop paths (i. e., triples) to balance scalability and inference power.

The second approach is graph neural networks (GNNs), of which the most commonly used variant is graph convolutional networks (GCNs)[6]. Compared with the first approach, GCNs possess transferability. While GCNs perform message passing by aggregating the neighborhood information of each node, they ignore the relation type. Relational graph convolutional networks (RGCNs)[6] add the element of relation type on GCNs by performing relational aggregation, making them applicable to multi-relational graphs. Later, the graph attention network (GAT) model emerged to combine GNN with attention mechanism, while CapsNet and graph attention network (CGAT), describing relationships graph attention network (DR-GAT) and other variants improve the performance on the basis of GAT. However, neither GCNs nor GATs can provide explicit relational paths for model behavior interpretation, and thus the results are opaque and not interpretable.

Considering the advantages and disadvantages of the above two approaches, this paper chooses to use the semantic information of multi-hop paths as the aggregation element of the GNN model, which makes the model interpretable on the basis of transferability.

2 Problem definition and framework

2.1 Problem definition

In this paper, some common notations are defined to represent the structure of graphs.

The graph G is used to represent a KG, which consists of many triples, i. e.,

$$G = \{(e_1, r_1, e_2), \cdots, (e_i, r_k, e_j), \cdots, (e_{N-1}, r_M, e_N) \mid e_i, e_j \in E, r_k \in R, 1 \leqslant i, j \leqslant N, 1 \leqslant k \leqslant M\}$$
$$E = \{e_1, e_2, \cdots, e_N\} \quad R = \{r_1, r_2, \cdots, r_M\} \quad (1)$$

where E denotes the set of nodes on G, N is the number of nodes and R denotes the set of relations on G, M is the number of relation types.

2.2 Overall framework

The framework has three main components: finding multi-hop paths in the graph using a random walk strategy[19], a combination of Word2Vec[8-10] and LSTM[11-13] to obtain semantic information about the paths, and a graph convolution neural network to perform message passing on multi-hop paths. Fig. 2 shows a brief workflow. And the top part of the figure is for word embedding, which uses Word2Vec to get the word vectors of entities and relations. The middle part is the part for path semantic acquisition, which first uses a random wandering algorithm to acquire k-hop paths in the graph and then uses LSTM networks to acquire the path semantic information. The bottom part is a graph convolution network for learning the hidden information in the paths to facilitate link prediction.

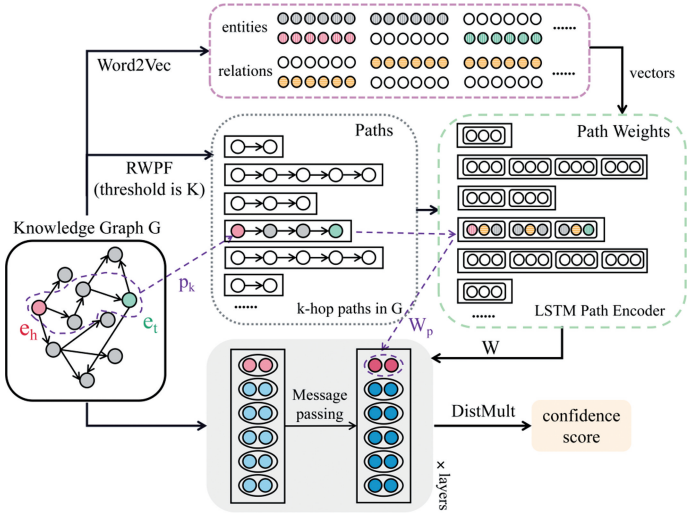

Fig. 2 The brief architecture of the graph convolution network using semantic information of multi-hop paths

Path set The set Paths consisting of all k-hop paths in graph G that satisfies $k \leqslant K$ is obtained using a random wandering strategy.

Path weights Based on the word vectors obtained by Word2Vec, the multi-hop paths in Paths are encoded using LSTM to obtain the corresponding semantic information as the weights of the paths.

Message passing on paths The path weights are passed as potential messages in the graph convolution neural network.

3 Proposed method

3.1 Path finding module

Separate triples are kept in the knowledge graph. To obtain different relational paths, this paper makes use of the random walk path finding (RWPF) strategy, which finds multi-hop paths centred on a single head node by randomly wandering over the knowledge graph. For example, because the end node of the triple Harry Potter novels → created → J. K. Rowling and the head node of the triple J. K. Rowling → career → British authors are the same J. K. Rowling, then they can be combined into one relational path, Harry Potter novels → created → J. K. Rowling → career → British authors.

Algorithm 1 shows the discovery process of multi-hop relational paths. First, a node e_1 is randomly selected as the current node from the graph, and the adjacent nodes can be reached by visiting each outgoing edge. A neighboring node e_2 is randomly selected as the new current node, and then each outgoing edge of the node is visited to get a random wandering path P. As long as the number of nodes in P is less than or equal to the path length K, the valid path is saved and the wandering continues. The above process is repeated until all valid paths are found. Note that visiting each neighbor node is random with the same probability.

Algorithm 1 Random walk path finding (RWPF)

Input: the number of nodes in KG N and the threshold walk length K;

Output: Paths (a set of effective paths);

Definition: define p as a cell path;
Initialization: initialize Paths to null and p to any triple in KG;
for $i = 0$ to N do
  $head = Node(i)$;
  $\boldsymbol{E} = Neighbors(head)$;
  for each e in **E** ***do***
    $p = (head, r, e)$;
    $\boldsymbol{T}$ = tails(Paths);
    if The length of p is less than or equal to K **then**
      if $\boldsymbol{T}$ contains the node e **then**
        Add p *to* Paths after the corresponding e;
      else
        Add p directly to Paths;
      end if
    end if
  end for
end for
return Paths;

In Algorithm 1, each neighbor node is selected randomly with the same probability, in which the probability of selecting a node and the corresponding current path are as

$$P_{e_1 \to e_2} = \begin{cases} 1 & \text{if the length of the current path is } = 0 \\ \dfrac{1}{|E|} & \text{if } 0 < \text{the length of the current path} \leqslant K \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$p = \begin{cases} s & \text{if the length of the current path is } = 0 \\ p + s & \text{if } 0 < \text{the length of the current path} \leqslant K \\ \text{null} & \text{otherwise} \end{cases} \tag{3}$$

where K is the threshold set for the path length, $|E|$ refers to the total number of neighboring nodes, and $p + s$ means to continue the algorithm along current path.

3.2 Path scoring module

In practical scenarios, it is possible that different triples have different semantics, and these differences may have an impact on the direction of the multi-hop path and gradually affect the final result. For example, if semantics are not taken into account, the triples (Harry Potter novels, created, J. K. Rowling) and (Harry Potter films, directed, Alfonso Cuaron) are only slightly different and would be represented as identical embeddings. But if their semantics are considered, the pathway of the triple (Harry Potter novels, created, J. K. Rowling) could be (J. K. Rowling, husband, Neil Murray), while the pathway of the triple (Harry Potter films, directed, Alfonso Cuaron) could be (Alfonso Cuaron, wife, Anna Lisa), and this differences can gradually accumulate and eventually have a huge impact on the results. As shown in Fig. 3, the triples in the dotted box differ little in terms of structure, but once the field of view is widened to look at the nodes after the longer path, it becomes clear that the two triples are very different.

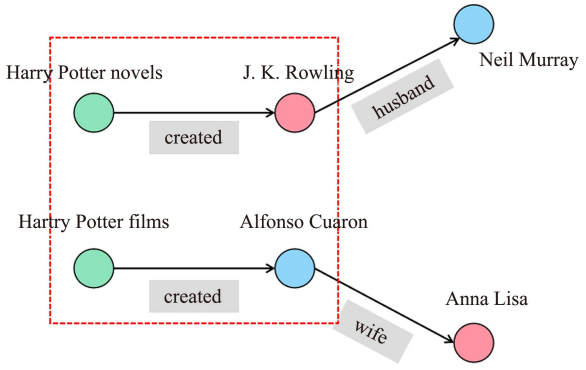

Fig. 3 An example of a path that influences the node characteristics.

It is therefore important to explicitly incorporate the semantics of entities and relations into path representation learning.

Embedding part The contents of entities and relations in the triples are often independent and there are no stopwords. Therefore, the Word2Vec[8-10] model is used for word embedding in order to obtain semantic information about the content of each entity and relation, and obtain an m-dimensional preprocessed word vector, which is convenient for later concatenating word vectors according to the multi-hop path. Algorithm 2 shows the process of connecting the semantics of the paths.

Algorithm 2 Semantic connection of paths

Input: Multi-hop path set $\boldsymbol{P}$, the set of entity word vectors $\boldsymbol{We}$ and the set of relational word vectors $\boldsymbol{Re}$;

Output: Multi-hop path feature set $\boldsymbol{F}$;

Definition: define p as a cell path, s as any triple in the path;
Initialization: initialize festure set $\boldsymbol{F}$ to null;
for each p in $\boldsymbol{P}$ **do**
  for each s in $\boldsymbol{E}$ **do**
    // h is the head node of s, often $s[0]$;

```
        //t is the tail node of s, often s[2];
        //r is the relation of s, often s[1];
        //Wp is the relation of s, often s[1];
        Add Wp to the feature set F;
    end for
end for
return the feature set F;
```

LSTM part RNN models can describe a path through a sequence and generate a vector of them to represent its overall semantics. Among various RNN methods, LSTM is used because LSTM is able to remember long-term dependencies in a sequence, and this memory and understanding of the sequential information of a sequence is essential for computing the confidence of a multi-hop relational path.

At step l of the current path, the LSTM hidden layer outputs a vector h_{l-1} and the current embedding of the entity e_l and the relation r_l are connected as the input vector.

$$x_l = e_l \oplus r_l \tag{4}$$

where $\oplus$ is the connection operation. Note that if e_l is the last entity, then an empty relation r_l is filled at the end of the path. Therefore, in order to make the input vector contain not only the sequential information of the nodes in the path, but also the semantic information of the entity and its relation to the next entity, h_{l+1} and x_l is used to learn the state of the l-th hidden layer of the LSTM as

$$f_l = \sigma(\boldsymbol{W}_f \cdot concat(h_{l-1}, x_l) + b_f) = \sigma(\boldsymbol{W}_f x_l + \boldsymbol{W}_h h_{l-1} + b_f) \tag{5}$$

$$i_c = \sigma(\boldsymbol{W}_i \cdot concat(h_{l-1}, x_l) + b_i) = \sigma(\boldsymbol{W}_i x_l + \boldsymbol{W}_h h_{l-1} + b_i) \tag{6}$$

$$d_l = \tanh(\boldsymbol{W}_c \cdot concat(h_{l-1}, x_l) + b_c) = \tanh(\boldsymbol{W}_c x_l + \boldsymbol{W}_c h_{l-1} + b_c) \tag{7}$$

$$c_l = f_l c_{l-1} + i_l d_l \tag{8}$$

$$o_l = \sigma(\boldsymbol{W}_o \cdot concat(h_{l-1}, x_l) + b_o) = \sigma(\boldsymbol{W}_o x_l + W_h h_{l-1} + b_o) \tag{9}$$

$$h_l = o_l \cdot \tanh(c_l) \tag{10}$$

where, i_l, o_l, f_l denote input gates, output gates and forgetting gates, respectively; $c_l \in R^d$, $d_l \in R^d$ denote the cell state vector and information conversion module, respectively, and d is the number of hidden cells; $\boldsymbol{W}_c, \boldsymbol{W}_i, \boldsymbol{W}_f, \boldsymbol{W}_o$ and $\boldsymbol{W}_h$ are mapping coefficient matrices, and b_c, b_i, b_f and b_o are bias vector; $\sigma(\cdot)$ is the activation function.

In order to score the different paths, i. e., to get the semantic confidence of the multi-hop path p_k as a whole, therefore the final output is used into the score using a fully-connected layer, given by

$$s(path) = \mathrm{ReLU}(\boldsymbol{W}_{\mathbf{fc}} p_k + b) \tag{11}$$

where, $\boldsymbol{W}_{\mathbf{fc}}$ is the fully connected layer coefficient weights and the ReLU function is used as the activation function.

3.3 Messaging module

In order to equip GNN with the ability to capture multi-hop path information, the semantic information of k-hop paths is modeled directly in the message passing process of GNN. And the set of valid k-hop relational paths is defined as

$$q_k = \{(e_h, r_1, \cdots, r_k, e_t) \mid (e_h, r_1, e_1), \cdots, (e_{k-1}, r_k, e_t) \in \boldsymbol{G}\} \quad (1 \leqslant k \leqslant K) \tag{12}$$

where e_h is the head node, e_t is the tail node, $r_1, \cdots, r_k$ are all relations of the path q_k, $e_1, \cdots, e_{k-1}$ are the other nodes of the path q_k, and G denotes the set of all triples in KG.

The messaging along the k-hop ($1 \leqslant k \leqslant K$) path is as

$$z_{e_h}^k = \sum_{r \in R} \sum_{(e_h, r_1, \cdots, r_k, e_t) \in q_k} \frac{1}{C_{e_{i,r}}} \frac{1}{k} \boldsymbol{W}_{e_h} \boldsymbol{W}_{r_1} \cdots \boldsymbol{W}_{r_k} x_{e_t} \quad (1 \leqslant k \leqslant K) \tag{13}$$

where $C_{e_{i,r}}$ refers to the total number of relationship types in all paths with e_h as the head node. And $\boldsymbol{W}_e = LSTM(e)$, $e \in \{e_h, \cdots, e_t\}$, $\boldsymbol{W}_r = LSTM(r)$, $r \in \{r_1, \cdots, r_k\}$.

Information from paths of different lengths is aggregated via attention mechanism as

$$z_{e_h} = \sum_{k=1}^{K} \mathrm{softmax}(a^{\mathrm{T}} z_{e_h}^k) \tag{14}$$

where a^{T} is a trainable parameter for calculating the attention coefficient.

And finally the embedding of the output node by nonlinear activation is as

$$h_{e_i}' = \sigma(\boldsymbol{W}_{\mathbf{self}} h_{e_i} + \boldsymbol{W}_z z_{e_h}) \tag{15}$$

where $\boldsymbol{W}_{\mathbf{self}}$ and $\boldsymbol{W}_z$ are learnable model parameters, and σ is a non-linear activation function.

The purpose is to predict new triples, i. e., to calculate the scores $f_{h,r,t}$ of possible triples (containing possible edges) (h, r, t), and thus determine the confidence of these edges. Firstly the negative sampling is sampling ω negative samples for each positive sample by randomly replacing the head and tail nodes h and t with other entities. And then the plausibility scores of the positive samples are maximized by minimizing the cross-entropy loss as follows.

$$L = -\frac{1}{\hat{\omega} \mid \boldsymbol{T} \mid} \sum_{(h,r,t) \in T} y \log \rho(f_{h,r,t}) + (1 - y) \log(1 - \rho(f_{h,r,t})) \tag{16}$$

where $\hat{\omega}$ is $\omega + 1$, ρ is the sigmoid function, and y is an indicator set to $y = 1$ for positive triples and $y = 0$ for negative ones. And DistMult[20] is used as the scoring function, i. e. , all other k-hop relational paths ($k > 1$) from h to t can be associated as a matrix $\boldsymbol{P}_r$, so $f_{h,r,t}$ is denoted as: $f_{h,r,t} = e_h^{\mathrm{T}} \boldsymbol{P}_r e_o$.

4 Experiments

4.1 Datasets

FB15K-237 is a subset of FB15K and FB15K is a subset of Freebase. WN18RR is a subset of WN18 and WN18 is a subset of WorldNet. Although both FB15K and WN18 have a large number of test triples, they can be obtained simply by inverting the training triples[21], while FB15K-237 and WN18RR avoid this problem, and therefore they are more convincing datasets. In this paper, these two more convincing datasets, FB15k-237 and WN18RR, are chosen to validate the proposed model. The details of the datasets are shown in Table 1, where PER denotes the average number of triples for each relationship type.

Table1 The statistics of different datasets

Dataset	Nodes	Edges	Train	Valid	Test	PER
FB15K-237	14541	237	272115	17535	20466	1309
WN18RR	40943	11	86836	3034	3134	8455

Few relations in the FB15K-237 dataset have hierarchical characteristics. Therefore, this dataset is sparse (as shown in the table, the number of entities corresponding to each class of relations is small), and the network hierarchical characteristics are not obvious. In contrast, the WN18RR dataset contains hierarchical relationships between words, such as hypernym, has_part. Therefore, the dataset is dense (as shown in the table, the number of entities corresponding to each type of relationship is larger), with a natural hierarchical structure, and the network hierarchical characteristics are obvious.

4.2 Evaluation metrics and parameter settings

Mean reciprocal rank (MRR) and Hitratio@K (Hits@K) are used to compare the performance of the model in experiments, where for Hits@K, Hits@1 and Hits@10 (the proportion of correct answers ranked in the top 1 and top 10 in all candidate sets) are chosen as the evaluation metrics. And all three metrics (MRR, Hits@1 and Hits@10) have higher values for better model performance.

To make the model work best, the same configuration for the following hyperparameters are used: initial embedding dimension of entities and relations $d = 100$, word vectors dimension $wd = 100$, dropout rate $\mu = 0.5$, learning rate $\gamma = 0.001$, the initial value of the threshold for path lengths $K = 2$, and the number of units of LSTM is set to 128 according to the path length of FB15K-237 and WN18RR datasets. In addition, the GNN depth l is set to 4 for the FB15K-237 dataset and 2 for the WN18RR dataset. The reason is that the FB15K-237 dataset has more data than the WN18RR dataset, so a deeper neural network is needed for training.

4.3 Baseline

In order to evaluate our method in terms of both knowledge transfer learning ability and multi-hop path information capture ability, several representative methods corresponding to them are selected for comparison.

Basic group (models without considering path information) including RGCN[6], DistMult[22], ComplEx[23], TransE[24], ConvKB[25], CGAT[26] and DR-GAT[27] are chosen.

Path group (adding path information into the embedding vector) including PSTransE[16], PTransE[17] and RTransE[18] are chosen.

4.4 Results and analysis

The results of the different models on FB15K-237 and WN18RR are given in Table 2. For the overall experimental results the comparison shows that the proposed method outperforms the other methods, including the basic and path groups.

Specifically, in the basic group, (1) for the FB15K-237 dataset, Hits@1 is improved by about 10.94%, Hits@10 is improved by about 15.15%, MRR is improved by about 9.18%; (2) for the WN18RR dataset, Hits@10 is improved by about 5.53%, MRR is improved by about 5.57%. This demonstrates that adding path information can improve the effect. And in the path group, (1) for the FB15K-237 dataset, Hits@1 is improved by about 78.45%, Hits@10 is improved by about 11.83%, MRR is improved by

about 42.32%; (2) for the WN18RR dataset, Hits@10 is improved by about 1.48%, MRR is improved by about 17.37%. This demonstrates the effectiveness of using GCN for learning about path information.

Table 2 The results of link prediction on FB15K-237 and WN18RR

	Methods	FB15K-237			WN18RR		
		Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
Basic	RGCN	0.140	0.380	0.194	0.180	0.487	0.193
	DistMult	0.198	0.419	0.281	0.39	0.504	0.430
	ComplEx	0.194	0.450	0.278	0.409	0.510	0.449
	TransE	0.199	0.471	0.291	0.123	0.532	0.366
	ConvKB	-	0.421	0.289	-	0.525	0.248
	CGAT	-	0.432	**0.295**	-	0.528	0.242
	DR-GAT	0.158	0.425	0.257	0.164	0.493	0.238
Path	PTransE(3-step)	0.164	0.443	0.268	0.156	0.560	0.368
	RTransE(3-step)	-	**0.487**	0.281	-	0.562	0.398
	PSTransE	-	0.476	0.279	-	0.571	0.390
	This work	**0.209**	0.483	0.294	**0.420**	**0.573**	**0.454**

According to Table 2, it is clear that PSGCN is less effective on the FB15K-237 dataset than the WN18RR, which is supposed to be due to the high number of paths in the FB15K-237 dataset, resulting in a large amount of information redundancy. Therefore, to confirm the above conjecture, experiments are conducted on the threshold value of the multi-hop path length (i.e. parameter K) and results are shown in Table 3 and Fig. 4.

Table 3 The numbers of k-hop paths on different values of K

	FB15K-237			WN18RR		
	Train	Valid	Test	Train	Valid	Test
$K=1$	272k	17k	20k	86k	3.03k	3.13k
$K=2$	233 30k	98k	131k	320k	3.32k	3.45k
$K=3$	825 546k	215k	316k	949k	3.34k	3.47k
$K=4$	46 229 366k	411k	700k	2761k	3.35k	3.47k
$K=5$	-	769k	1510k	8062k	3.35k	3.47k

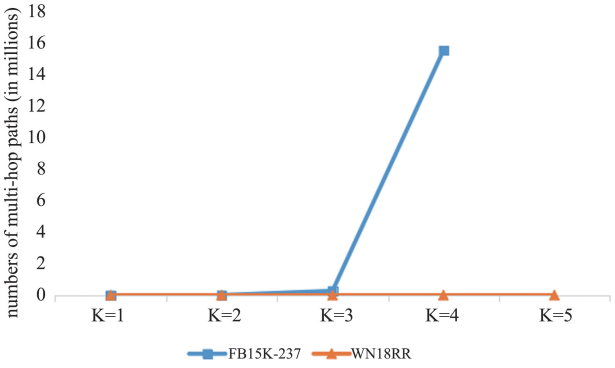

Fig. 4 Changes in the number of k-hop paths with the threshold value K

It is clear that the number of paths in FB15K-237 grows very rapidly as the path length threshold K increases, clearly far exceeding the rise in WN18RR. Specifically, a sharp increase in the number of paths occurs when $K \geqslant 3$. In order to investigate the effect of a larger number of paths on the effect of the model, experiments on the FB15K-237 and WN18RR datasets are conducted to obtain the variation of the effect of the model at different K values, and the results are shown in Tables 4 – 5 and Figs 5 – 6.

Table 4 Performance at different K values on the FB15K-237 dataset

	Hit@1	Hit@10	MRR
$K=1$	0.206	0.471	0.291
$K=2$	**0.241**	0.496	0.324
$K=3$	0.237	**0.506**	**0.338**
$K=4$	0.209	0.483	0.294

Table 5 Performance at different K values on the WN18RR dataset

	Hit@1	Hit@10	MRR
$K=1$	0.369	0.528	0.405
$K=2$	0.409	0.542	0.434
$K=3$	**0.423**	0.566	0.449
$K=4$	0.420	**0.573**	**0.454**
$K=5$	0.419	0.569	0.452

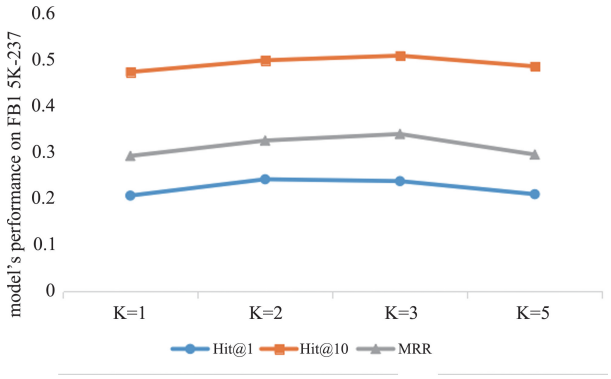

Fig. 5 Performance at different K on FB15K-237

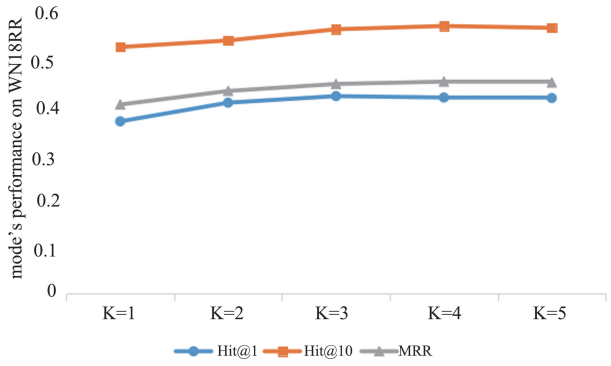

Fig. 6 Performance at different K on WN18RR

From Table 3 it can be seen that the number of paths in the test set in WN18RR remains essentially constant after $K \geqslant 3$. Accordingly, the impact of $K = 3$, $K = 4$ and $K = 5$ is basically stable for the WN18RR dataset, as shown in Fig. 6 and Table 5.

In addition, as shown in Fig. 5 and Fig. 6 there is an interesting phenomenon that the effect of the model on the FB15K-237 dataset decreases significantly at $K \geqslant 3$, while the effect of the model on WN18RR increases significantly at $K \leqslant 3$, which indicates that the model does not make effective use of the path information for the FB15K-237 dataset. Considering the sparsity of datasets, the experiments on FB15K-237 on the number of layers l of the neural network are conducted to study the effect of PSGCN's complexity on the inference performance of the model. And the experimental results are shown in Table 7.

When the threshold value K of path length is fixed as 1, 2, 3, respectively, the performance changes of the model when setting different neural network layers l are shown in Figs 7 – 9.

Table 7 Variation of model performance with parameters K and l

	Hits@1	Hits@10	MRR
$l=2, K=1$	0.187	0.398	0.223
$l=2, K=2$	**0.196**	0.403	0.258
$l=2, K=3$	0.194	**0.417**	0.257
$l=3, K=1$	0.192	0.442	0.258
$l=3, K=2$	**0.220**	0.471	**0.303**
$l=3, K=3$	0.202	**0.481**	0.295
$l=4, K=1$	0.206	0.471	0.291
$l=4, K=2$	**0.241**	0.496	0.324
$l=4, K=3$	0.237	**0.506**	**0.338**
$l=5, K=1$	0.210	0.476	0.294
$l=5, K=2$	**0.262**	0.511	0.336
$l=5, K=3$	0.248	**0.512**	**0.342**
$l=6, K=1$	0.213	0.479	0.294
$l=6, K=2$	**0.258**	**0.512**	0.331
$l=6, K=3$	0.244	0.509	**0.341**

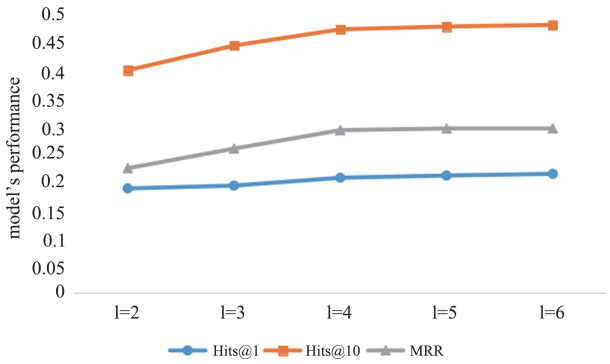

Fig. 7 The variation of model performance with parameter l for $K = 1$

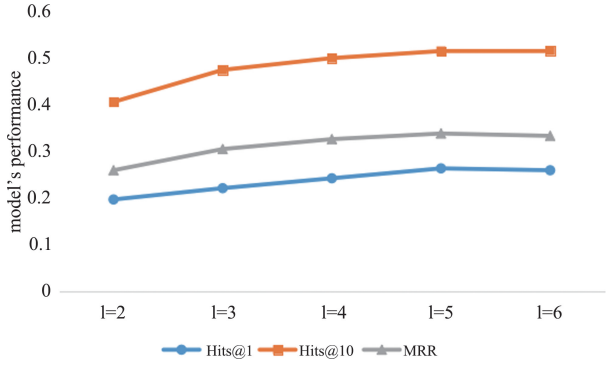

Fig. 8 The variation of model performance with parameter l for $K = 2$

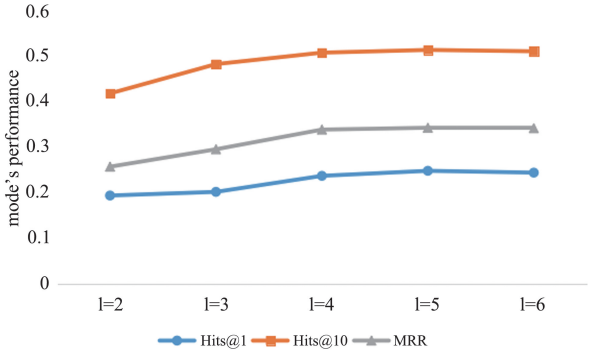

Fig. 9 The variation of model performance with parameter l for $K = 3$

According to Table 7 and Figs 7 – 9, it can be seen that the model has the best inference performance on the FB15K-237 dataset when the number of neural network layers l is taken as 5 or 6. Since the size of the FB15K-237 dataset is positively about 5 to 6 times larger than that of WN18RR, this phenomenon appears probably because the complexity of the model and the size of the dataset are closely and positively correlated.

Therefore, the effectiveness of the model to perform inference is related to the sparsity of the dataset. For datasets with more sparse data (e. g., FB15K-237), the number of paths in the dataset is higher and the amount of data that the model can process is larger, so the depth of the network structure in the model should be increased (the number of neural network layers l is set a bit higher) in order to effectively utilize these path data. Meanwhile, previous work on K-value illustrates that when the K-value is too high, the FB15K-237 dataset has a large amount of redundant information, which will lead the model to learn some useless data and eventually lead to poor performance of the model. Therefore, combining the two parameters l and K, the model should be set with higher l and lower K for more sparse and larger datasets, and conversely, for denser and relatively smaller datasets (e. g., WN18RR), the model should be set with lower l and higher K accordingly.

5 Conclusions

Most existing link prediction methods are divided into two types. One approach is based on graph neural networks for inference, and only focuses on the topological information of neighboring nodes in the graph data, while ignoring the knowledge paths and the semantic information of the paths in the graph data, which also leads to the models without interpretability; the other approach is to perform inference by path derivation only. Although such model can explain inference results by knowledge paths, their results are unstable when dealing with large datasets. To obtain an interpretable and efficient inference method, in this work, a neural network PSGCN capable of capturing semantic information of multi-hop paths is proposed to solve the link prediction problems, while achieving the interpretability of multi-hop path inference methods and the scalability of GNNs that are efficiently applicable to large datasets. In addition, experimental results show that the impact of the path length threshold parameter K and the number of neural network layers l on the effectiveness of the model is important, and the most appropriate parameter K value and l value are finally chosen in this work to further improve the results. In short, PSGCN combines the advantages of both GNNs and path-based inference models to efficiently perform multi-hop relational inference in an explicit and interpretable manner, obtaining better results on the FB15K-237 and WN18RR datasets.

Reference

[1] JI S X, PAN S R, CAMBRIA E, et al. A survey on knowledge graphs: representation, acquisition, and applications [J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(2): 494-514.

[2] SUCHANEK F M, KASNECI G, WEIKUM G. Yago: a core of semantic knowledge[C]//Proceedings of the 16th International Conference on World Wide Web. Banff: Association for Computing Machinery, 2007: 697-706.

[3] BOLLACKER K D, EVANS C, PARITOSH P K, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]//Proceedings of the ACM SIGMOD International Conference on Management of Data. Vancouver: Association for Computing Machinery, 2008: 1247-1250.

[4] LIN Q K, LIU J, PAN Y D, et al. Rule-enhanced iterative complementation for knowledge graph reasoning[J]. Information Sciences, 2021, 575: 66-79.

[5] SHI B X, WENINGER T. PROJ E: embedding projection for knowledge graph completion[C]//Proceedings of the 31st AAAI Conference on Artificial Intelligence. San Francisco: AAAI Press, 2017: 1236-1242.

[6] SCHLICHTKRULL M S, KIPF T N, BLOEM P, et al. Modeling relational data with graph convolutional networks[C]//The 15th International Conference on Semantic Web. Heraklion: Springer, 2018: 593-607.

[7] LIN B Y, CHEN X Y, CHEN J, et al. KagNet: knowledge-aware graph networks for commonsense reasoning[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong: EMNLP, 2019: 2829-2839.

[8] HA T, LEE M, YUN B, et al. Job forecasting based on the patent information: a word embedding-based approach[J]. IEEE Access, 2022, 10: 7223-7233.

[9] GIABELLI A, MALANDRI L, MERCORIO F, et al. WETA: automatic taxonomy alignment via word embeddings[J]. Computers in Industry, 2022, 138: 103626.

[10] HAISA G, ALTENBEK G. Deep learning with word embedding improves Kazakh named-entity recognition[J]. Information, 2022, 13(4): 180.

[11] WANG Z, YANG B, Attention-based bidirectional long short-term memory networks for relation classification using knowledge distillation from BERT[C]//IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress. Calgary: DASC/PiCom/CBDCom/CyberSciTech, 2020: 56-568.

[12] HUANG Z H, XU W, YU K. Bidirectional LSTM-CRF

models for sequence tagging [EB/OL]. (2015-08-09) [2023-02-10]. https://arxiv.org/pdf/1508.01991.pdf.
[13] ALAPARTHI S, MISHRA M. Bidirectional encoder representations from transformers (BERT): a sentiment analysis odyssey [EB/OL]. (2020-07-02) [2023-02-10]. https://arxiv.org/ftp/arxiv/papers/2007/2007.01127.
[14] LAO N, MITCHELL T M, COHEN W W. Random walk inference and learning in a large scale knowledge base [C]// Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Edinburgh: Association for Computational Linguistics, 2011: 529-539.
[15] LAO N, COHEN W W. Relational retrieval using a combination of path-constrained random walks [J]. Machine Learning, 2010, 81(1): 53-67.
[16] CHEN H X, ZHOU Q, LIU X J. A knowledge reasoning method combining path information and embedded model [J]. Minicomputer System, 2020, 41(6): 1147-1151.
[17] LIN Y K, LIU Z Y, SUN M S. Modeling relation paths for representation learning of knowledge bases [EB/OL]. (2015-08-15) [2023-02-10]. https://arxiv.org/pdf/1506.00379.pdf: arXiv.
[18] GARCÍA-DURÁN A, BORDES A, USUNIER N. Composing relationships with translations [C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: Association for Computational Linguistics, 2015: 286-290.
[19] NIKOLENTZOS G, VAZIRGIANNIS M. Random walk graph neural networks [C]// Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems. Vancouver: NEURIPS, 2020: 1-12.
[20] DONG X, GABRILOVICH E, HEITZ G, et al. Knowledge vault: a web-scale approach to probabilistic knowledge fusion [C]// The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2014: 601-610.
[21] FENG Y L, CHEN X Y, LIN Y C, et al. Scalable multi-hop relational reasoning for knowledge-aware question answering [C]// Proceedings of the 2020 Conference on Empirical Methods in Natural Language. Online: EMNLP, 2020: 1295-1309.
[22] YANG B S, YIH W T, HE X D, et al. Embedding entities and relations for learning and inference in knowledge bases [J]. [EB/OL]. (2015-08-29) [2023-02-10]. https://arxiv.org/pdf/1412.6575.pdf: arXiv.
[23] TROUILLON T, WELBL J, RIEDEL S, et al. Complex embeddings for simple link prediction [C]// Proceedings of the 33nd International Conference on Machine Learning. New York: JMLR, 2016: 2071-2080.
[24] BORDES A, USUNIER N, WESTON J, et al. Translating embeddings for modeling multi-relational data [C]// Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Lake Tahoe: Association for Computing Machinery, 2013: 2787-2795.
[25] NGUYEN D Q, NGUYEN T D, NGUYEN D Q, et al. A novel embedding model for knowledge base completion based on convolutional neural network [C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans: Association for Computational Linguistics, 2018: 327-333.
[26] WANG K L, ZHOU Z L, CHEN D H, et al. Research on link prediction combining GAT and CapsNet [J]. Communications Technology, 2022, 55(2): 143-150.
[27] ZHENG X B, CUI Y, ZHAO X L. Research on link prediction based on relation graph convolution neural network [J]. Modern Computer, 2021, 18: 50-55.

PENG Fei, born in 1998. She is pursuing her M. S. degree at the University of Chinese Academy of Sciences. She received her B. S. degree from Henan University in 2020. Her research interests include knowledge graphs, artificial intelligence, and natural language processing.