Entity Representation by Neighboring Relations Topology for Inductive Relation Prediction

Zhigui Chen, Hang Yu$^{(\boxtimes)}$, Jinpeng Li, and Xiangfeng Luo

School of Computer Engineering and Science, Shanghai University,
Shanghai 200444, China
{chenzhigui,yuhang,lijinpeng,luoxf}@shu.edu.cn

Abstract. Inductive relation prediction is to predict relations between unseen entities. The current methods implicitly learn the logical rules in the knowledge graph through the local subgraph structures, and obtain the latent semantic representation of the predicted triples. However, existing methods lack relation information of neighboring triples due to the incompleteness of the knowledge graph, and the representation of entities does not consider the connection structures between relations which contain different semantic information. To address these challenges, we propose a novel entity representation by Neighboring Relations Topology Graph (NRTG) for inductive relation prediction. Specifically, we divide connection structures between relations into several *topological patterns*, and design a module to extract relations of all neighboring triples for constructing Neighboring Relations Topology Graph (NRTG). In NRTG, the nodes represent the relations and the edges represent the *topological patterns*. Afterward, we design an information aggregation module to encode the NRTG as the entity representation, and then use the scoring network to predict relations between unseen entities. Experiments demonstrate that our model can effectively capture relation information of neighboring triples and semantic information of connection structures between relations. Moreover, it outperforms existing methods on benchmark datasets for the inductive relation prediction task.

Keywords: Knowledge graph · Knowledge graph completion · Inductive relation prediction

1 Introduction

Nowadays, knowledge graphs play a very important role in natural language processing [27], recommendation systems [21] and question answering [8]. However, the existing knowledge graphs are incomplete, so the relation prediction task is required for reasoning and completion. The relation prediction on the knowledge graph is divided into transductive and inductive. Transductive relation prediction [1,3,16] learns and operates on latent representations (i.e., embeddings) of

S. Khanna et al. (Eds.): PRICAI 2022, LNCS 13630, pp. 59–72, 2022.
https://doi.org/10.1007/978-3-031-20865-2_5

entities and relations in a knowledge graph. However, this method can only make relation predictions for entities that appear in the training set, and cannot represent unseen entities. On the contrary, inductive relation prediction [6,17,19] is entity-independent, and this approach can make relation prediction for entities that are not present in the training set. For the real world, existing knowledge graphs cannot cover all entities, so the problem of relation prediction for unseen entities has been paid more and more attention by researchers.

Existing models of inductive relation prediction mainly predict missing relations by learning logical rules in knowledge graphs. At present, there are mainly two types of methods for learning logic rules. Rule-based learning explicitly mines logical rules based on co-occurrence patterns of relations. Inductive relation prediction by local subgraph structures, such as GraIL [17], implicitly learns the logical rules in subgraph based on Graph Neural Networks (GNN) [7,12,14]. More recently, TACT [2] classifies relation pairs in subgraphs into several patterns, and incorporates these messages into the representation of relations.

Although subgraph-based models have shown inductive learning capability in validating unseen nodes, there are some disadvantages. First, many inference paths are disconnected due to the incompleteness of the knowledge graphs. Therefore, the subgraph will miss a lot of neighboring relation information. Taking Fig. 1 as an example, for entity "GSW", neighboring relations "coach_of" and "belong_to" are not connected to the tail node so the representation of "GSW" lacks these relation information. Moreover, existing methods do not take into account connection structures between relations in entity representation. For example, in Fig. 1, the relation "part_of" has three connection structures such as "parallel", "tail-to-head" and "tail-to-tail" with predicted relation "located_in" and these connection structures have different effects on the representation of "GSW". In this way, the representations of nodes "GSW" and "Californla" with topology information are obtained, respectively. Then, combine the embedding of "located_in" into the scoring function to get the likelihood of this triple.

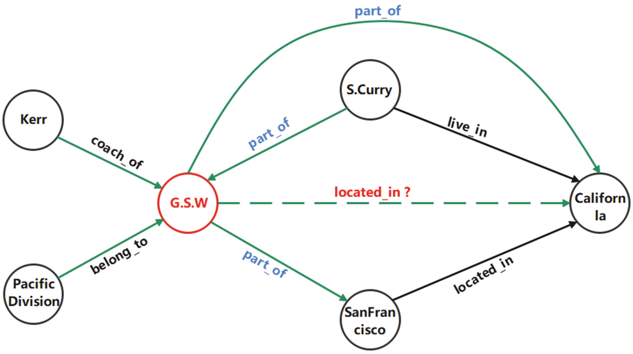

Fig. 1. An example in knowledge graphs.

To address these disadvantages, we propose a novel entity representation by Neighboring Relations Topology Graph (NRTG) for inductive relation prediction. Specifically, the NRTG extracts all neighboring triples and then divides the connection structures between relations into six topological patterns. Therefore, our method can capture relation information of neighboring triples and connection structures between relations by NRTG.

For predicted triples, our model consists of the following stages: (1) constructing NRTG via relations topology module. (2) getting head and tail entity representations of predicted triple via information aggregation module based on GNN [7]. (3) inputting head and tail entity representations and embedding the predicted relation into the scoring network to obtain the predicted triples score.

Our contributions are as follows. First of all, we propose a novel framework that uses two graph structures to represent the head and tail entities of predicted triples separately. This framework can more completely mine the logical information implied by the head and tail entities in the knowledge graph. Secondly, we design Neighboring Relations Topology Graph (NRTG) to capture the semantic information of connection structures among relations. Finally, it significantly outperforms existing inductive relation prediction methods on benchmark datasets.

The remainder of this article is structured as follows. Related works are introduced in Sect. 2. The specific details of our method are introduced in Sect. 3. The experiments used to analyze and verify the effectiveness of our method in Sect. 4. Section 5 concludes this article and proposes future works.

2 Related Work

At present, there are two main methods for relation prediction on knowledge graphs. One is the rule learning-based methods, and the other is the embedding-based reasoning methods:

Rule Learning-Based Methods. Rule-based methods [4] learn logical rules by relational co-occurrence patterns of knowledge graphs. Because these logic rules are independent of entities, these methods can predict relations between unseen entities. Despite the fact that these methods are inherently inductive, these methods are difficult to scale to large datasets. Recently, NeuralLP [23] proposed an end-to-end framework to address scalability issues. Based on NeuralLP, DRUM [13] can mine more correct logic rules. However, These logical rules cannot learn the complex topological structure between relations.

Embedding-Based Methods. Most of the existing methods are embedding-based methods such as TransE [1], ConvE [3], ComplEx [20] and RotatE [16], which is to learn a low-dimensional embedding vector for each entity and relation in a knowledge graph. In recent years, more and more researchers have applied graph neural networks (GNN) [7,12,14] to relation prediction, as knowledge graphs naturally have graph structures. Schlichtkrull et al. [15] propose a relational graph neural network that considers the connected relations to represent entities. Afterward, GAT [11] proposes a graph neural network based on an attention mechanism to teaches the representation of entities, which effectively learn the knowledge of neighboring triples. More recently, Zhang et al. [26] proposed a relational graph neural network with hierarchical attention to effectively utilize the neighborhood information of entities in knowledge graphs.

To predict the relation between unseen entities, GraIL [17] reasons via entity-independent local subgraph structures. On the basis of GraIL, TACT [2] considers semantic correlations between relations, and models correlation coefficients of the different semantic correlations into relation representation. Moreover, there are some neural networks [24,25] that learn topology.

However, these methods have limitations. The incompleteness of the graph can lead to insufficient learning of neighboring relations. Furthermore, these methods are too simplistic to model entity representations since these methods do not take into account the topological structure between neighboring relations and predicted relations.

3 Methods

In this section, we introduce our proposed model. The task of our model is inductive relation prediction, which predicts the relation between unseen entities. For inductive relation prediction, we need to represent entities that have not been seen in the training set. Therefore, our model uses two Neighboring Relations Topology Graphs (NRTGs), in which the nodes represent the relations and the edges represent the connection structures between relations, to represent the head and tail entity respectively. Then our model scores the predicted triple through head representation, tail entity representation and embedding of predicted relation. Our model consists of the following parts: (1) Relations topology module. (2) Information aggregation module based on GNN [7]. (3) Scoring network and Loss function. Figure 2 gives an overview of our model.

3.1 Relations Topology Module

In order to solve the problems that the existing model does not capture the complete neighboring relations and does not consider connection structures between relations, we design this module to fully mine the implicit logical rules of predicted triple in knowledge graphs in two aspects: neighboring relations extraction and Neighboring Relations Topology Graph (NRTG).

Neighboring Relations Extraction. For existing subgraph-based methods, they assume that the paths connecting the head and tail entity contain the logical information that could represent the predicted triple. Differing from the existing subgraph-based models, we assume that the relations of all neighboring triples imply the logical rules of relation prediction. Because the knowledge graph is incomplete, many reasoning paths are disconnected. Therefore, neighboring relations that do not exist on the reasoning path can also provide a basis for relation prediction. Furthermore, we extract two subgraphs from the knowledge graph to represent the head and tail entity of the predicted triple,

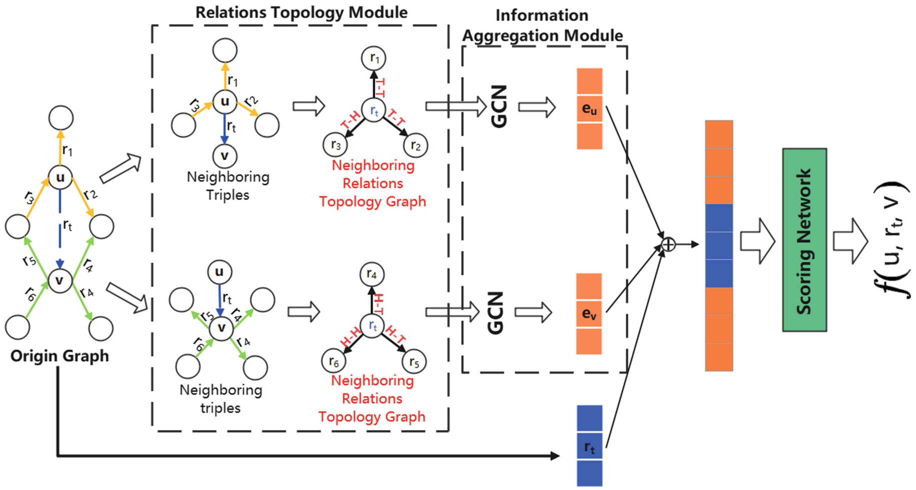
Relations Topology Module
Information Aggregation Module
Origin Graph
u
v
r1
r2
r3
r4
r5
r6
rt
Neighboring Triples
Neighboring triples
Neighboring Relations Topology Graph
Neighboring Relations Topology Graph
T-T
T-H
H-H
H-T
GCN
GCN
eu
ev
rt
Scoring Network
f(u, rt, v)

Fig. 2. An overview of our model. The framework consists of two modules. The blue vector represents the initial predicted relation embedding. We use a scoring network to score a triple. (Color figure online)

respectively. Compared with using an enclosing subgraph to represent triples, our method can better emphasize the logical information implied by entities.

In this module, we extract all n-hop neighboring triples of the head and tail entity, respectively. For example, given a predicted triplet (u, r_t, v), we iteratively obtain the n-hop neighboring triples of the node u and node v through the breadth-first search(BFS) algorithm. Let $\mathcal{N}_n(u)$ and $\mathcal{N}_n(v)$ be set of triples in the n-hop neighborhood of node u and node v in the KG. For existing subgraph-based methods, they compute the enclosing subgraph by taking the intersection, $\mathcal{N}_n(u) \cap \mathcal{N}_n(v)$, of these k-hop neighborhood sets. However, these models will lack many neighboring triples. Therefore, we respectively use $\mathcal{N}_n(u)$ and $\mathcal{N}_n(v)$ to represent node u and node v, which can fully capture the logical rules implied by the neighboring triples of the head and tail entity.

Neighboring Relations Topology Graph. Since the n-hop neighboring triples extracted from the KG do not consider the connection structures between relations, we design the Neighboring Relations Topology Graph (NRTG) to address this problem. Inspired by TACT [2], to model the connection structures between relations of neighboring triples, we categorize relation pairs, consisting of neighboring and predicted relations, into six topological patterns. As illustrated in Fig. 3, there are six connection structures for connected relations in the knowledge graph, namely "head-to-tail", "tail-to-tail", "head-to-head", "tail-to-head", "parallel", and "loop". The connection structures are called topological patterns and they are named "H-T", "T-T", "H-H", "T-H", "PARA" and "LOOP" respectively.

Based on the definition of different topological patterns, we can convert the n-hop neighboring triples to NRTG, where the nodes represent the relations

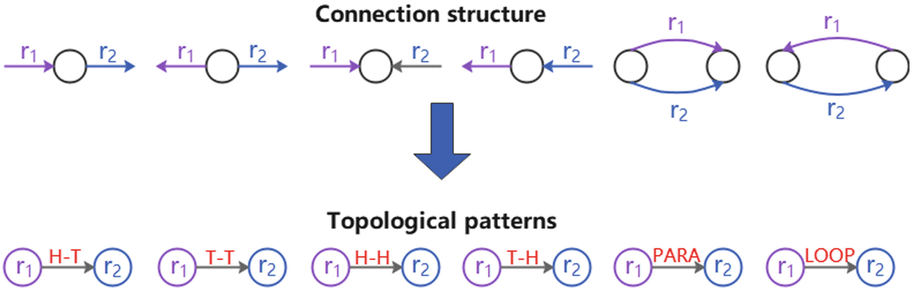

Fig. 3. An illustration of the transition from connection structure between relations to logical patterns.

and the edges indicate the topological patterns between neighboring relations and predicted relations. For example, the triples (e_1, r_1, e_2) and (e_2, r_2, e_3) are connected by e_2, and their topological pattern is "H-T". So, we construct a new triple $(r_1, H-T, r_2)$ in NRTG. For n-hop neighboring triples of head and tail entity, $\mathcal{N}_n(u)$ and $\mathcal{N}_n(v)$, we can convert the n-hop neighboring triples of entity u and entity v into NRTG in this way, respectively.

In this module, we extract the n-hop neighboring triples of the head and tail entity, respectively, and then convert the neighboring triples into NRTGs. As we can see, the NRTGs not only contain neighboring relations of the entities, but also take into account the connection structures between relations. Therefore, entity representation by NRTG can better mine the logical rules implied by entities predicted in KG. The detailed procedure is presented in Algorithm 1.

3.2 Information Aggregation Module

Based on the Neighboring Relations Topology Graphs (NRTGs) of the head and tail entity, we design a module to aggregate neighboring relations and topological patterns between relations in NRTGs as entity representations. Specifically, the information aggregation module is based on Relational Graph Convolutional Network (R-GCN) [15], and uses a message passing mechanism [5] in graph neural networks to update node representations. Finally, we use the average pooling of all the latent node representations to represent head, and tail entities of the predicted triple, respectively. As we can see, the entity representation contains the neighboring relations and topological patterns through this module. In this module, the message passing mechanism of node update is mainly divided into message function and aggregation function.

Message Function. The purpose of the message function is to pass the information to update the node in the NRTG. For each target node, it may receive messages from multiple nodes. Inspired by R-GCN [15], we define the message function of the k-th layer as:

Algorithm 1. Neighboring Relations Topology Graph Construction

Input: Origin graph $\mathcal{G}$, predicted triple (u, r_t, v), hop of neighboring n

Output: neighboring relations topology graph of head entity $\mathcal{G}_u$, neighboring relations topology graph of tail entity $\mathcal{G}_v$

```
$\mathcal{N}_0(u) \longleftarrow \{u\}$, $\mathcal{N}_0(v) \longleftarrow \{v\}$
for each triple $(s, r, t)$ in $\mathcal{G}$ do
    for $i \longleftarrow 1$ to $n$ do
        if $(s, r, t)$ connect with $\mathcal{N}_{i-1}(u)$ then
            $\mathcal{N}_i(u) \longleftarrow \mathcal{N}_{i-1}(u) \cup (s, r, t)$
        end if
        if $(s, r, t)$ connect with $\mathcal{N}_{i-1}(v)$ then
            $\mathcal{N}_i(v) \longleftarrow \mathcal{N}_{i-1}(v) \cup (s, r, t)$
        end if
    end for
end for
$\mathcal{G}_u \longleftarrow \{\}$, $\mathcal{G}_v \longleftarrow \{\}$
for each triple $(s, r, t)$ in $\mathcal{N}_n(u)$ do
    if $(s, r, t)$ connect with $(u_i, r_t, v_i)$ then
        Get pattern between $(s, r, t)$ and $(u, r_t, v)$ via the definition of topological patterns
        $\mathcal{G}_u \longleftarrow \mathcal{G}_u \cup (r_t, pattern, r)$
    end if
end for
for each triple $(s, r, t)$ in $\mathcal{N}_n(v)$ do
    if $(s, r, t)$ connect with $(u_i, r_t, v_i)$ then
        Get pattern between $(s, r, t)$ and $(u, r_t, v)$ via the definition of topological patterns
        $\mathcal{G}_v \longleftarrow \mathcal{G}_v \cup (r_t, pattern, r)$
    end if
end for
return $\mathcal{G}_u$, $\mathcal{G}_v$
```

$$m_t^k = \sum_{p=1}^{P} \sum_{s \in \mathcal{N}_r} a_{t,s}^k W_p^k n_s^{k-1}, \tag{1}$$

where $\mathcal{N}_r$ is the neighboring relations of predicted triple and P is the topological patterns between relations. n_s^{k-1} represent the node representation of the last layer, and it is represented as the embedding of relation when in the input layer. W_p^k represents the transformation matrix of the topological pattern p of the relation pair at the k-th layer. $a_{t,s}^k$ is the edge attention weight at the k-th layer corresponding to the edge connecting nodes s and t via topological patterns. The attention weights of the k-th layer are as follows:

$$a_{t,s}^k = \sigma\left(W_1^a \cdot ReLU\left(W_2^a\left[n_s^{k-1} \oplus n_t^{k-1} \oplus n_p^a\right]\right)\right). \tag{2}$$

Here W_1^a and W_2^a are the weight parameters in the attention mechanism, respectively. n_p^a is the attention vector of topological pattern p. $\sigma(\cdot)$ and $Relu(\cdot)$ are the activation functions.

Aggregation Function. The purpose of the aggregation function is to update the representation of the node according to the neighboring message. After obtaining the message vector m_t^k, we update the nodes in the NRTG. The aggregation function of the k-th layer is:

$$n_t^k = \sigma\left(W_0^k n_t^{k-1} + m_t^k\right), \tag{3}$$

where W_0^k is the weight parameters.

We acquire the node representations of the NRTG through the message function and aggregation function. Finally, the representation of the entity is obtained by average pooling of all the latent node representations in the NRTG:

$$e^k = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} n_i^k, \tag{4}$$

where $\mathcal{V}$ denotes the set of vertices in the graph.

In this module, based on two NRTGs, we adopt two identical R-GCN [15] to get the representation of head and tail entities, respectively.

3.3 Scoring Network and Loss Function

Scoring Network. The final step in our framework is to score the likelihood of predicted triples. For the predicted triple (u, r_t, v), the representations of entity u and entity v is obtained by the information aggregation module, and then we design a scoring network to output scores. The scoring function is defined as:

$$f(u, r_t, v) = W^T[e_u^k \oplus v_{r_t} \oplus e_v^u]. \tag{5}$$

In the scoring network, we obtain the scoring by a linear layer.

Loss Function. For each triple in the training graph, we sample a negative triple by replacing the head (or tail) entity. Afterward, we train our model to score positive triplets higher than the negative by using noise-contrastive hinge loss [1]. The specific loss function is as follows:

$$\mathcal{L} = \sum_{i=1}^{|\varepsilon|} \max\left(0, f\left(u_i', r_t', v_i'\right) - f\left(u, r_t, v\right) + \gamma\right), \tag{6}$$

where γ is the margin hyperparameter; ε is the set of all triplets in the neighboring relations topology graph. (u_i', r_t', v_i') denotes the i-th negative triple of the ground-truth triple (u, r_t, v).

Table 1. Statistics of inductive benchmarks. We use #E and #R and #TR to denote the number of entities, relations, and triples, respectively.

		WN18RR			FB15k-237			NELL-995		
		#R	#E	#TR	#R	#E	#TR	#R	#E	#TR
v1	Train	9	2746	6678	183	2000	5226	14	10915	5540
	Test	9	922	1991	146	1500	2404	14	225	1034
v2	Train	10	6954	18968	203	3000	12085	88	2564	10109
	Test	10	2923	4863	176	2000	5092	79	4937	5521
v3	Train	11	12078	32150	218	4000	22394	142	4647	20117
	Test	11	5084	7470	187	3000	9137	122	4921	9668
v4	Train	9	3861	9842	222	5000	33916	77	2092	9289
	Test	9	7208	15157	204	3500	14554	61	3294	8520

4 Experiments

In this section, there are the following parts. First, we introduce the experimental setup, such as datasets, training protocol, and evaluation protocol. Second, we compare our model with other approaches on several benchmark datasets. Third, we show the results of ablation studies to verify the effectiveness of our method. At last, we do some experiments to analyze the effect of hops on our model.

4.1 Experimental Setup

Datasets. In order to facilitate inductive testing, the test set needs to contain entities not seen in the training set. Therefore, we use some benchmark datasets for inductive relation prediction proposed in GraIL [17], which are derived from WN18RR [3], FB15k-237 [18], and NELL995 [22]. Specifically, each dataset consists of a pair of graphs: train-graph and ind-test-graph. We randomly select 10% of the edges/tuples in ind-test-graph as test edges. Details of the datasets are summarized in Table 1. The distribution of the six topological patterns in WN18RR and FB15k-237 is relatively uniform, and there are enough training examples. NELL-995 is a dataset with very sparse relationships, in which "PAPR" and "LOOP" are relatively rare. Furthermore, the same relational pairs have different topological patterns in each dataset.

Training Protocol. During training, we set the batch size to 32 and set the epoch to 100. We set the size of relations embedding to 32. In order to represent the entity, we convert 2-hop (or 3-hop) neighboring triples to Neighboring Relations Topology Graph (NRTG) and then use one-layer R-GCN [15] to represent an entity. We use Adam [9] to optimize all the parameters with an initial learning rate set at 0.01.

Evaluation Protocol. In the relation prediction task, the aim is to predict a triple (u, r_t, v) with u or v missing. We use the area under the precision-recall curve (AUC-PR) and Hits@10 to evaluate our models. To calculate the

Table 2. AUC-PR results on the inductive benchmark datasets extracted from WN18RR, FB15k-237 and NELL-995. The best score is in **bold** and second best score is underlined.

	WN18RR				FB15k-237				NELL-995			
	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
Neural-LP	86.02	83.78	62.90	82.06	69.64	76.55	73.95	75.74	64.66	83.61	87.58	85.69
DRUM	86.02	84.05	63.20	82.06	69.71	76.44	74.03	76.20	59.86	83.99	87.71	85.94
RuleN	90.26	89.01	76.46	85.75	75.24	88.70	91.24	91.79	84.99	88.40	87.20	80.52
GraIL	94.32	94.18	85.80	92.72	84.69	90.57	91.68	94.46	**86.05**	92.62	**93.34**	**87.50**
TACT	95.79	95.05	85.58	**96.60**	85.67	91.77	93.29	92.24	79.60	**94.40**	92.66	80.22
Our work	**97.35**	**96.55**	**89.86**	96.15	**90.79**	**95.32**	**95.40**	**95.83**	81.84	92.90	92.84	85.80

Table 3. H@10 results on the inductive benchmark datasets extracted from WN18RR, FB15k-237, and NELL-995.The best score is in **bold** and second best score is underlined.

	WN18RR				FB15k-237				NELL-995			
	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
Neural-LP	74.37	68.93	46.18	67.13	52.92	58.94	52.90	55.88	40.78	78.73	82.71	80.58
DRUM	74.37	68.93	46.18	67.13	52.92	58.73	52.90	55.88	19.42	78.55	82.71	80.58
RuleN	80.85	78.23	53.39	71.59	49.76	77.82	87.69	85.60	53.50	81.75	77.26	61.35
GraIL	82.45	78.68	58.43	73.41	64.15	81.80	82.83	89.29	**59.50**	**93.25**	91.41	**87.50**
TACT	84.04	81.63	58.59	76.34	64.39	82.11	84.04	90.58	57.50	92.96	**93.32**	74.07
Our work	**90.69**	**86.39**	**74.71**	**84.07**	**81.95**	**93.20**	**94.45**	**95.11**	59.00	91.80	89.80	81.25

AUC-PR, we replace the head or tail entity with a random entity to sample the negative triple, and then score the positive triples with an equal number of negative triples. To evaluate Hits@10, We select the top 10 triples among the 50 negative triples, and then calculate the proportion of correct triples.

4.2 Results and Analysis

We validate the models on classification metrics (AUC-PR) and ranking metrics (Hit@10), respectively. Then, we compare our method to several state-of-the-art methods on these metrics, such as NeuralLP [23], DRUM [13], RuleN [10], GraIL [17] and TACT [2].

Table 2 shows the mean AUC-PR results, averaged over 5 runs. The results show that our model achieves improvements on WN18RR and FB-237. Especially on the FB-237, the accuracy is improved by an average of 5%. Competitive results are achieved on the NELL-995. Table 3 shows the mean Hit@10 results, averaged over 5 runs. Our model achieves the state-of-the-art results on WN18RR and FB-237, and also competitive results on NELL-995.

As we can see, our model achieves huge improvements on all metrics on WN18RR and FB237. Therefore, our model successfully captures neighboring relations as well as the topological patterns between relations in entity rep-

Table 4. Ablation results the inductive benchmark datasets extracted from WN18RR, FB15k-237, and NELL-995. The best score is in **bold**

	WN18RR				FB15k-237				NELL-995			
	v1		v4		v1		v4		v1		v4	
	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10
Our work w/o NR	76.70	84.30	72.07	80.40	35.03	50.73	40.27	59.41	35.16	56.50	54.05	73.59
Our work w/o TP	69.71	85.37	71.52	77.92	34.45	70.73	55.52	90.66	41.17	49.50	25.93	63.67
Our work	**81.22**	**90.69**	**78.95**	**84.07**	**58.49**	**81.95**	**74.15**	**95.11**	**53.98**	**59.00**	**54.68**	**81.25**

resentation. Meanwhile, the improvement is particularly significant on FB237, which indicates that our method can better model complex topological structures between relations. Possible reasons why there is no improvement on NELL-995 are: compared to the other two datasets, the relational connection structures of the NELL-995 dataset are relatively sparse, which makes it difficult for our method to learn the topological patterns.

4.3 Ablation Study

In this part, we conduct ablation experiments to verify the effectiveness of our model. We mainly emphasize the effectiveness of our method by two experiments respectively: (1) Our work w/o NR (2) Our work w/o TP.

Our Work w/o NR. In order to learn the logical rules between the target nodes of predicted triples, the existing methods extract paths of head and tail nodes. However, these methods have obvious drawbacks: the incompleteness of the knowledge graph leads to missing paths that disconnect with target nodes, so the model lacks many useful neighboring relations. To verify the effectiveness of neighboring relations extraction in the relations topology module, we perform an ablation experiment with the subgraph construction method proposed by GraIL [17] instead of our method. We called this ablation experiment "Our work w/o NR".

Our Work w/o TP. In the relations topology module, we classify the connection structures between neighboring relations and predicted relations into six topological patterns, and use the information aggregation module based on R-GCN [15] to represent head and tail entities. To verify that we capture the topological patterns between neighboring relations and predicted relation in the entity representation, we set all topological patterns between relations to 1. We called this ablation experiment "Our work w/o TP".

Table 4 shows the performance of our method on three datasets. Results show that our model performs better than the two ablated models on three datasets. Experiments demonstrate that our method can more completely capture the relational logic rules for predicting triples, and better represent entities.

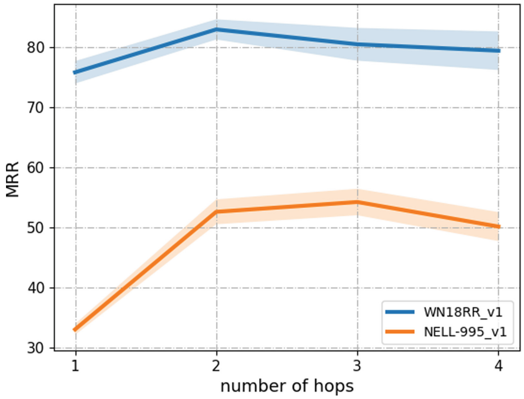

Fig. 4. The effect of different hops on MRR

4.4 Performance with Different Number of Hops

In this part, on WN18RR_v1 and NELL-995_v1, we extract the 1-hop, 2-hop, 3-hop and 4-hop neighboring triples of the head and tail entities respectively to construct NRTGs for inductive relation prediction. And we report the mean MRR, averaged over 5 runs. In Fig. 4, the performance of the model improves with the increase of the number of hops at the beginning, but after reaching 2 or 3 hops, the performance of the model does not improve, or even declines. The results show that the more hops, the more complete the logical information in the knowledge graph can be learned. However, the more hops will add a lot of noise information, which will reduce the performance of the model. Furthermore, with the number of hops increases, the fluctuation of the MRR value will also increase. Experiments demonstrate that our model can best learn the logical information implicit in the knowledge topology through the 2-hop (or 3-hop) NRTG.

5 Conclusion

We propose a novel entity representation method for inductive relation prediction. This entity representation method is based on a Neighboring Relations Topology Graph (NRTG), in which the nodes represent relations and the edges represent topological patterns between relations. The NRTG not only implies the logical rules of neighbor, but also is entity-independent. Thus, our model is able to make relation predictions in an inductive setting. Experiments demonstrate that our method significantly outperforms several existing state-of-the-art methods on benchmark datasets for the inductive link prediction task. In the future, we plan to extend our model further to capture the implicit logical rules in Few-shot Relations.

References

1. Bordes, A., et al.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems 26 (2013)
2. Chen, J., et al.: Topology-aware correlations between relations for inductive link prediction in knowledge graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35. 7, pp. 6271–6278 (2021)
3. Dettmers, T., et al.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, January 2018
4. Galárraga, L.A., et al.: AMIE: association rule mining under in-complete evidence in ontological knowledge bases. In: Proceedings of the 22nd international conference on World Wide Web, pp. 413–422 (2013)
5. Gilmer, J., et al.: Neural message passing for quantum chemistry. In: International Conference on Machine Learning. PMLR. 2017, pp. 1263–1272
6. Hamaguchi, T., et al.: Knowledge transfer for out-of-knowledge-base entities: a graph neural network approach. In: arXiv preprint arXiv:1706.05674 (2017)
7. Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: methods and applications. In: arXiv preprint arXiv:1709.05584 (2017)
8. Huang, X., et al.: Knowledge graph embedding based question answering. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 105–113 (2019)
9. Kingma, D.P., Ba, J.: Adam: a method for stochastic op- timization. In: arXiv preprint arXiv:1412.6980 (2014)
10. Meilicke, C., et al.: Fine-grained evaluation of rule-and embedding-based systems for knowledge graph completion. In: International Semantic Web Conference, pp. 3–20. Springer (2018)
11. Nathani, D., et al.: Learning attention-based embeddings for relation prediction in knowledge graphs. In: arXiv preprint arXiv:1906.01195 (2019)
12. Nguyen, D.Q., et al.: A novel embedding model for knowledge base completion based on convolutional neural network. In: arXiv preprint arXiv:1712.02121 (2017)
13. Sadeghian, A., et al.: Drum: end-to-end differentiable rule mining on knowledge graphs. In: Advances in Neural Information Processing Systems 32 (2019)
14. Scarselli, F., et al.: The graph neural network model. IEEE Trans. Neural Networks **20**(1), 61–80 (2008)
15. Schlichtkrull, M., et al.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference, pp. 593–607. Springer (2018)
16. Sun, Z., et al.: Rotate: knowledge graph embedding by relational rotation in complex space. In: arXiv preprint arXiv:1902.10197 (2019)
17. Teru, K., Denis, E., Hamilton, W.: Inductive relation pre- diction by subgraph reasoning. In: International Conference on Machine Learning. PMLR, pp. 9448–9457 (2020)
18. Toutanova, K., et al.: Representing text for joint embedding of text and knowledge bases. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1499–1509 (2015)
19. Tran, H.D., et al.: Towards nonmonotonic relational learning from knowledge graphs. In: International Conference on Inductive Logic Pro- gramming, pp. 94–107. Springer (2016)
20. Trouillon, T., et al.: Knowledge graph completion via complex tensor factorization. In: arXiv preprint arXiv:1702.06879 (2017)

21. Wang, H., et al.: Ripplenet: propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 417–426 (2018)
22. Xiong, W., Hoang, T., Wang, Y.: Deeppath: a reinforcement learning method for knowledge graph reasoning. In: arXiv preprint arXiv:1707.06690 (2017)
23. Yang, F., Yang, Z., Cohen, W.W.: Differentiable learning of logical rules for knowledge base reasoning. In: Advances in Neural Information Processing Systems 30 (2017)
24. Yu, H., Lu, J., Zhang, G.: Online topology learning by a gaussian membership-based self-organizing incremental neural network. In: IEEE Trans. Neural Networks Learn. Syst. 31(10), 3947–3961 (2019)
25. Yu, H., Lu, J., Zhang, G.: Topology learning-based Fuzzy random neural network for streaming data regression. IEEE Trans, Fuzzy Syst (2020)
26. Zhang, Z., et al.: Relational graph neural network with hierarchical at- tention for knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34(05), 9612–9619 (2020)
27. Zhang, Z., et al.: ERNIE: enhanced language representation with informative entities. In: arXiv preprint arXiv:1905.07129 (2019)