A Unified Joint Approach with Topological Context Learning and Rule Augmentation for Knowledge Graph Completion

Jingtao Guo, Chunxia Zhang*, **Lingxi Li, Xiaojun Xue, Zhendong Niu**
School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
{guo_jing_tao, cxzhang, lilingxi, xiaojunx, zniu}@bit.edu.cn

Abstract

Knowledge graph completion (KGC) task is to infer the missing knowledge in the knowledge graph based on known factual triples. However, present KGC approaches still face the following two challenges. Those methods perform simple linear update on relation representation, and only local neighborhood information is aggregated, which makes it difficult to capture logic semantic between relations and global topological context information. To tackle the above challenges, we propose a unified joint approach with Topological Context learning and Rule Augmentation (TCRA) for KGC. The TCRA framework consists of an entity topological context learning mechanism based on dual-branch hierarchical graph attention network, and a relation rule context learning mechanism based on Rule-Transformer and rule-to-relation aggregator. The former mechanism encodes the topological structure features of entities, aggregates the local neighborhood topological context information of entities on the three levels (entity, relation and triple), and build clusters of global head or tail entities related to the same relation. It can capture the local and global topological context information of entities related to the same relation. The latter mechanism introduces chain-like Horn rules as the context information of relations, and encodes the logical semantic of relations to enrich the relation representation. Experimental performances on three benchmark datasets FB15k-237, WN18RR and Kinship indicate the effectiveness and superiority of our proposed approach. The codes are publicly available.[1]

1 Introduction

Currently, large-scale knowledge graphs (KGs) such as Freebase (Bollacker et al., 2008), DBpedia (Lehmann et al., 2015) and NELL (Mitchell et al., 2018) have been widely used in knowledge-intensive applications including semantic retrieval, question answering, and recommendation systems. Specially, knowledge graphs provide strong technical support for applications such as public opinion monitoring, intelligent decision-making, and credit assessment. However, knowledge graphs constructed in a manual or automated way are usually incomplete and sparse. Knowledge graph completion (KGC) task addressed in this paper is a vital technique to infer new knowledge and complete missing entities and relations within knowledge graphs.

At present, KGC approaches are mainly divided into methods based on neural reasoning, symbolic reasoning, and neural-symbolic reasoning (Zhang et al., 2021). The first family techniques (Sun et al., 2019; Yang et al., 2015; Dettmers et al., 2018; Schlichtkrull et al., 2018; Zhang et al., 2022; Li et al., 2022) first embed entities and relations into a low-dimensional dense vector space. And then they measure the plausibility of unobserved triples by calculating their scores in the continuous space. The methods in the second family (Galárraga et al., 2015; Yang et al., 2017; Qu et al., 2020; Cheng et al., 2022, 2023) deduce general logical rules from knowledge graphs, and apply the logical rules to infer missing facts, which can achieve explainable reasoning. The third family approaches (Cheng et al., 2021; Lin et al., 2021; Tang et al., 2023) fuse the former two kinds of methods, and have become a mainstream technology in the fields of natural language processing and knowledge graph construction.

However, existing KGC methods still face the following problems: (a) the existing KGC approaches based on graph neural network (GNN) only perform simple linear updates on the relation representation, which can not capture the correlation information between the relations in KGs. (b) Those methods only aggregate local neighborhood

* Corresponding author.
[1] https://github.com/starlet122/TCRA.

Findings of the Association for Computational Linguistics: ACL 2024, pages 13686–13696
August 11-16, 2024 ©2024 Association for Computational Linguistics

information. Even if message passing is applied multiple times, it is difficult for nodes that are far apart in KGs to effectively transfer information. Hence, they cannot capture global topological context information.

To solve the above problems, we propose a unified joint representation learning approach with **T**opological **C**ontext learning and **R**ule **A**ugmentation (**TCRA**) for KGC task. The TCRA model adopts an encoder-decoder architecture. The encoder consists of two modules: the entity topology context learning module based on a dual-branch hierarchical graph attention network and the relation rule context learning module based on Rule-Transformer and rule-to-relation aggregator, which are used to generate embeddings of entities and relations, respectively.

Architecturally, the entity context learning module leverages hierarchical graph attention network (H-GAT) to aggregate local neighborhood structure information to generate local embeddings of entities. Parallelly, it builds global head entity or tail entity clusters related to the same relation to capture global semantic associations between entities. In addition, the relation context learning module leverages chain-like Horn rules as context information of relations, and designs the Rule-Transformer to encode the logical semantic information of the relations in rule body relation sequences to generate the representation of rule bodies. Further, the rule-to-relation aggregator fuses the rule body representations related to the rule head relations, thereby updating the representations of the rule head relations.

The contributions of this paper are summarized as follows:

- A unified joint representation learning approach with topological context learning and rule augmentation is proposed for KGC task. On the one hand, our TCRA can effectively model the local topological structure information of entities, and capture the global structural characteristics of entities in the form of entity clusters for enriching the context representation of entities. On the other hand, logical rules are introduced to mine logical semantic association of relations, and those rules are explicitly modeled to constrain relation embeddings for enhancing the representation of relations.

- We design an entity topological context learning mechanism based on dual-branch hierarchical graph attention network. That mechanism can capture the local structural features of entities by modelling the neighborhood information at three levels of entities, relations, and triples. Moreover, the cluster of global head or tail entities related to the same relation is introduced to capture the global structural characteristics of entities by using cluster encoder and cluster-to-entity aggregator.

- A relation rule context learning mechanism based on Rule-Transformer and rule-to-relation aggregator is developed. That mechanism can effectively capture the logical association between relations in the rule body, and simultaneously aggregate the rule body representation associated with a certain rule head relation for enriching the representation of the relations.

- Extensive comparative experiments on three benchmark datasets FB15k-237, WN18RR and Kinship show that our model TCRA outperforms the state-of-the-art methods.

2 Related Work

KGC approaches can be roughly classified into methods based on neural reasoning, symbolic reasoning, and neural-symbolic reasoning.

Neural reasoning is also known as knowledge graph embedding (KGE) or knowledge graph representation learning. Further, the present neural reasoning techniques can roughly fall into three categories: translational distance based models, semantic matching based models, deep learning based models. The basic idea of the translational distance models is to regard relations as conversion factors between head and tail entities (Bordes et al., 2013; Sun et al., 2019). The semantic matching models calculate the confidence of triples by measuring the similarity of the underlying semantics between entities and relations (Nickel et al., 2011; Yang et al., 2015). The deep learning models have stronger representation and generalization ability than the former two kinds of methods. Recently, GNN is utilized to fulfill the KGC task (Schlichtkrull et al., 2018; Vashishth et al., 2019; Li et al., 2022), which can learn the topological structure of KGs in an end-to-end manner. SE-GNN (Li et al., 2022) introduced different levels of semantic evidence to explain the extrapolation ability of KGE model and explicitly treated each semantic evidence as a different neighbor pattern.

The methods based on symbolic reasoning deduce general logical rules from the knowledge graph, and then apply the logical rules to infer the missing facts, which have good interpretability. Many rule mining approaches have been developed to extract rules from large-scale KGs, such as AMIE+ (Galárraga et al., 2015), RNNLogic (Qu et al., 2020), RLogic (Cheng et al., 2022) and NCRL (Cheng et al., 2023).

Recent techniques used in knowledge graph reasoning combine neural reasoning and symbolic reasoning. One of the fused methods is the symbol-driven neural reasoning method, which utilizes logic to enhance embedding. Specifically, the fusion mechanism can be classified into the following three types of ones. The first family of mechanisms including KALE (Guo et al., 2016) and RUGE (Guo et al., 2018) are to use logical rules as additional regularization for KGE training. The second kind of methods such as IterE (Zhang et al., 2019) and UniKER (Cheng et al., 2021) are to employ logical rules to obtain new hidden triples and generate additional triples for KGE training. The third kind of approaches such as RPJE (Niu et al., 2020), RulE (Tang et al., 2023) are to leverage logical rules as additional information to enhance KGs representation learning. This paper focuses on the symbol-driven neural reasoning method to solve KGC task.

3 Methodology

3.1 Problem Formulation

A knowledge graph G can be represented as a set of factual triples in the form of $G = \{(h, r, t)|h, r \in E, r \in R\}$, where E denotes a set of entities, R is a set of relations, h means a head entity, r is a relation, and t is a tail entity. KGC task means to predict the remaining missing element given two elements of a triple, including predicting the tail entity t in $\langle h, r, ?\rangle$ given h and r, or predicting the head entity h in $\langle ?, r, t\rangle$ given r and t, where $?$ is the element to be inferred.

3.2 Overview of TCRA Framework

Our TCRA model for knowledge graph completion adopts the encoder-decoder architecture. The encoder consists of two modules: the entity topology context learning module based on a dual-branch hierarchical graph attention network and the relation rule context learning module based on Rule-Transformer and rule-to-relation aggregator. The former module encodes the topological structure characteristics of entities. On the one hand, it aggregates three levels of local neighborhood topological context of entities including entity, relation and triple levels. On the other hand, global head or tail entity clusters related to the same relation are built to capture global semantic associations between entities. Thereby, we generate embedding representation of entities, which can capture the local structural features and the global topological context of entities. The latter module encodes the causal association information between relations, and introduces chain-like Horn rules as the context information of relations to learn the relation embedding, which aims to encode the logical semantic information of relation. ConvE is chosen as the decoder. The overall architecture of our TCRA model is shown in Figure 1.

3.3 Entity Topology Context Learning

A dual-branch hierarchical graph attention network is designed to learn entity topological context of entities. H-GAT is exploited to aggregate local neighborhood topological structure information to generate local embeddings of entities. Additionally, the global head entity clusters or tail entity clusters associated with the same relation is introduced to capture the global association information between entities.

3.3.1 Local Branch: Hierarchical Graph Attention Network

Inspired by the works in Li et al. (2022), the neighborhood topological context information on the three levels of entity, relation, and triple are jointly modeled as information sources of complementary reinforcing local neighbor structure to obtain richer local structure features of entities. H-GAT is leveraged to generate those three-levels local representations of the central entity. Further, the three embeddings are integrated to generate the updated embedding of that entity. In addition, a multi-layer iterative aggregation mechanism is introduced to capture multi-hop neighbor information and deep interactions between different levels of information. We will explain the single aggregation layer.

For the entity-level neighborhood information, we use s_i^{ent} for the entity-level local representation

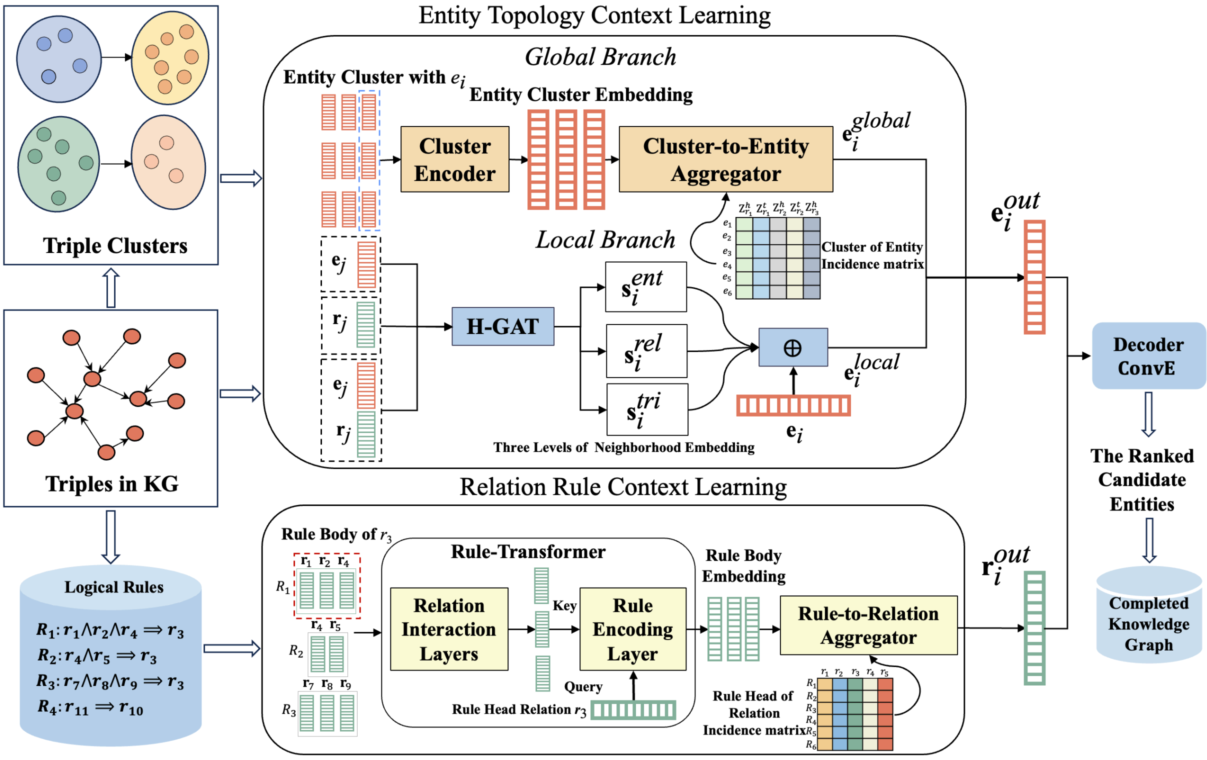

Figure 1: The framework of our TCRA model for knowledge graph completion.

of the central entity e_i, which is computed as:

$$\mathbf{s}_i^{ent} = \sigma\left(\sum_{(e_j,r_j)\in\mathcal{N}_i} \alpha_{ij}^{ent} W_{ent}\mathbf{e}_j\right), \quad (1)$$

where $\mathcal{N}_i$ denotes the set of neighbor entities and connecting relation of e_i in train set, $W_{ent} \in \mathbb{R}^{d_G\times d_G}$ is the linear transformation matrix, d_G is the embedding dimension of entities and relations in G. $\mathbf{e}_j$ is the embedding representation of e_j. σ is a non-linear activation function. α_{ij}^{ent} is an aggregation attention score of neighbor entity e_j, which is calculated as follows:

$$\alpha_{ij}^{ent} = \frac{\exp\left(\mathbf{e}_j^T\mathbf{e}_i\right)}{\sum_{(e_k,r_k)\in\mathcal{N}_i}\exp\left(\mathbf{e}_k^T\mathbf{e}_i\right)}. \quad (2)$$

The relation-level neighborhood information $\mathbf{s}_i^{rel}$ is computed as:

$$\mathbf{s}_i^{rel} = \sigma\left(\sum_{(e_j,r_j)\in\mathcal{N}_i} \alpha_{ij}^{rel} W_{rel}\mathbf{r}_j\right), \quad (3)$$

$$\alpha_{ij}^{rel} = \frac{\exp\left(\mathbf{r}_j^T\mathbf{e}_i\right)}{\sum_{(e_k,r_k)\in\mathcal{N}_i}\exp\left(\mathbf{r}_k^T\mathbf{e}_i\right)}, \quad (4)$$

where $\mathbf{r}_j$ is the embedding representation of r_j.

The triple-level neighborhood information $\mathbf{s}_i^{tri}$ is calculated as:

$$\mathbf{s}_i^{tri} = \sigma\left(\sum_{(e_j,r_j)\in\mathcal{N}_i} \alpha_{ij}^{tri} W_{tri}\varphi(\mathbf{e}_j,\mathbf{r}_j)\right), \quad (5)$$

$$\alpha_{ij}^{tri} = \frac{\exp\left(\varphi(\mathbf{e}_j,\mathbf{r}_j)^T\mathbf{e}_i\right)}{\sum_{(e_k,r_k)\in\mathcal{N}_i}\exp\left(\varphi(\mathbf{e}_j,\mathbf{r}_j)^T\mathbf{e}_i\right)}, \quad (6)$$

where $\varphi(\mathbf{e}_j,\mathbf{r}_j) = \mathbf{e}_j * \mathbf{r}_j$.

A multi-layer iterative aggregation approach is adopted to integrate the neighborhood information on levels of entity, relation, and triple. At each layer l, the three-levels embeddings $\left(\mathbf{s}_i^{ent}\right)^l$, $\left(\mathbf{s}_i^{rel}\right)^l$ and $\left(\mathbf{s}_i^{tri}\right)^l$ with original input embedding $\mathbf{e}_i^l$ are integrated to be feed into the next layer. In first layer, the embedding $\mathbf{e}_i^1$ is initialized. Then, after K layers aggregation, $\mathbf{e}_i^K$ is regarded as the final entity local topological context embedding $\mathbf{e}_i^{local}$.

$$\mathbf{e}_i^{l+1} = \mathbf{e}_i^l + \left(\mathbf{s}_i^{ent}\right)^l + \left(\mathbf{s}_i^{rel}\right)^l + \left(\mathbf{s}_i^{tri}\right)^l, \quad (7)$$

$$\mathbf{e}_i^{local} = \mathbf{e}_i^K. \quad (8)$$

3.3.2 Global Branch: Cluster Encoder and Cluster-to-Entity Aggregator

We observed that entities with the same relation exist global topological associations with each other in KGs. Thereby, the factual triples containing the same relation are constructed into a cluster of triples for capturing global topological context information between entities. Further, each relation has a corresponding head entity cluster $Z^h_{r_i}$ and a tail entity cluster $Z^t_{r_i}$. Hence, the cluster triple can be represented as $C_{r_i} = \{(Z^h_{r_i}, r_i, Z^t_{r_i}) \mid r_i \in R\}$.

First, we collect the representations of entities contained in each entity cluster and generate the entity cluster representation, as shown in Eq.(9).

$$\mathbf{U} = \mathbf{D}^{-1}\mathbf{H}^T\mathbf{E}W_c, \tag{9}$$

where $\mathbf{U} \in \mathbb{R}^{2|R|\times d_S}$ is the entity cluster embedding matrix, d_S is the embedding dimension of entity clusters. $\mathbf{H} \in \mathbb{R}^{|E|\times 2|R|}$ is the incidence matrix, which shows whether an entity belongs to a cluster. $\mathbf{D} \in \mathbb{R}^{2|R|\times 2|R|}$ is a diagonal matrix of cluster degree, i.e. the number of entities contained in clusters, and $D_{jj} = \sum_i H_{ij}$. $\mathbf{E} \in \mathbb{R}^{|E|\times d_G}$ is the entity embedding matrix. $W_c \in \mathbb{R}^{d_G\times d_S}$ is a trainable weight matrix.

Then, we design a cluster-to-entity aggregator to aggregate the representation of the entity cluster to which the entity e_i belongs, and generate the global context representation $\mathbf{e}_i^{global}$ of the entity e_i,

$$\mathbf{e}_i^{global} = \sigma\left(\frac{1}{\sum_j h_{i,j}}\mathbf{h}_i^T\mathbf{U}W_g\right), \tag{10}$$

where $\mathbf{h}_i \in \mathbb{R}^{2|R|}$ is an incidence vector, which indicates whether the entity e_i belongs to cluster j by taking values 0 or 1. $W_g \in \mathbb{R}^{d_S\times d_G}$ is a trainable weight matrix. Finally, the entity embedding is calculated as $\mathbf{e}_i^{out} = \mathbf{e}_i^{local} + \mathbf{e}_i^{global}$.

3.4 Relation Rule Context Learning

The relation rule context learning module is intended to encode the logical semantic features of relations. Horn rule is a special first-order logic rule in the language of symbolic logic, typically is expressed in the form of "*body* $\Rightarrow$ *head*". The body of a Horn rule is defined as a connection normal form (CNF), which connects a set of predicates through logical connectives. Here, the head is a single predicate. With regard to relation context learning, chain-like Horn rules are mined in the following form:

$$r_h(x,y) \Leftarrow r_{b_1}(x,z_1)\wedge\cdots\wedge r_{b_n}(z_{n-1},y). \tag{11}$$

Here, $r_{b_1}(x,z_1)\wedge\cdots\wedge r_{b_n}(z_{n-1},y)$ is the rule body, $r_h(x,y)$ is the rule head. Integrating rule head and rule body, we denote a Horn rule as (R_b, r_h), where the rule body $R_b = [r_{b_1},\ldots,r_{b_n}]$. The rule body of a chain-like Horn rule can be regarded as a relation sequence. In addition, the length of a Horn rule refers to the number of predicates (or atoms) in its body.

Chain-like Horn rules are introduced to capture logical semantic of relations. Thereby, a Rule-Transformer is designed to mine the interactive information between relations in the rule body by encoding the rule body. It includes N Transformer encoder layers. We then aggregate the rule body embeddings to update the representation of the corresponding rule head relation. For example, the relation r_3 in Figure 1 has three rule body relation sequences.

$$\begin{aligned} N_{r_3} &= \{R_1, R_2, R_3\} \\ &= \{(r_1,r_2,r_4),(r_4,r_5),(r_7,r_8,r_9)\}. \end{aligned} \tag{12}$$

Logical rules focus on capturing the semantic constraints among different types of relations, which have the rich semantic intension correlations. Inspired by this observation, the relation interaction layers are designed to learn the semantic associations of the relations in the rule body and update the representation of each relation in the rule body. At first, for the rule (R_1, r_3), i.e., $r_1 \wedge r_2 \wedge r_4 \Rightarrow r_3$, the relation interaction layers use the first former N-1 Transformer encoder layers and adopt multi-head self-attention mechanism. The input $\mathbf{E}_{R_1}$ of the relation interaction layers is the representation matrix of the rule body sequence R_1, and the output $\mathbf{E}'_{R_1}$ is the encoding matrix of R_1,

$$\mathbf{E}'_{R_1} = transformer\ encoders\left(\mathbf{E}_{R_1}\right), \tag{13}$$

where $\mathbf{E}_{R_1}, \mathbf{E}'_{R_1} \in \mathbb{R}^{l\times d_G}$.

Secondly, the rule encoding layer employes the N-th Transformer encoder layer and adopts a multi-head cross-attention mechanism to generate the representation of the rule body for r_3,

$$\mathbf{e}_{R_i}^{out} = transformer\ encoder\left(\mathbf{E}'_{R_i}, \mathbf{r}_3\right). \tag{14}$$

Finally, we develop a rule-to-relation aggregator to aggregate the rule body representation related to the rule head relation r_i, introduce the rule head association matrix, and update the representation of the rule head relation r_i,

$$\mathbf{r}_i = \sigma\left(\frac{1}{\sum_j h_{i,j}}\mathbf{h}_i^T\mathbf{B}W_r\right), \tag{15}$$

where $\mathbf{H} \in \mathbb{R}^{|B| \times \mathbf{2}|R|}$ is the incidence matrix, which indicates whether the rule body b corresponds to the rule head relation r. $h_{b,r} = 1$ if $b \implies r$ otherwise $h_{b,r} = 0$. $|B|$ represents the number of rules. $\mathbf{h}_i \in \mathbb{R}^{|B|}$ is an incidence vector, which indicates whether the rule body b_j corresponds to the rule head relation r_i. $\mathbf{B} \in \mathbb{R}^{|B| \times d_R}$ is the rule body embedding matrix, which is output by the rule encoding layer. d_R is the embedding dimension of rule bodies. $W_r \in \mathbb{R}^{d_R \times d_G}$ is a trainable weight matrix.

3.5 Decoder

We leverage the embeddings of entities and relations to perform KGC task. ConvE is chosen as the decoder, which uses a 2D convolutional neural network to match the query (h, r) and the answer t. The ConvE's scoring function is shown in Eq.(16):

$$f(h, r, t) = \sigma(f(\sigma([\tilde{h}; \tilde{r}]) * \psi))W_d)t, \tag{16}$$

where $\tilde{h}$ and $\tilde{r}$ represent the 2D tensor corresponding $\mathbf{h}$ and $\mathbf{r}$ respectively, $*$ denotes the convolution operator. ψ represents a set of convolution kernels. $f(\cdot)$ is the vectorized function. W_d is a learnable weight matrix. σ is the ReLU activation function.

The standard binary cross-entropy loss function is chosen and label smoothing is adopted:

$$\mathrm{L} = -\frac{1}{N} \sum_t (t_i \log(p_i) + (1 - t_i) \log(1 - p_i)), \tag{17}$$

where N denotes the number of entities, t_i is the true label of triple i, and p_i is the corresponding prediction score.

4 Experiments

4.1 Experiment Settings

Datasets and Rules. In order to evaluate the performance of our proposed method on the KGC task, we have conducted comparative experiments on three public benchmark datasets, including FB15k-237 (Toutanova and Chen, 2015), WN18RR (Dettmers et al., 2018), and Kinship (Kok and Domingos, 2007). The statistics of datasets is given in Table 1. For rule mining, RNNLogic (Qu et al., 2020) is chosen as our rule mining tool. In addition, we have also conducted experiments to demonstrate the impact of the selection of decoders (ConvE and DistMult) and rule miners (RNNLogic and RLogic). Additional experimental details are listed in Appendix A.1.1.

Dataset	#Ent	#Rel	#Tri		
			Train	Valid	Test
FB15k-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,134
Kinship	104	25	8,544	1,068	1,074

Table 1: Statistics of datasets used in the experiments ("#Ent", "#Rel", and "#Tri" denote entity, relation, and triple, respectively).

Baselines. We compare our TCRA model with a comprehensive suite of baselines, including three classes of models: (a) KGE-based models: RotatE (Sun et al., 2019), DistMult (Yang et al., 2015), ConvE (Dettmers et al., 2018), R-GCN (Schlichtkrull et al., 2018), SACN (Shang et al., 2019), CompGCN (Vashishth et al., 2019), LTE-KGE (Zhang et al., 2022), SE-GNN (Li et al., 2022), KGT5 (Saxena et al., 2022), (b) Rule learning-based models: AMIE+ (Galárraga et al., 2015), NeuralLP (Yang et al., 2017), RNNLogic (Qu et al., 2020), RLogic (Cheng et al., 2022), NCRL (Cheng et al., 2023), (c) joint models: RuleIC (Lin et al., 2021), RulE (Tang et al., 2023).

Evaluation metrics. KGC performance can be evaluated through five common evaluation metrics: MR (the Mean Rank), MRR (Mean Reciprocal Rank), Hits@N for N is 1,3 and 10 (the proportion of correct entity rankings in top-N).

4.2 Experimental Results

4.2.1 Performance Comparison

As shown in Table 2 and Table 3, TCRA achieves competitive performance on FB15k-237, WN18RR and Kinship. Especially on Kinship, we obtain the highest MRR, Hits@1, and Hits@3 that are 1.8%, 2.5%, and 1.7% higher than the best baseline, respectively.

Comparison with KGE-based and rule learning-based models. Our TCRA is essentially a rule-enhanced GNN-based KGE model. (a) TCRA was first compared with traditional KGE models, including the translational distance model RotatE, the semantic matching model DistMult, and the CNN-based model ConvE. Table 2 and Table 3 show that TCRA achieves the higher performance than those of RotatE, DistMult and ConvE, indicating the effectiveness of our GNN architecture. (b) We further compare TCRA with the GNN-based KGE models including R-GCN, SACN, CompGCN, LTE-KGE and SE-GNN. The performance of TCRA is superior to those of all

Models	FB15k-237					WN18RR				
	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
TransE(Bordes et al., 2013)$^{\diamond}$	173	0.330	23.1	36.9	52.8	3380	0.223	1.40	40.1	52.9
RotatE (Sun et al., 2019)$^{\diamond}$	177	0.338	24.1	37.5	53.3	3340	0.476	42.8	49.2	57.1
DistMult (Yang et al., 2015)$^{\diamond}$	173	0.308	21.9	33.6	48.5	4723	0.439	39.5	45.2	53.3
ConvE (Dettmers et al., 2018)$^{\diamond}$	244	0.325	23.7	35.6	50.1	4187	0.430	40.0	44.0	52.0
R-GCN (Schlichtkrull et al., 2018)$^{\diamond}$	-	0.248	15.1	-	41.7	-	-	-	-	-
SACN (Shang et al., 2019)$^{\diamond}$	-	0.350	26.0	39.0	54.0	-	0.470	43.0	48.0	54.0
CompGCN (Vashishth et al., 2019)$^{\diamond}$	197	0.355	26.4	39.0	53.5	3533	0.479	44.3	49.4	54.6
LTE-KGE (Zhang et al., 2022)	182	0.355	26.4	38.9	53.5	3290	0.472	43.7	48.5	54.4
SE-GNN (Li et al., 2022)	157	0.365	27.1	39.9	54.9	**3211**	0.484	44.6	50.9	57.2
KGT5 (Saxena et al., 2022)	-	0.276	21.0	-	41.4	-	0.508	48.7	-	54.4
NeuralLP (Yang et al., 2017)	-	0.237	17.3	25.9	36.1	-	0.381	36.8	38.6	40.8
RNNLogic (Qu et al., 2020)	232	0.344	25.2	38.0	53.0	4615	0.483	44.6	49.7	55.8
RLogic (Cheng et al., 2022)	-	0.310	20.3	-	50.1	-	0.470	44.3	-	53.7
NCRL (Cheng et al., 2023)	-	0.310	22.0	-	48.2	-	**0.670**	**56.8**	-	**85.2**
Rule-IC (Lin et al., 2021)	166	0.355	27.2	40.2	55.2	3304	0.436	39.9	45.1	54.5
RulE (Tang et al., 2023)	-	0.354	26.1	39.1	54.3	-	0.506	46.6	**52.2**	58.9
TCRA(DistMult+RLogic)	183	0.344	25.3	37.6	52.5	4018	0.453	41.6	46.6	52.6
TCRA(DistMult+RNNLogic)	183	0.344	25.3	37.8	52.6	3171	0.453	41.6	46.3	52.7
TCRA(ConvE+RLogic)	160	0.365	27.1	40.2	55.1	**3210**	0.492	45.2	50.4	57.0
Our TCRA (ConvE+RNNLogic)	**156**	**0.367**	**27.5**	**40.3**	**55.4**	3303	0.496	45.7	51.1	57.4

Table 2: Results of KGC on FB15k-237 and WN18RR. $^{\diamond}$ means that the results are reported from (Li et al., 2022). Results of NeuralLP are taken from (Qu et al., 2020). Other results are from the original papers.

Models	Kinship			
	MRR	Hits@1	Hits@3	Hits@10
TransE(Bordes et al., 2013)$^{\diamond}$	0.251	1.62	37.8	72.8
RotatE (Sun et al., 2019)$^{\diamond}$	0.651	50.4	75.5	93.2
DistMult (Yang et al., 2015)$^{\diamond}$	0.354	18.9	40.0	75.5
ConvE (Dettmers et al., 2018)	0.833	73.8	91.7	98.1
SE-GNN (Li et al., 2022)†	0.848	76.4	91.9	**98.4**
NeuralLP (Yang et al., 2017)$^{\diamond}$	0.302	16.7	33.9	59.6
RNNLogic (Qu et al., 2020)	0.722	59.8	81.4	94.9
RLogic (Cheng et al., 2022)	0.580	43.4	-	87.2
NCRL (Cheng et al., 2023)	0.650	49.4	-	93.6
RulE (Tang et al., 2023)	0.740	62.0	82.9	95.7
TCRA(DistMult+RLogic)	0.570	42.4	63.9	89.2
TCRA(DistMult+RNNLogic)	0.578	43.5	64.5	89.3
TCRA(ConvE+RLogic)	0.847	76.1	92.0	98.2
Our TCRA (ConvE+RNNLogic)	**0.866**	**78.9**	**93.6**	98.3

Table 3: Results of KGC on Kinship. $^{\diamond}$ means that the results are reported from (Tang et al., 2023). † denotes that we reproduce the results of SE-GNN. Results of ConvE and RLogic are taken from (Zeb et al., 2021) and (Cheng et al., 2023) respectively. Other results are from the original papers.

other models about MRR, Hits@1, Hits@3, and Hits@10 on FB15k-237 and WN18RR, while it outperforms all other models about MRR, Hits@1 and Hits@3 on Kinship. Those experimental results in Table 2 and Table 3 demonstrate the effectiveness of our proposed entity topology context learning and the relation context learning mechanism. The former mechanism solves the problem of effective message transmission between adjacent distant entities to supplement rich topological information. Within the latter mechanism, logical rules are introduced as relation context to capture the logical semantic characteristics of relations. (c) The performance of TCRA is better than the rule learning-based model including AMIE+, NeuralLP, RNNLogic, RLogic and NCRL in most cases. This can be attributed to the fact that TCRA jointly learns logical rules and topological structures, which can encode the knowledge graph in a more complementary way and obtain better KGC performance.

Comparison with joint models. Compared with joint models including Rule-IC and RulE, TCRA achieves the convincing performance. Specifically, TCRA obtain the best performance on FB15k-237 and Kinship, and better performance than Rule-IC on WN18RR. The reason is that the logical rules and topology structures in WN18RR and Kinship contain more complementary information.

4.2.2 Ablation Study

We have conducted ablation experiments to demonstrate how Cluster Encoder and cluster-to-entity Aggregator (abbreviated as CEA) and Relation Rule Context learning (abbreviated as RRC) contribute to the overall performance of TCRA. "w/o CEA" means that CEA is removed from TCRA, while "w/o RRC" denotes that RRC is removed from TCRA. "w/o CEA&RRC" represents that TCRA simultaneously removes CEA

Models	FB15k-237				WN18RR				Kinship			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TCRA	**0.367**	**27.5**	**40.3**	**55.4**	**0.496**	**45.7**	**51.1**	**57.4**	**0.866**	**78.9**	**93.6**	98.3
w/o CEA	0.364	26.9	40.2	55.3	0.490	44.5	51.0	57.7	0.860	78.2	92.8	**98.6**
w/o RRC	0.365	27.1	40.0	55.3	0.494	45.5	50.8	57.2	0.853	77.1	92.3	98.4
w/o CEA&RRC	0.365	27.1	39.9	54.9	0.484	44.6	50.9	57.2	0.848	76.4	91.9	98.4
w/o CEA&Rule-Transformer	0.365	27.1	40.0	55.3	0.494	45.5	50.6	57.0	0.855	77.1	92.9	98.2

Table 4: Ablation experiment results.

and RRC. "CEA&Rule-Transformer" means that TCRA simultaneously eliminates CEA and Rule-Transformer. Table 4 summarizes the results of our ablation experiments on three datasets.

For WN18RR and Kinship, these four ablation models have decreased significantly on MRR, Hits@1, and Hits@3 metrics. The ablation experiments not only indicate the effectiveness of the proposed global aggregation and relation rule context learning mechanism, but also show the two modules CEA and RRC complementarily contribute to our TCRA model. In particularly, the performance of "w/o CEA&RRC" is worse than that of "w/o CEA&Rule-Transformer", which supports the effectiveness and correctness of logical rules.

4.2.3 Hyperparameter Sensitivity Analysis

This subsection analyzes the sensitivity of hyperparameters used in TCRA, including number of rules and encoder layers in Rule-Transformer. The corresponding experimental results are illustrated in Figure 2. Additional hyperparameter experiments about learning rate and embedding size are provided in the appendix A.1.2.

First, we analyze the effects of different number of rules on the performance of our model. We randomly select a fixed number of rules from the mined logical rules for training. For FB15k-237, the overall trend of performance is flat. For WN18RR, the performance of TCRA generally shows a decline first and then an increase. For Kinship, four curves show a trend of first falling and then rising.

Further, we investigated the sensitivity of the number of encoder layers used in the Rule-Transformer, and its value range of N is in $\{2,3,4,5,6,7\}$. The performance of TCRA shows a trend of rising first and then falling on all three datasets. All in all, our model TCRA is not sensitive to number of rules and encoder layers in the Rule-Transformer.

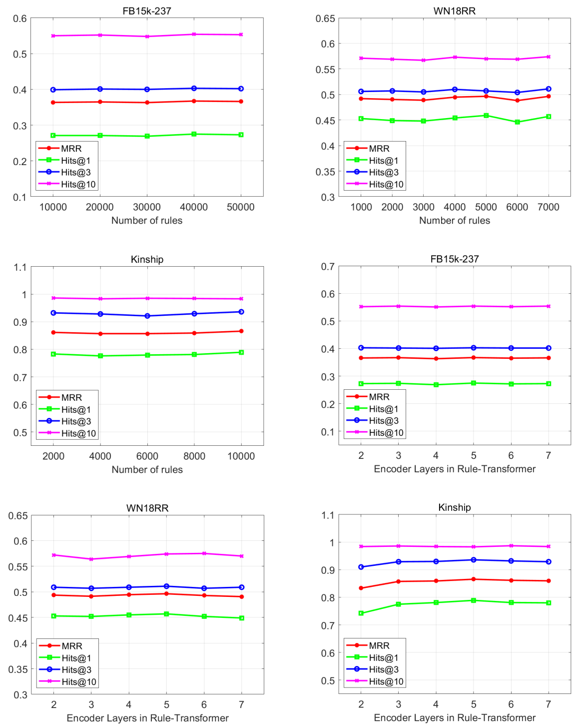

Figure 2: Hyperparameter experiments about number of rules and encoder Layers in Rule-Transformer.

5 Conclusion and Future Work

In this paper, a unied joint approach with Topological Context learning and Rule Augmentation (TCRA) has been proposed to perform knowledge graph completion, which can infer missing knowledge in a more complementary way. The designed entity topological context learning mechanism based on dual-branch hierarchical graph attention network can capture local structural features of entities at three levels, and incorporate global entity cluster to capture the global structural characteristics of entities. Meanwhile, the relation rule context learning mechanism based on Rule-Transformer and rule-to-relation aggregator is developed to facilitate adequate interaction between relations by leveraging chain-like Horn rules, which can capture the logical semantic of relations to enrich the relation representation. Extensive experiments demonstrate that the proposed TCRA model outperforms present methods. In the future, we will further design an iterative manner to implement rule-enhanced KGE methods on this basis, so as to achieve full complementarity between logical rules and KGE.

Limitations

Knowledge graphs contain rich interrelated knowledge, including structural association and semantic association. The models based on GNNs are easily affected by the sparsity of local structures. To alleviate the local sparsity, we build clusters of global head or tail entities related to the same relation to learn global structure association about entities. In addition, the utilization of logical rules increases the risk of being affected by noisy data in KGs. We make the attempt to randomly select a fixed number of rules to participate in training to reduce risks.

Ethics Statement

We used the publicly available datasets FB15k-237, WN18RR, and Kinship to train and evaluate KGC models, and there is no ethical issue.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (62072039).

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, page 27872795.

Kewei Cheng, Nesreen K Ahmed, and Yizhou Sun. 2023. Neural compositional rule learning for knowledge graph reasoning. In *Proceedings of International Conference on Learning Representations*.

Kewei Cheng, Jiahao Liu, Wei Wang, and Yizhou Sun. 2022. Rlogic: Recursive logical rule learning from knowledge graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 179–189.

Kewei Cheng, Ziqing Yang, Ming Zhang, and Yizhou Sun. 2021. Uniker: A unified framework for combining embedding and definite horn rule reasoning for knowledge graph inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 9753–9771.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1811–1818.

Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. 2015. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal*, 24(6):707–730.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2016. Jointly embedding knowledge graphs and logical rules. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 192–202.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2018. Knowledge graph embedding with iterative guidance from soft rules. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4816–4823.

Stanley Kok and Pedro Domingos. 2007. Statistical predicate invention. In *Proceedings of the 24th International Conference on Machine Learning*, pages 433–440.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, S. Auer, and Christian Bizer. 2015. Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6:167–195.

Ren Li, Yanan Cao, Qiannan Zhu, Guanqun Bi, Fang Fang, Yi Liu, and Qian Li. 2022. How does knowledge graph embedding extrapolate to unseen data: a semantic evidence view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5781–5791.

Qika Lin, Jun Liu, Yudai Pan, Lingling Zhang, Xin Hu, and Jie Ma. 2021. Rule-enhanced iterative complementation for knowledge graph reasoning. *Information Sciences*, 575:66–79.

Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. 2018. Never-ending learning. *Communications of the ACM*, 61(5):103–115.

Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel, et al. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 3104482–3104584.

Guanglin Niu, Yongfei Zhang, Bo Li, Peng Cui, Si Liu, Jingyang Li, and Xiaowei Zhang. 2020. Rule-guided compositional representation learning on knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2950–2958.

Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. 2020. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *Proceedings of International Conference on Learning Representations*.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 2814–2828.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of the 15th International Conference on Extended Semantic Web Conference*, pages 593–607.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3060–3067.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Xiaojuan Tang, Song-Chun Zhu, Yitao Liang, and Muhan Zhang. 2023. Rule: Neural-symbolic knowledge graph reasoning with rule embedding. In *International Conference on Learning Representations*.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based multi-relational graph convolutional networks. In *Proceedings of International Conference on Learning Representations*.

Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of International Conference on Learning Representations*.

Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 23162325.

Adnan Zeb, Anwar Ul Haq, Defu Zhang, Junde Chen, and Zhiguo Gong. 2021. Kgel: A novel end-to-end embedding learning framework for knowledge graph completion. *Expert Systems with Applications*, 167:114164.

Jing Zhang, Bo Chen, Lingxi Zhang, Xirui Ke, and Haipeng Ding. 2021. Neural, symbolic and neural-symbolic reasoning on knowledge graphs. *AI Open*, 2:14–35.

Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019. Iteratively learning embeddings and rules for knowledge graph reasoning. In *Proceedings of the World Wide Web Conference*, pages 2366–2377.

Zhanqiu Zhang, Jie Wang, Jieping Ye, and Feng Wu. 2022. Rethinking graph convolutional networks in knowledge graph completion. In *Proceedings of the ACM Web Conference 2022*, pages 798–807.

A Appendix

A.1 Experiments

A.1.1 Implementation Details.

We train our model using the Adam optimizer and tune the model hyperparameters via grid search on the validation set. Specifically, the hyperparameter ranges are as follows: the learning rate is chosen between 0.0001 and 0.01, and the embedding size is chosen between 100 and 500. Pytorch is used to implement the TCRA model. We conduct all experiments on the Ubuntu system with 3090 GPU.

A.1.2 Hyperparameter Experiments

This subsection analyzes the sensitivity of hyperparameters used in our model TCRA, including learning rate and embedding size. The extensive experiments of hyperparameters on FB15k-237, WN18RR, and Kinship have been coducted. The corresponding experimental results are illustrated in Figure 3.

We explore the effects of different learning rates on the performance of our TCRA model. The performance of TCRA shows a tiny fluctuation. It is seen that TCRA obtains the highest performance on FB15k-237, WN18RR, and Kinship when learning rate is 4e-4, 3e-3, and 5e-3, espectively.

In addition, the value of the embedding size is taken in {100,200,300,400,500}. It is seen that TCRA obtains the highest performance on FB15k-237, WN18RR, and Kinship when embedding size is 400, 300, and 400, espectively. It shows a trend of rising along the increasing of embedding size

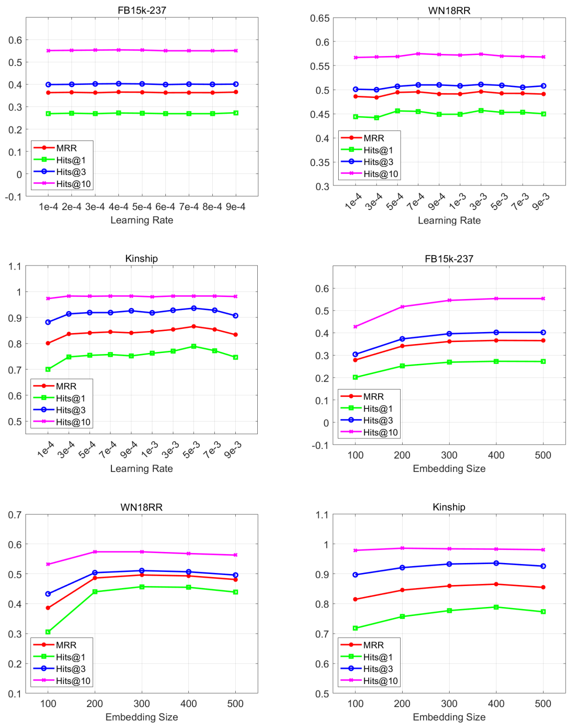

Figure 3: Hyperparameter experiments about learning rate and embedding size in Rule-Transformer.

on all three datasets FB15k-237, WN18RR, and Kinship. All in all, our model TCRA is sensitive to embedding size and not sensitive to learning rate.