IterG: An Iteratively Learning Graph Convolutional Network with Ontology Semantics

Xingya Liang[1], Fuxiang Zhang[1], Xin Liu[1], and Yajun Yang[1,2(✉)]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
{liangxingya,zhangfx,liuxin_tiei,yjyang}@tju.edu.cn
[2] Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, China

Abstract. Knowledge reasoning aims to infer new triples based on existing triples, which is essential for the development of large knowledge graphs, especially for knowledge graph completion. With the development of neural networks, Graph Convolutional Networks (GCNs) in knowledge reasoning have been paid widespread attention in recent years. However, the GCN model only considers the structural information of knowledge graphs and ignores the ontology semantic information. In this paper, we propose a novel model named IterG, which is able to incorporate ontology semantics seamlessly into the GCN model. More specifically, IterG learns the embeddings of knowledge graphs in an unsupervised manner via GCNs and extracts the semantic ontology information via rule learning. The model is capable of propagating relation layerwisely as well as combining both rich structural information in knowledge graphs and ontological semantics. The experimental results on five real-world datasets demonstrate that our method outperforms the state-of-the-art approaches, and IterG can effectively and efficiently fuse ontology semantics into GCNs.

Keywords: Graph convolutional neural networks · Knowledge reasoning · Knowledge graphs

1 Introduction

With the rapid development of artificial intelligence, Knowledge Graphs (KGs) have become a large-scale semantic network on top of the existing World Wide Web. KGs store facts as triples in the form of (*head entity*, *relation*, *tail entity*), abbreviated as (h, r, t). Entities and relations in the real world can be formally described in the form of a KG, where nodes represent entities and edges represent relations. With the development of big data and semantic web technology, a large number of KGs, such as YAGO [15,16], WordNet [9], and Freebase [1], have been developed, which have also supported a wide range of applications, including question answering [22], relation extraction [25], and recommendation systems [5].

X. Wang et al. (Eds.): APWeb-WAIM 2020, LNCS 12317, pp. 227–241, 2020.
https://doi.org/10.1007/978-3-030-60259-8_18

With the emergence of KGs, knowledge reasoning has become a basic service to support upper-level applications and attracted widespread attention. KG-oriented knowledge reasoning is intended to use various learning methods to infer the existing relations between entity pairs, and automatically identify wrong knowledge based on existing data to supplement KGs. For example, if a KG contains facts such as ($HUAWEI$, $isBasedIn$, $Shenzhen$), ($Shenzhen$, $stateLocatedIn$, $Guangdong$), and ($Guangdong$, $countryLocatedIn$, $China$), we can find the missing link ($HUAWEI$, $headquarterLocatedIn$, $China$). The target of knowledge reasoning is not only the attributes and relations between entities, but also the attribute values of entities and the conceptual level of ontologies. For instance, if an entity's ID number is known, the gender, age, and other attributes of this entity can be obtained through inference. Therefore, it is very important to efficiently and accurately realize the knowledge reasoning task on KGS.

To address these knowledge reasoning tasks, one of the solutions is to directly model the triples of KGs through a neural network, and obtain the embeddings of the elements of triples for further reasoning based on a score function. Each entire network forms a scoring function, and the output of the neural network is the scoring value. Socher et al. [14] proposed a neural tensor network named NTN, which replaced the traditional neural network layer with a bilinear tensor layer, and linked the head entity and the tail entity in different dimensions to characterize the entity complex semantic relations between them. Chen et al. [3] introduced a similar neural tensor network model to predict new relations in KGs. By initializing the entity representations learned from text using an unsupervised method, the model can be improved. Recently, Shi and Weninger [13] proposed a shared variable neural network model named ProjE. The main method of ProjE is to treat the entity prediction expectation as a multi-candidate ranking problem, and take the candidate with the highest ranking as the entity prediction result.

With the development of neural networks, GCNs in knowledge reasoning has been paid widespread attention in recent years, and it can perform convolution on arbitrary structural graphs [4,8]. However, GCNs are suitable for processing undirected graphs, and the relations in KGs are directed. Therefore, in order to apply GCNs to knowledge reasoning, Schlichtkrull et al. [12] proposed the Relation Graph Convolutional Networks (R-GCNs) model to solve the problem of knowledge reasoning from a structural perspective. The R-GCN model introduces GCNs into knowledge reasoning for the first time from a graph perspective, and has achieved outstanding results on some datasets on link prediction and entity classification. However, in this method, the evolutionary design based on GCNs is not well described, and for datasets with fewer types of relations, the quality of the processing results will be reduced. Therefore, the R-GCN model is not mature enough compared with other inference models, and there is still abundant room for improvement.

To this end, we propose a novel model named **Iter**atively learning **G**raph convolutional network with ontology semantics **(IterG)** for knowledge reasoning,

which learns the embeddings of KGs in an unsupervised manner via GCNs. In particular, the semantic ontology information in KGs is extracted via rule learning. The model is capable of propagating relations layer-wisely as well as combining both rich structural information in KGs and the semantic ontology information. We evaluate our proposed methods with the link prediction task and verify the running performance on public benchmark datasets, i.e., WN18 and FB15K. Experimental results show that our approach achieves better performance compared with the state-of-the-art approaches.

The major contributions of our work are three-fold:

1. We propose an iteratively learning graph convolutional network model **IterG**, which is regarded as a framework to complement the KGs. It can effectively accomplish the knowledge reasoning problem on KGs.
2. To enhance the reasoning ability of the model, we extract the semantic ontology information in KGs via rule learning and integrating semantics into IterG. The model is capable of propagating relations layer-wisely as well as combining both rich structural information in KGs with ontological semantics.
3. The experimental results on five benchmarks demonstrate that our proposed IterG outperforms the current state-of-the-art methods, including both traditional and deep learning based methods. And IterG can effectively and efficiently fuse ontology semantics into GCNs.

The rest of this paper is organized as follows. Section 2 reviews related work. In Sect. 3, the preliminaries of GCNs are introduced. In Sect. 4, we provide the details of the proposed algorithm for learning the embeddings of the entities and relations in KGs. Section 5 shows the experimental results, and we conclude in Sect. 6.

2 Related Work

In this paper, we focus on iteratively learning graph convolutional network and integrating ontology semantics from KGs. Thus the related work includes two parts: knowledge reasoning based on GCNs and rule learning.

2.1 R-GCN Models

The entities in KGs are connected to each other with relations. Each entity and its neighboring entities form a star structure. In a star structure, there is a relation from a central entity to an adjacent entity, and vice versa. In order to be able to learn KGs from the perspective of neighboring entities and apply them to KG completion, Schlichtkrull et al. [12] introduced R-GCNs from the perspective of graphs, which modeled the KGs via encoding the star structure from the micro level. Unlike knowledge reasoning from the perspective of text processing, R-GCN considers the problem of knowledge reasoning from the perspective of structure, which evolved from GCNs. Since GCNs only deal with undirected graphs and KGs are mostly directed graphs, R-GCNs are designed to be adapted

to directed relations. The R-GCN model can be viewed as a set of autoencoders, including an encoder and a decoder.

However, the experimental results of the R-GCN model are not stable enough: the improvements obtained on the AIFB and AM standard datasets are significant, whereas the experimental results on the MUTAG and BGS datasets are not good, which is caused by the nature of the datasets. MUTAG, a molecular map data set, is relatively simple from the perspective of both representing atomic bonds and the existence of a certain characteristic. BGS is a rock type data set with hierarchical feature descriptions released by the British Geological Survey, where the relations only indicate the existence of a specific feature or feature hierarchy. To address this issue, an improvement is to introduce the attention mechanism and replace the normalization constants with attention weights [12].

The R-GCN model introduced GCNs to knowledge reasoning for the first time from the perspective of graphs. It has achieved good results on some datasets on link prediction and entity classification. However, in this method, the evolutionary design based on GCNs is not well described, and for datasets with fewer types of relations, the quality of the processing results will be reduced. In addition, the experimental results of R-GCNs lacks comparison with the latest baselines, and the reliability remains to be verified.

2.2 Rule Learning

The rule-based reasoning methods are well-studied in traditional knowledge engineering for decades, which use logical rules for reasoning on KGs. The reasoning component inside the NELL KG uses first-order relational learning algorithms for reasoning [2]. The reasoning component learns the probabilistic rules, and after manual screening and filtering, it brings in specific entities to instantiate the rules and infer new relationship instances from other relationship instances that have been learned. The YAGO KG uses an inference machine named Spass-YAGO to enrich KG content [17]. Spass-YAGO abstracts the triples in YAGO to equivalent rule classes and uses chain superposition to calculate the transitivity of the relationship. The superposition process can be iterated arbitrarily, and the expansion of YAGO is completed by using these rules. Wang et al. [20,21] proposed a first-order probabilistic language model ProPPR (programming with personalized PageRank) for knowledge reasoning on KGs. Paulheim and Bizer [11] proposed two algorithms, SDType and SDValidate which use the statistical distribution of attributes and types to complete triples, and to identify false triples. SDType infers the types of entities by statistically distributing the types of head and tail entities, similar to the weighted voting mechanism, which assigns weight to the voting of each attribute. SDValidate first calculates the frequency of the relation-tail entity, and the low-frequency triples are further calculated by the statistical distribution of attributes and types. The triples with scores less than the threshold are considered to be potentially wrong. Jang et al. [7] proposed a pattern-based method to evaluate the quality of KG triples. This method directly analyzes the data pattern in KGs. According to the assumption that more frequent patterns are more reliable, the patterns with high occurrence

rates are selected, including the head entity patterns and the tail entity patterns, etc., and then these patterns are used for triples quality analysis.

Unlike the above previous works, we focus on the GCNs with ontology semantics, and propose an iteratively learning graph convolutional network model called IterG for the knowledge reasoning on large-scale KGs. The model is capable of propagating relations layer-wisely as well as combining both rich structural information in KGs and the ontology semantic information. To the best of our knowledge, the IterG is the first work to integrate ontology semantic information into GCNs for knowledge reasoning.

3 Preliminaries

The notations used throughout this paper are defined first. A KG $G = \{E, R, T\}$ contains a set of entities E, a set of relations R, and a set of triples $T = \{(h, r, t) \mid h, t \in E; r \in R\}$. Given a triple (h, r, t), the symbols h, r, and t denote head entity, relation, and tail entity, respectively. For instance, a triple is $(\texttt{Tianjin}, isLocatedIn, \texttt{China})$, which means that Tianjin is located in China.

3.1 Graph Convolutional Networks

Graph Convolutional Neural Networks (GCNNs) generalize traditional convolutional neural networks to the graph domain. There are mainly two types of GCNNs: spatial GCNNs and spectral GCNNs. Spatial GCNNs view the convolution as "patch operator" which constructs a new feature vector for each node using its neighborhood information. Spectral GCNNs define the convolution by decomposing a graph signal $\mathbf{s} \in R^n$ (a scalar for each vertex) on the spectral domain and then applying a spectral filter g_θ (a function of eigenvalues of L_{sym}) on the spectral components. However, this model requires explicitly computing the Laplacian eigenvectors, which is impractical for real large graphs. A way to avoid this problem is approximating the spectral filter g_θ with Chebyshev polynomials up to k-th order. Defferrard et al. [4] applied this technique to build a k-localized ChebNet, where the convolution is defined as:

$$g_\theta \star \mathbf{s} \approx \sum_{k=0}^{K} \theta'_k T_k(L_{\text{sym}})\mathbf{s}, \tag{1}$$

where $\mathbf{s} \in R^n$ is the signal on the graph, g_θ is the spectral filter, $\star$ denotes the convolution operator, T_k is the Chebyshev polynomials, and $\theta' \in R^K$ is a vector of Chebyshev coefficients. By the approximation, the ChebNet is actually spectrum-free.

However, some disadvantages exist in the first-generation parameter method, for example, the convolution kernel does not have spatial localization. In [8], Kipf and Welling simplified GCNNs by limiting $K = 1$ and approximating the largest eigenvalue λ_{max} of L_{sym} by 2. In this way, the convolution becomes

$$g_\theta \star \mathbf{s} = \theta \left(I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) \mathbf{s}, \tag{2}$$

where θ denotes the only Chebyshev coefficient left. The advantages of the convolution kernel designed by Eq. (2) are: (1) the convolution kernel has only one parameter, so the complexity of the parameters is greatly reduced; (2) convolution kernel has good spatial localization. We focus on this simplified GCN model in the rest of this paper.

3.2 OWL Web Ontology Language Axioms

In this paper, we mainly study how to integrate ontology semantic information into GCNs. Axioms are the main components of KG ontologies, since they are important to enrich semantics in KGs.

OWL (Web Ontology Language) is a semantic web ontology language with formally defined meaning and is designed to represent rich and complex knowledge about entities and relations. OWL defines multiple types of axioms, which can be used for rule reasoning. Our model is inspired by the IterE [24] model, which proposes seven object attribute expression axioms selected from the OWL ontology language. Essentially, for each type of axioms, we can draw rule conclusions through the embeddings of relations based on the linear mapping hypothesis. For instance, considering axiom SymmetricOP($hasFriend$), if a KG contains the triple ($Alice, hasFriend, Bob$), according to the rule conclusion of symmetric axiom in Table 1, a new triple (Bob, $hasFriend$, $Alice$) can be inferred. So the axioms that the relations satisfy can be obtained by calculating the similarity between the embeddings of relations and the rule conclusions. In general, the higher the similarity, the more likely the relations is to satisfy the corresponding axioms. The details of the conclusions of each axiom are listed in Table 1, where the rule form $(x, r, x)^1$ of ReflexiveOP(OPE) means reflexive.

Table 1. Seven types of axioms and translated rule formulation.

Object property axiom	Rule form
`ReflexiveOP(OPE)`	$(x, r, x)^1$
`SymmetricOP(OPE)`	$(y, r, x) \leftarrow (x, r, y)$
`TransitiveOP(OPE)`	$(x, r, z) \leftarrow (x, r, y), (y, r, z)$
`EquivalentOP(OPE`$_1$ ... `OPE`$_n$`)`	$(x, r_2, y) \leftarrow (x, r_1, y)$
`SubOP(OPE`$_1$ `OPE`$_2$`)`	$(x, r_2, y) \leftarrow (x, r_1, y)$
`InverseOP(OPE`$_1$ `OPE`$_2$`)`	$(x, r_1, y) \leftarrow (y, r_2, x)$
`SubOP(OPChain(OPE`$_1$ ... `OPE`$_n$`) OPE)`	$(y_0, r, y_2) \leftarrow (y_0, r_1, y_1), (y_1, r_2, y_2)$

4 The IterG Model

In this section, we describe our proposed IterG model in detail. Given a KG $G = \{E, R, T\}$, our objective is to learn structural and ontological information

at the same time and complement each other's advantages, while graph convolutional networks only learn the structural characteristics of nodes without considering the semantic information on the KGs. So we seamlessly integrate ontology semantic information into GCNs via the IterG model.

4.1 Intuition

We first introduce the overall architecture of IterG before reporting the detailed implementation of model's design principle that conforms to the template method design pattern, which is shown in Fig. 1 and includes two main parts: (i) auto-encoder layer and (ii) reasoning layer.

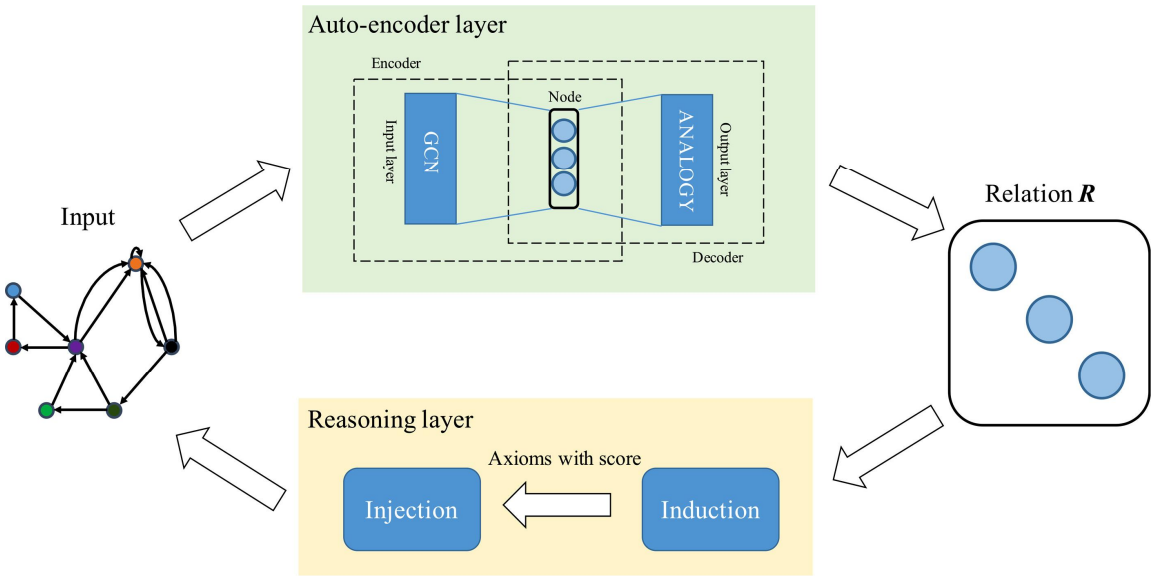

Fig. 1. The IterG architecture.

Auto-encoder layer extracts the structural information from KGs through a two-layer graph convolutional neural network, thereby obtaining the embeddings of nodes and relations.

Reasoning layer uses the embeddings of relations to conduct axiom induction, and then uses the axiom injection to select triples with high confidence and adds them to the original KG for the next iterative learning.

4.2 Graph Auto-Encoder Model

In order to obtain the embeddings of nodes and relations in KGs, we introduce a graph auto-encoder model comprised of an entity encoder and a scoring function (decoder). In this paper, GCNs are used as the encoder and ANALOGY as the decoder. First, we use a two-layer graph convolutional neural network to obtain the embeddings of the nodes in KGs, and then use ANALOGY to get the embeddings of the relations, which is shown in Fig. 2.

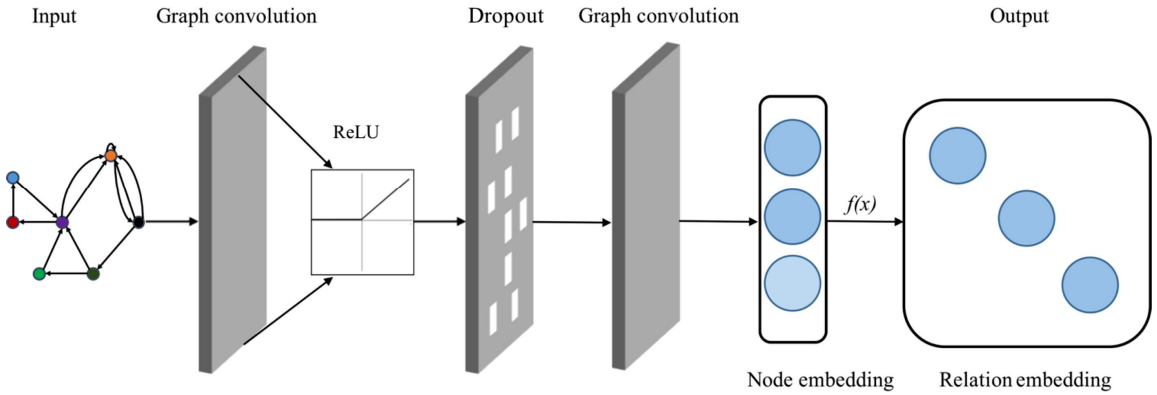

Fig. 2. Graph auto-encoder model.

As the encoder, the R-GCN model maps each entity $v_i \in \mathcal{V}$ to a real-valued vector $e_i \in \mathbb{R}^d$. The R-GCN model use the following propagation rule expressed in the message-passing architecture [6] that aggregates information from a node's local neighbors and forwards the aggregated information to the next layer,

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in R}\sum_{j\in N_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right), \tag{3}$$

where N_i^r denotes the neighbors of node i under the relation $r \in R$, $c_{i,r}$ is a normalization constant, δ denotes an activation function, and $h_i^{(l)}$ is the hidden state of i-th node at the relation $r \in R$. This layer-wisely propagation model can be implemented in sparse-dense matrix multiplications and has a computational complexity linear to the number of edges.

ANALOGY is used as the decoder in the experiments and ANALOGY performs well on the standard link prediction task. In ANALOGY, every relation r is associated with a diagonal matrix $M_r \in \mathbb{R}^{d\times d}$ and a triple (h, r, t) is scored as

$$f(h, r, t) = e_h^T R_r e_t\,. \tag{4}$$

The main innovation of ANALOGY is to apply analogy inference to the KG embedding, which adds constraints to the model's score function to capture the information of the analogy structure in KGs, thereby optimizing the embedding representation of entities and relations in KGs.

4.3 Rule Learning

After graph auto-encoder learning, the learning entity is represented by real-valued vectors and the relation is represented by matrices, which are used for rule learning. In order to learn new rules via relation matrices, we introduce the IterE [24] model, which employs seven object attribute expression axioms selected from the OWL ontology language. IterE is proposed based on the basis of embeddings learned with linear map assumption. Essentially, for each

type of axioms, we can draw rule conclusions through the embeddings of relations based on the linear mapping hypothesis. For instance, considering axiom SymmetricOP($hasFriend$), if a KG contains the triple ($Alice, hasFriend, Bob$), according to the rule conclusion of symmetric axiom in Table 1, a new triple (Bob, $hasFriend$, $Alice$) can be inferred. So the axioms that the relations satisfy can be obtained by calculating the similarity between the relation embeddings and the rule conclusions. In general, the higher the similarity, the more likely the relations is to satisfy the corresponding axioms. The rule conclusions of the OWL axioms are listed in Table 2.

Table 2. OWL axioms and rule conclusion.

Object property axioms	Rule conclusion
$\texttt{ReflexiveOP}(r)$	$\mathbf{M}_r = \mathbf{I}$
$\texttt{SymmetricOP}(r)$	$\mathbf{M}_r\mathbf{M}_r = \mathbf{I}$
$\texttt{TransitiveOP}(r)$	$\mathbf{M}_r\mathbf{M}_r = \mathbf{M}_r$
$\texttt{EquivalentOP}(r_1, r_2)$	$\mathbf{M}_{r_1} = \mathbf{M}_{r_2}$
$\texttt{subOP}(\texttt{r}_1, \texttt{r}_2)$	$\mathbf{M}_{r_1} = \mathbf{M}_{r_2}$
$\texttt{inverseOP}(\texttt{r}_1, \texttt{r}_2)$	$\mathbf{M}_{r_1}\mathbf{M}_{r_2} = \mathbf{I}$
$\texttt{subOP}(\texttt{OPChain}(r_1, r_2), r)$	$\mathbf{M}_{r_1}\mathbf{M}_{r_2} = \mathbf{M}_r$

Axiom Induction. After we calculated the relation embeddings using the graph auto-encoder model, relation embeddings are used to induce a set of axioms, denoted as A. To this end, IterG employs an effective pruning strategy to generate a possible axiom pool P which contains all possible axioms. Then we calculate the similarity between the relation embeddings and the rule conclusions to predict the score for each axiom $p \in P$.

Before calculating axiom scores with relation embeddings, concrete relations are applied to replace relation variables r, r_1, r_2, and r_3 in Table 2. As long as more than one axioms is satisfied, this axiom will be added to the pool, and a pool of possible axioms P is produced. In general, there are two methods for generating axiom pool. One way is to find possible axioms by traversing all the relations, however, the complexity of this method is too high. Another method is to generate axiom pool using random walks on KGs, but it is cannot be ensured that all possible axioms are covered. Therefore, we adopt a pruning strategy that combines traversal and random selection, which achieves a good balance between complexity and the coverage of possible axioms.

After getting relation embeddings and axiom pool P, a score s_p for each axiom $p \in P$ can be calculated based on the rule conclusions for each type of axioms. $\mathbf{M}_1^p$ and $\mathbf{M}_2^p$ denote the relation embeddings and the rule conclusions, respectively, which may be a single matrix or the dot product of two matrices. Generally, the values of $\mathbf{M}_1^p$ and $\mathbf{M}_2^p$ will be quite similar but not equal during the calculating process. So we calculate the similarity between $\mathbf{M}_1^p$ and $\mathbf{M}_2^p$ and when the higher the similarity, the more confident the axiom p will be.

Axiom Injection. IterG can infer a set of new triplets T_{new} through axiom injection with a KG G and a possible axiom set A, which employs axiom injection to infer new triples. The process of reasoning can be summarized in the following form:

$$(h^p, r^p, t^p) \leftarrow (h_1, r_1, t_1), (h_2, r_2, t_2), ..., (h_n, r_n, t_n) \tag{5}$$

where the right side triples $(h_k, r_k, t_k) \in T$ with $k \in [1, n]$ are generated from the rule conclusions of axioms, and $(h^p, r^p, t^p) \notin T$ is a new inferred triple which will be added into KGs.

A new set of triples $T_{new} = \{(h^p, r^p, t^p) \mid h^p \in E \text{ or } t^p \in E\}$ can be obtained via high-confidence axioms after axiom injection. Thus, the previous KG is updated. Then the process goes back to the graph auto-encoder model to start a new learning iteration.

5 Experimental Evaluation

In this section, our method is evaluated on the link prediction task. Extensive experiments are conducted to verify the validity of our IterG model on both benchmark and real-world KGs.

5.1 Datasets

In this research, we evaluate our models on benchmarks WN18, FB15k, and FB15k-237. Link prediction tasks are usually performed on FB15k and WN18, which are subsets of relational database Freebase and WordNet, respectively. WordNet is a semantic vocabulary KG, which has been widely used in the field of natural language processing. Freebase is a well-known knowledge base containing general facts. We also select FB15k-237 as the experimental data set, which removed all inverse triple pairs, as in [18] Toutanova and Chen found that both FB15k and WN18 have serious flaws. A simple baseline LinkFeat using a linear classifier on the sparse feature vector of the observed training relationship can greatly outperform other methods [18]. Table 3 gives a summary of these datasets.

Table 3. Datasets used in the experiments.

Dataset	WN18	FB15k	FB15k-237
Entities	40,943	14,951	14,541
Relations	18	1,345	237
Train edges	141,442	483,142	272,115
Val. edges	5,000	50,000	17,535
Test edges	5,000	59,071	20,466

In addition to the above three datasets, we also use two sparse datasets WN18-sparse and FB15k-sparse, which contain only sparse entities. It is explored whether IterG really contributes to sparse entity embeddings on sparse datasets. Table 4 gives a summary of these datasets.

Table 4. Sparse datasets used in the experiments.

Dataset	WN18-sparse	FB15k-sparse
Entities	40,943	14,951
Relations	18	1,345
Train edges	141,442	483,142
Val. edges	3,624	18,544
Test edges	3,590	22,013

5.2 Baselines

DisMult [23] is selected as the first baseline, which is a common baseline for link prediction experiment, and it can perform well on standard data sets such as FB15k. However, DisMult cannot model antisymmetric and inverse modes due to the symmetric nature of the model. We add LinkFeat proposed in [18] as a second baseline, which is a simple neighbor-based LinkFeat algorithm.

We further compare IterG to ComplEx [19], HolE [10], and R-GCN [12], which are the state-of-the-art models for link prediction. ComplEx solves the problem of DisMult and can infer symmetric and antisymmetric modes in the complex space. In addition, it can also derive inverse rules because of the existence of conjugate complex numbers. HolE is simlar to ComplE, however, HolE replaces the vector-matrix product with circular correlation. Finally, we also compare with IterE on sparse datasets.

5.3 Experimental Settings

The experimental settings are mainly divided into two parts, including graph auto-encoder model and rule learning. We first introduce the experiment settings of the graph auto-encoder model. For FB15k and WN18, a basic decomposition, with a single encoding layer and two basic functions, is employed to obtain the results. For FB15k-237, when the block dimension is 5×5 and embedding dimensional equals 500, the block decomposition performs the best. Before normalization, encoder is regularized via edge dropout. The dropout rate of the self-loops is equal to 0.2, and the dropout rate of the other edges is equal to 0.4. And $l2$ regularization is applied to the decoder with a parameter of 0.01. Adam optimizer is used in the graph auto-encoder model with a learning rate of 0.01. Finally, our model and baselines are trained with full-batch optimization.

Then we introduce the experimental settings of the rule learning part. In the part of axiom induction, the minimum axiom probability p is set to 0.5 and

the inclusion probability t is set to 0.95. For axiom injection, in order to choose axioms with high confidence as much as possible and introduce as little noise as possible, a threshold θ is set for each dataset and axioms with scores $s_{axiom} > \theta$ are regarded as high quality axioms.

5.4 Results

Two commonly evaluation metrics are employed to provide results: Mean Reciprocal Rank (MRR) and Hits@n, which can be calculated in the raw and the filtered setting [12]. The experimental results show both filtered and raw MRR, and filtered Hits@1, Hits@3, and Hits@10.

Table 5. Results on the Freebase and WordNet datasets.

Model	FB15k					WN18				
	MRR		Hits @			MRR		Hits @		
	Raw	Filtered	1	3	10	Raw	Filtered	1	3	10
LinkFeat		0.779			0.804		0.938			0.939
DistMult	0.248	0.634	0.522	0.718	0.814	0.526	0.813	0.701	0.921	0.943
R-GCN	**0.251**	0.651	0.541	0.736	0.825	0.553	0.814	0.686	0.928	**0.955**
HolE	0.232	0.524	0.402	0.613	0.739	**0.616**	0.938	0.930	0.945	0.949
ComplEx	0.242	**0.692**	0.599	0.759	0.840	0.587	0.941	**0.936**	0.945	0.947
IterG	0.245	0.684	**0.603**	**0.765**	**0.853**	0.592	**0.943**	0.933	**0.947**	0.951

Table 6. Results on FB15k-237.

Model	MRR		Hits @		
	Raw	Filtered	1	3	10
LinkFeat		0.063			0.079
DistMult	0.100	0.191	0.106	0.207	0.376
R-GCN	**0.158**	0.248	**0.153**	0.258	0.414
IterG	0.153	**0.253**	0.148	**0.267**	**0.421**
CP	0.080	0.182	0.101	0.197	0.357
TransE	0.144	0.233	0.147	0.263	0.398
HolE	0.124	0.222	0.133	0.253	0.391
ComplEx	0.109	0.201	0.112	0.213	0.388

Table 5 demonstrates the experimental results of the IterG model and other models on FB15k and WN18. On the FB15k and WN18 datasets, IterG outperforms the DistMult, but is not as good as LinkFeat like all other systems on

these two dataset. Compared with R-GCNs, the experimental results of IterG are also improved, which exactly demonstrates that the semantic information in KGs is effective for knowledge reasoning, and the ontology semantic information can improve the performance of GCNs.

Table 7. Results on the sparse datasets.

Model	FB15k-sparse					WN18-sparse				
	MRR		Hits @			MRR		Hits @		
	Raw	Filtered	1	3	10	Raw	Filtered	1	3	10
TransE	0.335	0.418	0.102	0.711	0.847	0.255	0.398	0.258	0.486	0.645
DistMult	0.558	0.738	0.593	0.875	0.931	0.324	0.600	**0.618**	0.651	0.759
ComplEx	0.677	0.911	0.890	0.933	0.944	0.327	0.616	0.540	0.657	0.761
ANALOGY	0.675	**0.913**	0.890	**0.934**	**0.944**	0.331	0.620	0.543	0.661	0.763
R-GCN	0.673	0.907	**0.894**	0.933	**0.944**	0.328	0.613	0.537	0.659	0.763
ITerE	0.675	0.901	0.870	0.931	**0.948**	0.359	0.613	0.529	0.662	0.767
IterG	**0.682**	0.908	0.885	0.923	0.945	**0.365**	**0.617**	0.548	**0.667**	**0.768**

The results of the IterG model and other models on FB15k-237 are demonstrated in Table 6. It can be inferred from the results that our IterG model is much better than DistMult, highlighting the importance of a separate encoder model. And in FB15k-237, the performance of the LinkFeat is worse than other models since inverse relations have been deleted. As aforementioned, the performance of IterG and R-GCN on FB15k-237 is similar. The IterG model is further compared with other models and it also exhibits superior performance. The above results indicate that the ontology semantic information can effectively enhance the reasoning ability of IterG.

In Table 7, we evaluate the IterG model and other models on sparse datasets. First, the link prediction results of IterG perform better than ANALOGY, which means most of the triplets injected into GCNs learning are useful. And the link prediction results also show that learning axioms from graph auto-encoder model works well. Second, IterG outperforms IterE on WN18-sparse and FB15k-sparse, which shows that GCNs can better extract the structural information in KGs, thereby generating more accurate relation embeddings. The results show that, even in the sparse datasets, our IterG model demonstrates its superiority over other models.

Based on the experimental results, we can conclude that: (1) the results in link prediction demonstrate that IterG outperforms all baselines, which indicates that IterG is capable of fusing ontology semantic information into GCNs, and (2) ontology semantic information in the IterG model can significantly improve the knowledge reasoning ability of GCNs.

6 Conclusion

In this paper, we propose an novel GCN framework, named IterG, for knowledge reasoning. In IterG, the structural and ontology semantic information on KGs are extracted at the same time, and the rule learning and GCNs are seamlessly fused to better accomplish the knowledge reasoning task. In particular, to enhance the reasoning ability of the model, we extract the ontology semantic information in KGs via rule learning. The model is capable of propagating relations layer-wisely as well as combining both rich structural information in KGs with the ontology semantic information. The evaluation on five real-world datasets demonstrates that our method outperforms the state-of-the-art approaches, and IterG can effectively and efficiently fuse ontology semantics into GCNs.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (61972275).

References

1. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pp. 1247–1250 (2008)
2. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, 11–15 July 2010 (2010)
3. Chen, D., Socher, R., Manning, C.D., Ng, A.Y.: Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. arXiv preprint arXiv:1301.3618 (2013)
4. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, pp. 3844–3852 (2016)
5. Felfernig, A., et al.: Persuasive recommendation: serial position effects in knowledge-based recommender systems. In: de Kort, Y., IJsselsteijn, W., Midden, C., Eggen, B., Fogg, B.J. (eds.) PERSUASIVE 2007. LNCS, vol. 4744, pp. 283–294. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77006-0_34
6. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1263–1272. JMLR. org (2017)
7. Jang, S., Choi, J., Yi, M.: Semi-automatic quality assessment of linked data without requiring ontology (2015)
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
9. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
10. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs (2015)

11. Paulheim, H., Bizer, C.: Improving the quality of linked data using statistical distributions. Int. J. Semant. Web Inf. Syst. **10**(2), 63–86 (2014)
12. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
13. Shi, B., Weninger, T.: ProjE: embedding projection for knowledge graph completion. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
14. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems, pp. 926–934 (2013)
15. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706 (2007)
16. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a large ontology from wikipedia and wordnet. J. Web Semant. **6**(3), 203–217 (2008)
17. Suda, M., Weidenbach, C., Wischnewski, P.: On the saturation of YAGO. In: Giesl, J., Hähnle, R. (eds.) IJCAR 2010. LNCS (LNAI), vol. 6173, pp. 441–456. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14203-1_38
18. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: The 3rd Workshop on Continuous Vector Space Models and their Compositionality (2015)
19. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. International Conference on Machine Learning (ICML) (2016)
20. Wang, W.Y., Mazaitis, K., Cohen, W.W.: Programming with personalized pagerank: a locally groundable first-order probabilistic logic (2013)
21. Wang, W.Y., Mazaitis, K., Lao, N., Mitchell, T., Cohen, W.W.: Efficient inference and learning in a large knowledge base: Reasoning with extracted information using a locally groundable first-order probabilistic logic. Comput. Sci. (2014)
22. Weston, J., et al.: Towards AI-complete question answering: a set of prerequisite toy tasks. Comput. Sci. (2015)
23. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
24. Zhang, W., et al.: Iteratively learning embeddings and rules for knowledge graph reasoning. In: The World Wide Web Conference, pp. 2366–2377 (2019)
25. Zhou, G., Su, J., Zhang, J., Zhang, M.: Exploring various knowledge in relation extraction. In: ACL, Meeting of the Association for Computational Linguistics, Conference, June, University of Michigan, USA (2005)