Information Fusion 90 (2023) 253–264

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Fusing topology contexts and logical rules in language models for knowledge graph completion

Qika Lin [a], Rui Mao [b], Jun Liu [c,d], Fangzhi Xu [a], Erik Cambria [b,*]

[a] School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China
[b] School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore, 639798, Singapore
[c] Shaanxi Province Key Laboratory of Satellite and Terrestrial Network Tech. R&D, Xi'an, Shaanxi, 710049, China
[d] National Engineering Lab for Big Data Analytics, Xi'an, Shaanxi, 710049, China

ARTICLE INFO

Keywords:
Knowledge graph completion
Information fusion
Topology context
Logical rule
Language model

ABSTRACT

Knowledge graph completion (KGC) aims to infer missing facts based on the observed ones, which is significant for many downstream applications. Given the success of deep learning and pre-trained language models (LMs), some LM-based methods are proposed for the KGC task. However, most of them focus on modeling the text of fact triples and ignore the deeper semantic information (e.g., topology contexts and logical rules) that is significant for KG modeling. For such a reason, we propose a unified framework **FTL-LM** to **F**use **T**opology contexts and **L**ogical rules in **L**anguage **M**odels for KGC, which mainly contains a novel path-based method for topology contexts learning and a variational expectation–maximization (EM) algorithm for soft logical rule distilling. The former utilizes a heterogeneous random-walk to generate topology paths and further reasoning paths that can represent topology contexts implicitly and can be modeled by a LM explicitly. The strategies of mask language modeling and contrastive path learning are introduced to model these topology contexts. The latter implicitly fuses logical rules by a variational EM algorithm with two LMs. Specifically, in the E-step, the triple LM is updated under the supervision of observed triples and valid hidden triples verified by the fixed rule LM. And in the M-step, we fix the triple LM and fine-tune the rule LM to update logical rules. Experiments on three common KGC datasets demonstrate the superiority of the proposed FTL-LM, e.g., it achieves 2.1% and 3.1% Hits@10 improvement over the state-of-the-art LM-based model LP-BERT in the WN18RR and FB15k-237, respectively.

1. Introduction

Knowledge graphs (KGs) have attracted extensive attention from the Artificial Intelligence (AI) community as they store vast amounts of real-world knowledge of facts [1,2]. Each fact is normally represented as a triple (h, r, t), where h, r and t denote a head entity, a relation, and a tail entity, respectively, e.g., *(Yao Ming, marriedTo, Ye Li)*. With high-quality KGs, many AI applications could achieve excellent accuracy and explainable reasoning processes [3–5], such as question answering [6,7], language modeling [8,9], semantic reasoning [10,11], recommendation systems [12,13], sentiment analysis [14,15], and medical intelligence [16]. However, the knowledge of most KGs is incomplete, due to limited annotation resources and technologies, while an incomplete KG cannot meet the information needs of divers downstream applications [17]. For such a reason, the task of knowledge graph completion (KGC) is proposed to infer new valid hidden triples, based on the observed ones in a KG.

Some KG embedding methods, such as TransE [18] and DistMult [19], are proposed to embed entities and relations into a continuous vector space. Then, they calculate the scores of triples to complete reasoning. Meanwhile, to model the topology of KGs, some graph convolution networks (GCNs) are proposed to fuse the neighbor information of entities, such as R-GCN [20] and CompGCN [21]. Despite the great success of these two types of methods, they only utilize either individual triples or neighbor information, while the intrinsic semantics of entities and relations is ignored by their algorithms. For example, none of these methods take into account the actual semantics of the entity *Yao Ming* in Fig. 1, i.e., he was an NBA player born in China, which leads to insufficient information modeling and reasoning performance. In light of this, some language model (LM) based KGC methods are proposed [22–26], among which KG-BERT [22] and StAR [25] are representative studies.

* Corresponding author.
E-mail addresses: qika@sentic.net (Q. Lin), rui.mao@ntu.edu.sg (R. Mao), liukeen@xjtu.edu.cn (J. Liu), Leo981106@stu.xjtu.edu.cn (F. Xu), cambria@ntu.edu.sg (E. Cambria).

https://doi.org/10.1016/j.inffus.2022.09.020
Received 20 May 2022; Received in revised form 23 September 2022; Accepted 25 September 2022
Available online 30 September 2022

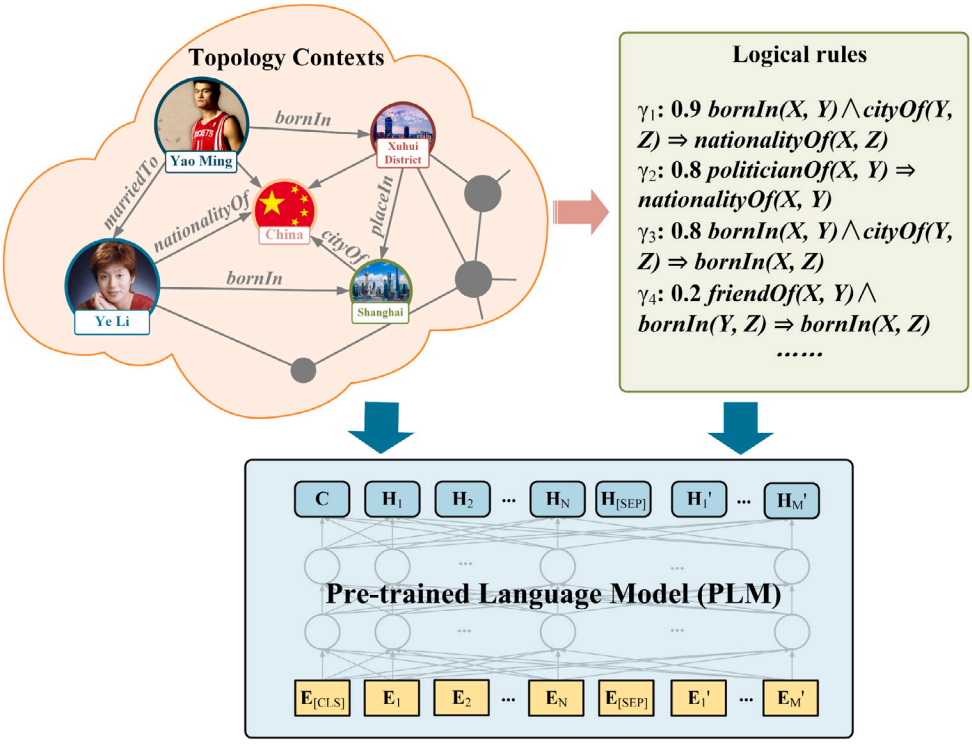

Fig. 1. An example of fusing topology contexts and logical rules in language models.

They input the textual descriptions of entities and the relation of a triple into LMs to calculate a score that is to model the plausibility of this triple. They only model the local information of triples, whereas the long-distance information association in KGs is ignored, e.g., topology contexts and logical rules (see examples in Fig. 1). Both of the topology contexts and logical rules play significant roles for KGC [27], because the former focuses on the entity topological features of the graph; the latter lays emphasis on the causal associations between relations.

However, fusing topology contexts and logical rules in LMs is challenging. Firstly, topology contexts in KGs are represented in the form of a graph structure, which is obviously different from normal word sequences that can be processed by LMs. Thus, one cannot use LMs to directly model topology contexts. Secondly, logical rules of KGs are composed of rule confidence and atomic formulas, containing relations and variables, and representing abstract meanings, e.g., rule γ_1 in Fig. 1: $0.9\ bornIn(X, Y) \wedge cityOf(Y, Z) \Rightarrow nationalityOf(X, Z)$. It is intractable to fuse logical rules into LMs. On one hand, the number of predicates of logical rules in KGs is very limited, and the semantics of rules is expressed through their permutation. This is significantly different from the natural language. Thus, LMs cannot directly model the logical rules as well. On the other hand, there are no labeled logic rules with semantic confidence as the supervision information of LMs.

To address the above issues, we propose a novel two-stage framework to implicitly **F**use **T**opology contexts and **L**ogical rules in **L**anguage **M**odels for KGC, termed **FTL-LM**. Specifically, for the learning of topology contexts, we firstly propose a heterogeneous random-walk algorithm to generate diverse topology paths that comprehensively considers various factors in KGs, including breadth-first sampling, depth-first sampling and different relations. By omitting intermediate entities, these topology paths are transformed into reasoning paths, and their positive and negative instances are then sampled. Afterwards, the strategies of mask language modeling and contrastive path learning are utilized to model the semantic information of both topology paths and reasoning paths. In addition, soft logical rule distilling is introduced to fuse logical rules in LMs, where two LMs with the same architecture but different parameters are utilized for triplet modeling and rule modeling, respectively. A variational expectation–maximization (EM) algorithm is proposed to iteratively optimize these two LMs. In the E-step, the triple LM is updated under the supervision of observed triples and valid hidden triples, verified by the fixed rule LM. In the M-step, we fix the triple LM and fine-tune the rule LM to update logical rules. Through the above strategies, both the topology contexts and logical rules of KGs are implicitly fused in LMs. Our main contributions are summarized as follows:

- A unified framework FTL-LM that fuses both topology contexts and logical rules of KGs in LMs is proposed. To our best knowledge, this is the first study that simultaneously integrates these two types of information.
- A novel path-based method for the learning of topology contexts is proposed, where we first generate topology paths with a heterogeneous random-walk algorithm, and then construct reasoning paths and their positive and negative samples. Afterwards, mask language modeling and contrastive path learning are utilized to model the semantics of these topology contexts.
- Because of the intractability of directly fusing logical rules, a variational EM algorithm is introduced to alternatively optimize two LMs for triple modeling and rule modeling, respectively. By using this soft distillation, logical rules of KGs are incorporated into the LM for the first time.
- Experiments on two common KGC datasets demonstrate the superiority of our method, which shows that our FTL-LM surpasses all current LM-based methods. Furthermore, the ablation study demonstrates the effectiveness of each module in the FTL-LM framework.

The rest of this paper is organized as follows: Section 2 introduces the related work of the KGC task. The preliminary is given in Section 3. Our proposed method FTL-LM is detailed in Section 4. Section 5 carries out extensive experiments and analysis results on two commonly used KGs. Finally, we give the conclusion and discuss the future work in Section 6.

2. Related work

Currently, many studies are carried out for the KGC task, which can be mainly divided into four categories: fact-embedding, topology-embedding, rule-enhanced and LM-based methods.

2.1. Fact-embedding methods

Fact-embedding methods only consider each fact triple with the form of (h,r,t) in a KG as the basis. They mainly optimize randomly initialized entity and relation representations through the preset strategy for triple score calculation, among which translational distance models and semantic matching models are the two most representative types [27]. Translational distance models define the triple score by the distance between the head entity and tail entity after a specific relation translation. TransE [18] is the first and most popular model of this type. It requires that the head entity representation is close to the tail's after relation translation of addition operation in the vector space $\mathbb{R}^d$, i.e., $\mathbf{h}+\mathbf{r}\approx\mathbf{t}$. Based on TransE, related methods have been proposed and achieved great success, such as TransH [28] and RotatE [29]. The former introduces relation-specific hyperplanes to extend the relation translation and the latter converts the addition operation in real space of TransE into a rotation from head to tail entity in complex space.

Semantic matching models express the validity of triples by the matching degree of entity and relation vectors in the embedded space [27]. For example, RESCAL [30] assigns a vector $\mathbf{e}\in\mathbb{R}^d$ to each entity and a matrix $\mathbf{M}\in\mathbb{R}^{d\times d}$ to each relation to represent their intrinsic semantics. And the triple validity is defined by a bilinear function $\mathbf{e}_h^\top\mathbf{M}_r\mathbf{e}_t=\sum_{i=0}^{d-1}\sum_{j=0}^{d-1}[\mathbf{e}_h]_i[\mathbf{M}_r]_{i,j}[\mathbf{e}_t]_j$. DistMult [19] utilizes a semantic vector $\mathbf{r}\in\mathbb{R}^d$ to simplify $\mathbf{M}$ in RESCAL by assuming that it is a diagonal matrix, i.e., $\mathbf{M}=\mathrm{diag}(\mathbf{r})$. Based on DistMult, ComplEx [31] converts embedding vectors in real space to complex space for better asymmetric relations modeling. Although these fact-embedding methods are simple and efficient, they only consider the triple information of KGs, resulting in insufficient modeling and performance degradation.

2.2. Topology-embedding methods

To comprehensively embed the information of KGs, some topology-embedding methods are proposed, which are mainly based on GCNs. Their strategy is to iteratively aggregate information from neighbor nodes to the target node to integrate local topology structures. R-GCN [20] is the first work applying GCN to KGs, where relation specific matrices are introduced to handle the heterogeneity of edges. Meanwhile, the basis and block-diagonal decomposition are proposed to avoid over-parameterization and over-overfitting. Moreover, VR-GCN [32] and CompGCN [21] learn embeddings of both entities and relations simultaneously for multi-relational KG. A variety of entity-relation composition operations are utilized in CompGCN, through which it is very efficient and can generalize to several of multi-relational GCN methods. In general, these methods have achieved excellent performance and are widely used because they are capable of modeling topology contexts of KGs. However, the related works of this type cannot effectively deal with the intrinsic semantics of entities and relations. Besides, modeling high-level semantic associations in KGs (e.g., logical rules) is also challenging for these methods. Last but not least, since these topology-embedding models embed a fixed number of entities and relations, they can simply process static KGC whose entities and relations will not increase in the future. This significantly narrows their application scopes, because to achieve a knowledgeable and up-to-date KG, the triples continuously grow in the real world [33–35].

2.3. Rule-fused methods

Rule-fused methods mainly perform interpretable KGC by mining semantic associations between relations and local structures in the KG. Neural LP [36] is the first study of learning first-order logic rules in an end-to-end differentiable manner. It models the parameter and structure of rules by compiling reasoning tasks into sequences of differentiable operations. Based on Neural LP, DRUM [37] extends the rule learning and reasoning to a variable-length pattern by introducing an additional empty relation. In this way, it can learn richer rules and conduct more accurate reasoning. In order to learn topology contexts and logic rules of KGs simultaneously, JSSKGE [38] employs graph attention networks to aggregate the local structural information of entities. Then, it utilizes soft logical rules implicated in KGs as an expert to further rectify the embeddings. It can conduct KGC and rule learning by a joint learning method.

Due to the open-world assumption of KGs [27], the above methods of one-time rule learning cannot fully reflect the real situation. Therefore, some iterative models for rule learning and reasoning are proposed. For example, pLogicNet [39] combines the KG embedding and rule learning in a variational EM framework. In the E-step, a fact-embedding model is used for inferring missing triples, while in the M-step, the weights of logic rules are updated based on the observed and predicted triples. Similarly, RNNLogic [40] introduces a rule generator as well as a reasoning predictor for iterative optimization. In each iteration, the reasoning predictor is first updated to explore some logic rules. And then high-quality rules are selected with both the rule generator and reasoning predictor via posterior inference. Finally, the rule generator is updated under the supervision of these high-quality rules. In general, these rule-fused methods have great application potential due to its interpretable advantages, but are limited by the scalability of rule learning. Besides, they are usually difficult to effectively integrate with other features of KGs to complete accurate reasoning.

2.4. LM-based methods

To take into consideration the inherent semantics of entities and relations for better representation, some text-enhanced methods are proposed. These methods usually add the representations of text descriptions on the basis of fact-embedding methods. Then, they are optimized by the strategy of joint learning, such as TEKE [41] and DKRL [42]. In recent years, pre-trained language models, e.g., BERT [43] and RoBERTa [44], have achieved great success in natural language processing (NLP) tasks [45,46]. Naturally, they can be transferred to the KGC task. KG-BERT [22] concatenates the text descriptions of a head entity, a relation and a tail entity as the input of BERT, and takes the final `[CLS]` representation as embedded vector of the target triple. Then, it is passed into a two-way classifier to determine whether the triple is plausible or not. To enhance the representation of structured knowledge in the textual encoder, StAR [25] partitions each triple into two asymmetric parts as in a translation-based graph embedding approach, and encodes both parts into contextualized representations. Both a deterministic classifier and a spatial measurement for the learning of representations and structures are then employed. Afterwards, multi-task learning for KGC with LMs is proposed. For example, MTL-KGC [23] introduces link prediction, relation prediction and relevance ranking simultaneously. LP-BERT [26] conducts multi-task pre-training for KGC, where not only the original mask language modeling is utilized, but also mask entity modeling and mask relation modeling are introduced. All these methods have achieved excellent performance through the introduction of structure knowledge. However, the modeling structure of them is only limited to the triple level, which fails to capture the long-distance structure semantics in KGs, e.g., topology contexts and logical rules.

To overcome the defects of current LM-based methods, we propose a unified framework FTL-LM that fuses both topology contexts and logical rules of KGs in LMs. It can be viewed as an extension of the general LM-based methods that additionally incorporates the modeling information of topology-embedding methods and rule-fused methods. Different from the conventional topology-embedding methods that utilize GCNs, we propose a novel path-based method for the learning of topology contexts using LMs. Besides, for the rule fusion, a variational EM algorithm is introduced to alternatively optimize two LMs for triple modeling and rule modeling, respectively. These specific improvements allow LMs to indirectly achieve the purpose of fusing topology contexts and logical rules.

3. Preliminary

3.1. Knowledge graph and topology contexts

A knowledge graph can be formally expressed as: $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}_o\}$, where $\mathcal{E}$ and $\mathcal{R}$ denote the sets of entities and relations, respectively. $\mathcal{T}_o \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is the observed triples. As the open-world assumption (OWA) states that KGs contain only true facts and non-observed facts can be either false or just missing [27,47], so there exists valid hidden triples, we formalize as $\mathcal{T}_u$ ($\mathcal{T}_u \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E}, \mathcal{T}_o \cap \mathcal{T}_u = \emptyset$). The text descriptions of entities and relations can be viewed as $\mathcal{D}_\mathcal{E} = \{s^e_1, s^e_2, \dots, s^e_{|\mathcal{E}|}\}$ and $\mathcal{D}_\mathcal{R} = \{s^r_1, s^r_2, \dots, s^r_{|\mathcal{R}|}\}$, where $|\cdot|$ denotes the number of elements. s^e_i and s^r_i are sentences composed of different numbers of word tokens, i.e., $s_i = [w_1, w_2, \dots, w_n]$.

There are two main types of topology contexts in KGs: local and long-distance topology contexts [48]. The local topology contexts represent the most basic graph features of an entity, which can be modeled by processing triples in the form of (h, r, t). Although some fact-embedding methods achieve great performance using this kind of topology [18,29], they are not sufficient for modeling the comprehensive semantics of KGs. Long-distance topology contexts model for longer topology in KGs, such as the path in Fig. 1: (*Yao Ming, marriedTo, Ye Li, bornIn, Shanghai, placeIn, China*). It has more complex and richer semantics. For more accurate KG embeddings, both the two types of topology contexts are needed.

3.2. Logical rules in KGs

The general logical rules connect the causal relationship between a premise and a hypothesis through implication symbols, i.e., *premise* ⇒ *hypothesis*. The premise and hypothesis are all composed of atoms that are facts connecting variables or constants by a predicate. Due to the particularity of KG structures, we introduce the Horn rule [49] as it is tractable and expressive [50,51]. Each atom in Horn rules is represented as a predicate (relation in KG) connecting two variables, e.g., *bornIn(X,Y)*. Meanwhile, it limits *premise* to a list of atoms and *hypothesis* to an atom, which are called rule body and rule head respectively. Furthermore, for the convenience without the loss of generality, the closed Horn rule requires its rule body to connect transitively by shared variables, where the first and the last variable are the same as the counterpart of the rule head. To model the uncertainty, a confidence $\epsilon \in [0, 1]$ is usually introduced for each Horn rule. The length of Horn rules is the number of atoms in the rule body. An example closed Horn rule with length 2 is shown below:

$$\epsilon, bornIn(X, Y) \wedge cityOf(Y, Z) \Rightarrow nationalityOf(X, Z), \tag{1}$$

where *bornIn(X, Y)* ∧ *cityOf(Y, Z)* is a rule body and *nationalityOf(X, Z)* is a rule head. By substituting the variables with concrete entities in KGs, we can obtain a ground Horn clause corresponding to the original Horn rule. For example, a ground Horn clause of rule (1) can be *bornIn(Yao Ming, Shanghai)*∧*cityOf(Shanghai, China)*⇒*nationalityOf(Yao Ming, China)*.

Table 1
Notations used in the paper.

Symbol	Description
$\mathcal{G}$	The knowledge graph
$\mathcal{E}, \mathcal{R}, \mathcal{T}_o$	The set of entities, relations, observed triples
$\mathcal{T}_u$	Hidden triples
α, β	Weights for DFS and BFS in heterogeneous random-walk
θ	Attenuation coefficient in heterogeneous random-walk
p	A topology path
p_h, p_t	Head part and tail part of the path p
$\mathbf{p}_h, \mathbf{p}_t$	Embeddings of p_h and p_t via LMs
p_r	The reasoning path of p_h
p_r^ρ, p_r^η	The positive instance and negative instance of p_r
$\mathbf{p}_r, \mathbf{p}_r^\rho, \mathbf{p}_r^\eta$	Embeddings of p_r, p_r^ρ and p_r^η via LMs
τ	Contrastive temperature
γ, Γ	A logical rule and a rule set of the KG $\mathcal{G}$
P_w, Q_v	The distribution for observed triples and hidden triples

4. Methodology

In this section, we will introduce our proposed FTL-LM model for KGC, which mainly contains four technical components: topology context learning, soft logical rule distilling, triple embedding, and overall process and training regime. The main architecture is illustrated in Fig. 2 and the notations used in the paper are summarized in Table 1.

4.1. Topology context learning

Directly fusing long-distance topology contexts into LMs is intractable because there is a natural representation gap between the complex topology and the natural language processed by LMs. Thus, we refer to the strategy of random walks [52,53] used in many homogeneous graphs to generate paths and then use them as implicit representation of topology contexts. Specifically, suppose the current sampled path is p_i, we sample the next neighbor for path growth by the following probability:

$$Pr\big((r_j, e_j)|p_i\big) = \frac{\Phi(p_i, r_j, e_j)}{\sum_{(r_k, e_k) \in \mathcal{N}_{p_{i,-1}}} \Phi(p_i, r_k, e_k)}, \tag{2}$$

where $p_{i,-1}$ denotes the last entity of path p_i and $\mathcal{N}$ is the relation-entity pair neighbor of the target entity. As Fig. 3 shows, the current path has the last entity e_3 and its neighbors include (r_9, e_4), (r_8, e_5), (r_6, e_6) and (r_7, e_7). Φ is the function to calculate the sample probability of neighbors, which is defined as:

$$\Phi(p_i, r_j, e_j) = (\alpha + \beta) \cdot \phi(p_i, r_j). \tag{3}$$

Similar to node2vec [53], α and β denote weights for the neighbor nodes under depth-first sampling (DFS) and breadth-first sampling (BFS), respectively. Since we have limited the sampling range to neighbors, α is set to a constant value of 1. β indicates the degree to which the model pays attention to BFS. As Fig. 3 shows, the current neighbors e_4 and e_5 are connected with sampled nodes so they can also be viewed as BFS neighbors. A larger value of β means that the sampling process will pay more attention to e_4 and e_5, indicating that the model generally focuses more on the local structures of KG. $\phi(p_i, r_j)$ represents the semantic relevance between a current path p_i and the relation of a sample neighbor r_j. Formally, $\phi(p_i, r_j) = \cos(\mathbf{p}_i, \mathbf{r}_j)$ where $\mathbf{p}_i$ and $\mathbf{r}_j$ denote the embeddings of the current path and the neighbor relation, respectively. SimCSE [54] is utilized for computing $\phi(p_i, r_j)$, as it has modeled the semantic relevance between two sentences through contrastive learning. While add a new neighbor for the path, the relation semantics is updated:

$$\mathbf{p}_i = \theta \cdot \mathbf{p}_i + (1 - \theta) \cdot \mathbf{r}_j, \tag{4}$$

where $\theta \in [0, 1]$ represents the proportion for retained information of the previous sampled path.

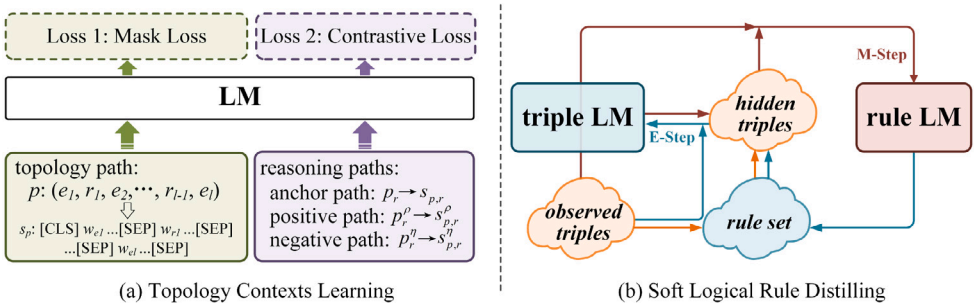

Fig. 2. The overall architecture of FTL-LM, which mainly contains two parts: topology context learning and soft logical rule distilling. The former models topology paths and reasoning paths involving topology contexts through MLM loss and contrastive loss. The latter softly distils the logical rules into the LM by a variational EM algorithm, where orange arrows indicate to preliminarily search rules and generate hidden triples. The blue and red arrows denote the process of E-step and M-step, respectively.

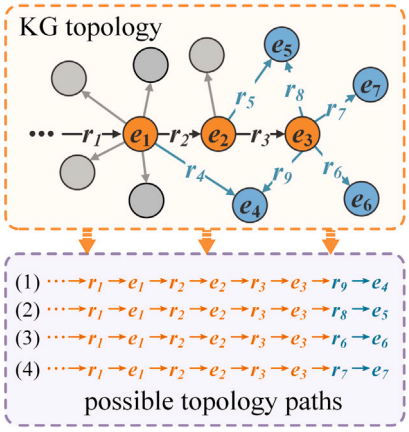

Fig. 3. An example for the topology path generation by the heterogeneous random-walk. The orange part indicates the sampled items of the path while the blue ones represent the possible samples in the current step. The gray part indicates the entities or relations that are not sampled.

Algorithm 1: The heterogeneous random-walk algorithm for topology paths generation.

Input: the knowledge graph $\mathcal{G}$, number of topology path M, min path length N_{min}, max path length N_{max}.
Output: the topology path set $\mathcal{P}$.

while $len(\mathcal{P}) < M$ **do**
  $l \leftarrow$ sample an integer in $[N_{min}, N_{max}]$;
  path $p \leftarrow$ sample a triple in $\mathcal{T}_o$;
  path embedding $\mathbf{p} \leftarrow$ embed the relation of p using SimCSE;
  while $len_{path}(p) < l$ **do**
    Get neighbors $\mathcal{N}$ for last entity p_{-1} from $\mathcal{G}$;
    if $len(\mathcal{N}) = 0$ **then**
      break;
    Calculate the weight for each pair (r, e) ; ▷ Eq. (3)
    $(r_j, e_j) \leftarrow$ sample neighbor with the weights;
    $p \leftarrow$ add (r_j, e_j) to p;
    Update the path embedding $\mathbf{p}$ by Eq. (4) ; ▷ Eq. (4)
  $\mathcal{P} \leftarrow$ add p to $\mathcal{P}$;
Return topology path set $\mathcal{P}$.

And the latter half of the equation represents the information transformation after adding a new relation and an entity. Generally, the heterogeneous random-walk algorithm for topology path generation can be summarized as Algorithm 1. In this way, we have collected sampled topology paths, which can implicitly reflect long-distance topology contexts of the KG. So LMs can learn topology contexts by modeling these paths. We carry out two tasks for the joint optimization: mask language modeling and contrastive path learning.

Mask Language Modeling. For a topology path $e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} \cdots \xrightarrow{r_l} e_{l+1}$ of length l, we replace all entities and relations with corresponding text descriptions as the input to LMs, i.e., $\left[\texttt{[CLS]}, \texttt{[e}_1\texttt{]}, \texttt{[SEP]}, \texttt{[r}_1\texttt{]}, \texttt{[SEP]}, \texttt{[e}_2\texttt{]}, \texttt{[SEP]}, \ldots, \texttt{[e}_{l+1}\texttt{]}, \texttt{[SEP]}\right]$ where [SEP] represents the split symbol. $\texttt{[e}_i\texttt{]}$ and $\texttt{[r}_j\texttt{]}$ denote word sequences of entity e_i and relation r_j. For modeling these topology paths, we utilize the strategy of word masking to fine-tune LMs on KGs. Word masking was firstly proposed in BERT [43] to learn the semantic representations. To enable LMs to model the topology context of KGs, the whole words of the relation or the first several words of the entity are selectively masked out. We select the text of an entity (except for the first and the last entities in a topology path) or a relation to mask each time until all masked words reach 15% tokens in the topology path. Similar to BERT, 80% of masked words are replaced with a [MASK] flag, 10% are substituted with a random token and the others remain unchanged.

Contrastive Path Learning. The above mask language modeling only considers existing paths of KGs. To further reflect the real or non-existent topology contexts in embeddings, we carry out the contrastive path learning to preserve the semantic information. To achieve this, first, we have limited the implementation of above mask language modeling. Each topology path $e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} \cdots \xrightarrow{r_l} e_{l+1}$ is divided into two parts: head part $e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} \cdots \xrightarrow{r_l}$, and a tail part e_{l+1}, respectively. We only mask entities or relations in the head part, excluding the first and last item, i.e., e_1 and r_l of the example.

Then, we introduce the reasoning path for the head part of topology path, which only keeps the first entity and omits other intermediate entities, i.e., $e_1 \xrightarrow{r_1} \xrightarrow{r_2} \cdots \xrightarrow{r_l}$. In fact, the information represented by the reasoning path in this form is largely equivalent to the head part of the corresponding topology path, because the later can be obtained by the iterative inference of the former in KGs. Moreover, through such a transformation, the confidence of logic rules can be effectively calculated by LMs, which is detailed in Section 4.2. By the iterative inference over a reasoning path, the last entity (the reasoning tail) can be obtained, i.e., e_{l+1}. From this point of view, we define the positive contrastive instance for the anchor reasoning path: both share the major topology contexts and have the same reasoning tail. We can realize this by a simple strategy that discards some beginning parts of the original path. In this way, the positive contrastive instance can be $e_2 \xrightarrow{r_2} \xrightarrow{r_3} \cdots \xrightarrow{r_l}$ or $e_3 \xrightarrow{r_3} \xrightarrow{r_4} \cdots \xrightarrow{r_l}$ and other similars. For negative contrastive instance, we replace one of the relations of the anchor reasoning path with a random relation that ensures the new path never appear in the KGs. For example, changing r_2 to another relation: $e_1 \xrightarrow{r_1} \xrightarrow{r_2'} \cdots \xrightarrow{r_l}$. Consequently, the original reasoning tail cannot be reached by the iterative inference in a negative contrastive instance.

Formally, a topology path p has a head part p_h and a tail part p_t, which are represented as $\mathbf{p}_h$ and $\mathbf{p}_t$ by LMs. The reasoning path p_r corresponds to p_h. p_r has a positive contrastive instance p_r^ρ and a negative counterpart p_r^η. They are embedded as $\mathbf{p}_r$, $\mathbf{p}_r^\rho$ and $\mathbf{p}_r^\eta$ respectively. For mask language modeling, the classical cross entropy loss is carried out to optimize:

$$\mathcal{L}_{MLM} = \sum_{p}\sum_{i=1}^{n}\sum_{j=1}^{m} -y_{i,j}^{p}\log \hat{y}_{i,j}^{p}, \tag{5}$$

where n is the input length and m is the number of tokenized tokens of LMs. $y_{i,j}^p$ and $\hat{y}_{i,j}^p$ denote the actual label and the predict label of the word tokens in path p, respectively. For the contrastive path learning, the infoNCE [55,56] loss is utilized:

$$\mathcal{L}_{CPL} = \sum_{p_r} -\log \frac{e^{\mathrm{sim}(\mathbf{p}_r,\mathbf{p}_r^\rho)/\tau}}{e^{\mathrm{sim}(\mathbf{p}_r,\mathbf{p}_r^\rho)/\tau} + \sum_{\mathbf{p}_r^\eta} e^{\mathrm{sim}(\mathbf{p}_r,\mathbf{p}_r^\eta)/\tau}}, \tag{6}$$

where τ is the hyperparameter of contrastive temperature. $\mathrm{sim}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|\cdot\|\mathbf{y}\|}$ is the cosine similarity of two vectors. In the batch training, all negative reasoning paths of an anchor path and other anchor ones are as collected as negative instances. For more comprehensive modeling, the contrastive score $\mathrm{sim}(\mathbf{p}_r,\mathbf{p}_r^p)$ of the positive pair in Eq. (6) is replaced as: $\frac{1}{3}[\mathrm{sim}(\mathbf{p}_r,\mathbf{p}_r^p) + \mathrm{sim}(\mathbf{p}_r,\mathbf{p}_h) + \mathrm{sim}(\mathbf{p}_r,\mathbf{p}_t)]$. In this way, the abstract reasoning path and the head part of the topology path should have a similar semantic representation. Meanwhile, the reasoning path and the tail part are also similar in embedding space, by which the KGC can be achieved indirectly, because the query of KGC is actually a reasoning path of length 1, for example, reasoning path $h \xrightarrow{r}$ corresponds to the query $(h, r, ?)$. We jointly optimize LMs for topology context learning by combining the above two losses:

$$\mathcal{L}_{TCL} = \mathcal{L}_{MLM} + \mathcal{L}_{CPL}. \tag{7}$$

4.2. Soft logical rule distilling

Because of the intractability of explicitly fusing logical rules in LMs, we introduce the Markov logic network (MLN) [57] to conduct soft logical rule distilling with the inspiration of pLogicNet [39]. For a given KG $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}_o\}$ and its hidden triples $\mathcal{T}_u$, different values of observed and hidden triples form their corresponding random variables $\mathcal{V}_o$ and $\mathcal{V}_u$. $\mathcal{V}_{(h,r,t)} = 1$ if triple (h,r,t) if true, otherwise $\mathcal{V}_{(h,r,t)} = 0$. Suppose there is a rule set Γ, for an arbitrary rule $\gamma \in \Gamma$, it may have several groundings in the KG, i.e., $g(\gamma) = \{g_\gamma^1, g_\gamma^2, \dots\}$. Each grounding g_γ^i replaces the variables of the rule with specific entities, which can be referenced to Section 3.2. The body (b) and head (h) of the grounding can be transformed into the corresponding reasoning path $p_{r,\gamma^i,b}$ and $p_{r,\gamma^i,h}$, for example, $Yao\ Min \xrightarrow{bornIn} \xrightarrow{cityOf}$ and $Yao\ Min \xrightarrow{nationalityOf}$, respectively. Then the semantic confidence of rule γ can be calculated:

$$\epsilon_\gamma = \frac{1}{|g(\gamma)|}\sum_{g_\gamma^i \in g(\gamma)} \mathbf{Norm}\big(\mathrm{sim}(\mathbf{p}_{r,\gamma^i,b}, \mathbf{p}_{r,\gamma^i,h})\big), \tag{8}$$

where **Norm** is the normalization function to convert the value range of cosine similarity [−1,1] to [0,1]. Based on learned rules, the joint probability of all observed ones can be calculated by the Markov logic network [57]:

$$P_w(\mathcal{V}_o) = \frac{1}{Z(w)}\exp\left[\sum_{\gamma\in\Gamma}\epsilon_\gamma\Big(\sum_{g_\gamma^i \in g(\gamma)}\psi(g_\gamma^i)\Big)\right], \tag{9}$$

where $Z(w)$ is the partition function summing over all possible variable groundings. $\psi(\cdot) \in \{0,1\}$ denotes the discriminant function that is used to calculate whether a rule ground is correct. Directly calculating and maximizing this joint probability is demanding, because it requires to integrate over all possible values of $\mathcal{V}_o$ to compute the partition function $Z(w)$.

Thus, we introduce the evidence lower bound of its log-likelihood:

$$\begin{aligned}\log P_w(\mathcal{V}_o) &\geq \mathcal{L}_{ELBO}(Q_v, P_w)\\ &= \log P_w(\mathcal{V}_o) - \mathbf{KL}\big[Q_v(\mathcal{V}_u) \parallel P_w(\mathcal{V}_u|\mathcal{V}_o)\big]\\ &= \mathbb{E}_{Q_v(\mathcal{V}_u)}\big[\log P_w(\mathcal{V}_o, \mathcal{V}_u) - \log Q_v(\mathcal{V}_u)\big],\end{aligned} \tag{10}$$

where $Q_v(\mathcal{V}_u)$ is the variational distribution of the latent variables $\mathcal{V}_u$. **KL** denotes the KL divergence and the equality holds when KL divergence is 0, i.e., the variational distribution equals to the true posterior distribution: $Q_v(\mathcal{V}_u) = P_w(\mathcal{V}_u|\mathcal{V}_o)$. To train this objective, we follow the strategy of variational EM algorithm [58] from two steps: E-step and M-step by introducing triple LM $\mathbf{LM}_v$ and rule LM $\mathbf{LM}_w$.

E-Step: Inference Procedure. In the E-step for the inference, P_w is fixed to optimize Q_v by minimizing the KL divergence between the variational distribution $Q_v(\mathcal{V}_u)$ and the true posterior distribution $P_w(\mathcal{V}_u|\mathcal{V}_o)$. $Q_v(\mathcal{V}_u)$ is given by using mean-field approximation [59]:

$$Q_v(\mathcal{V}_u) = \prod_{(h,r,t)\in\mathcal{T}_u} Q_v(\mathcal{V}_{(h,r,t)}) = \prod_{(h,r,t)\in\mathcal{T}_u} f(\mathcal{V}_{(h,r,t)}|\mathbf{LM}_v). \tag{11}$$

$\mathbf{LM}_v$ denotes the used triple LM to infer triples' plausibility. Through minimizing the KL divergence, the optimal Q_v can be computed by [39, 60]:

$$\log Q_v(\mathcal{V}_{(h,r,t)}) = \mathbb{E}_{Q_v(\mathcal{V}_{\mathbf{MB}(h,r,t)})}[\log P_w(\mathcal{V}_{(h,r,t)}|\mathcal{V}_{\mathbf{MB}(h,r,t)})] + c, \tag{12}$$

where c is a constant and $\mathbf{MB}(h,r,t)$ denotes the Markov blanket of (h,r,t) which is triples appearing together with (h,r,t) in all rule groundings. To realize the above objective, we first calculate $P_w(\mathcal{V}_{(h,r,t)}|\mathcal{V}_{\mathbf{MB}(h,r,t)})$ using MLN with learned rules. After that, $\mathbf{LM}_v$ is utilized to calculate triple plausibility and then is optimized under the supervision of MLN output. Meanwhile, to retain the existing knowledge of KGs in $\mathbf{LM}_v$, we also add the observed triples $\mathcal{T}_o$ to the training set. Totally, the loss function of Q_v can be formally expressed as:

$$\mathcal{L}_{Q_v} = \begin{cases} \sum\limits_{(h,r,t)} \mathbb{E}_{P_w(\mathcal{V}_{(h,r,t)}|\mathcal{V}_{\mathbf{MB}(h,r,t)})}[\log Q_v(\mathcal{V}_{(h,r,t)})], (h,r,t)\in\mathcal{T}_u \\ \sum\limits_{(h,r,t)} \log Q_v(\mathcal{V}_{(h,r,t)} = 1), (h,r,t)\in\mathcal{T}_o \end{cases}. \tag{13}$$

M-Step: Learning Procedure. In M-step, the target is to learn semantic confidence for logic rules with fixed Q_v which maximize the log-likelihood function of all the triplets, i.e., $\mathbb{E}_{Q_v(\mathcal{V}_u)}\big[\log P_w(\mathcal{V}_o, \mathcal{V}_u)\big]$. However, it is intractable to directly optimize because of the partition function. In our model, the confidence of logical rules is calculated through an extremely discrete form as Eq. (8) shows. This leads to a huge amount of memory and computational overhead if the model is directly optimized by backpropagation. Actually, the rule confidence is obtained by calculating the similarity between reasoning paths. So we can optimize the model $\mathbf{LM}_w$ of rule LM indirectly with the similar strategy of topology context learning, because it constructs the semantic association between reasoning paths through contrastive learning. To reflect the prior knowledge of Q_v, we sample the training path set $\mathcal{P}_{Q_v}$ under the supervision of Q_v. In this way, the model $\mathbf{LM}_w$ can be optimized by the following loss function, which shares the same calculation process with Eq. (7):

$$\mathcal{L}_{P_w} = \mathcal{L}_{TCL}(\mathcal{P}_{Q_v}). \tag{14}$$

4.3. Triple embedding

In this section, we introduce how to implement $\mathbf{LM}_v$ to acquire the local context of fact triples. Specifically, for a triple (h,r,t) modeling, the strategy of translational distance approaches is to conduct structure learning by measuring spatial distance. They defines a score function, such as $s(h,r,t) = -\|f(\mathbf{h},\mathbf{r}) - \mathbf{t}\|$ for triples. $f(\mathbf{h},\mathbf{r})$ is the combination function for a head embedding $\mathbf{h}$ and a relation embedding $\mathbf{r}$, i.e. $f(\mathbf{h},\mathbf{r}) = \mathbf{h} + \mathbf{r}$ in TransE.

Algorithm 2: The overall process of FTL-LM.

Input: knowledge graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}_o\}$, hyperparameters.
Output: model $\mathbf{LM}_v$ fused topology contexts and logical rules.

Search rules and filrer through standard confidence, generate rule set Γ and hidden triples $\mathcal{T}_u$;
Initialize $\mathbf{LM}_c$ with pre-trained language model;
Generate topology path set $\mathcal{P}$ via Algorithm 1;
for *iter i=1 to* $Epoch_c$ **do**
  Optimize $\mathbf{LM}_c$ on $\mathcal{P}$; ▷ Eq. (7)
Initialize $\mathbf{LM}_v$ and $\mathbf{LM}_w$ with weights of $\mathbf{LM}_c$;
for *iter k=1 to* $Iter_l$ **do**
  /*——E-step: fix $\mathbf{LM}_w$——*/
  Calculate rule confidence using $\mathbf{LM}_w$; ▷ Eq. (8)
  Generate valid hidden triples $\mathcal{T}_u'$;
  Optimize $\mathbf{LM}_v$ on $\mathcal{T}_o$ and $\mathcal{T}_u'$; ▷ Eq. (20)
  /*——M-step: fix $\mathbf{LM}_v$——*/
  Generate valid hidden triples $\mathcal{T}_u'$ using $\mathbf{LM}_v$, and conduct random-walk for path set $\mathcal{P}_{Q_v}$;
  Optimize $\mathbf{LM}_w$ on $\mathcal{P}_{Q_v}$; ▷ Eq. (14)
Return model $\mathbf{LM}_v$.

Whereas, in our model, the cosine similarity of the head-relation pair vector and the tail vector acquired by LMs are introduced to model the validity of triples:

$$s(h,r,t) = \mathrm{sim}(\mathbf{e}_{h,r}, \mathbf{e}_t) = \frac{\mathbf{e}_{h,r}^{\top}\mathbf{e}_t}{\|\mathbf{e}_{h,r}\| \cdot \|\mathbf{e}_t\|}, \tag{15}$$

where $\mathbf{e}_{h,r}$ is the embedding vector of the head-relation pair, while $\mathbf{e}_t$ is the counterpart of the tail entity. They are all obtained by LMs with the following inputs:

$$X_{(h,r)} : \big[\texttt{[CLS]}, w_{h1}, w_{h2}, \ldots, \texttt{[SEP]}, w_{r1}, w_{r2}, \ldots, \texttt{[SEP]}\big], \tag{16}$$

$$X_t : \big[\texttt{[CLS]}, w_{t1}, w_{t2}, \ldots, \texttt{[SEP]}\big], \tag{17}$$

$$\mathbf{e}_{h,r} = \mathrm{Pool}\big(\mathbf{LM}_v(X_{(h,r)})\big), \tag{18}$$

$$\mathbf{e}_t = \mathrm{Pool}\big(\mathbf{LM}_v(X_t)\big). \tag{19}$$

To train the model, Eq. (13) is usually inefficient because it needs to sample among all hidden triples, even if its score is low. Specifically, we filter the hidden triples with the highest scores through the triple threshold η for loss calculation:

$$\mathcal{L}_{triple} = \sum_{(h,r,t)\in\mathcal{T}} -\log \frac{e^{s(h,r,t)/\tau}}{e^{s(h,r,t)/\tau} + \sum_{t'\in N_{h,r}} e^{s(h,r,t')/\tau}}, \tag{20}$$

where $N_{h,r}$ are sampled negative entities for the triple (h,r,t), i.e., $N_{h,r} = \{e|(h,r,e) \notin \mathcal{T}\}$. $\mathcal{T} = \mathcal{T}_o \cup \mathcal{T}_u'$ where $\mathcal{T}_u'$ is the most valid hidden triples whose score is large than η using $\mathbf{LM}_w$ among original hidden triple $\mathcal{T}_u$.

4.4. Overall process and training regime

Our model LM-FTL actually follows a two-stage fine-tuning strategy. In the first stage, we sample topology paths and corresponding reasoning paths from the KG. Then the mask language modeling and contrastive path learning is conducted by optimizing the model $\mathbf{LM}_c$ of topology LM, by which the model fuses topology contexts of the KG. Based on this, we teach it with logical rules in the second stage, where a variational EM method is utilized to conduct soft distilling. Language model $\mathbf{LM}_v$ and $\mathbf{LM}_w$ are utilized to model Q_v (in E-step) and P_w (in M-step), respectively. In each step, we use triple threshold η to filter valid hidden triples. If the score of a triple is larger than η, we then consider it as a positive triple.

Table 2
The statistics of WN18RR, FB15k-237 and UMLS.

Dataset	#Ent	#Rel	#Train	#Val	#Test
WN18RR	40,943	11	86,835	3034	3134
FB15k-237	14,541	237	272,115	17,535	20,466
UMLS	135	46	5216	652	661

To obtain candidate rules, the brute-force search similar to AMIE+ [61] is utilized to obtain closed Horn rules. The standard confidence which is the quotient of the number of rule body groundings and rule groundings is used to filter. Further, we can narrow down the hidden triple $\mathcal{T}_u$ to the part that these rules can deduce, which greatly reduces the computational load of the model. In the process of topology context learning, triple embedding and rule learning of FTL-LM, reasoning paths are all used as inputs to modeling objects. Although their generation methods and specific purposes are different, they are all directly input into LM and then the semantic similarity is calculated by cosine metric. Through such a consistent calculation process, we fuse topology contexts and logical rules of a KG into LMs. The overall process of FTL-LM is summarized in Algorithm 2.

In our LM-FTL, we utilize the similar strategy with pLogicNet for soft rule distilling. However, their implementations are totally different. On one hand, pLogicNet models triple validity with simple embedding method in E-step, such as TransE and DistMult. Differently, it is realized by introducing LM without additional parameters in our method. On other hand, LM-FTL calculates the rule confidence through the comparison of reasoning paths and updates them with new generated paths involving valid hidden triples. But pLogicNet simply conducts updates by the gradient descent which is calculated by predicted and labeled values.

5. Experiments and results

5.1. Datasets and evaluation metrics

We conduct experiments on three popular KGs, i.e., WN18RR [62], FB15k-237 [63] and UMLS [62]. WN18RR is a subset from WordNet [64], which consists of English phrases and their semantic relations. FB15k-237 is from Freebase [65] that contains abundant facts of the real world. UMLS contains medical semantic entities and their relations, which can be viewed as a domain KG that is widely used for the KGC task. All of them are challenging and most popular benchmarks. The statistics of them are summarized in Table 2.

To evaluate the KGC effectiveness of FTL-LM and other baselines, the head or tail of a test triple is removed. This means the model will predict the correct entity given query $(h,r,?)$ or $(?,r,t)$ for a valid triple (h,r,t). Uniformly, a reverse relation for each relation is added to the training set, so predicting $(?,r,t)$ is equivalent to predicting its inverse query $(t,r^{-1},?)$. As usual KGC tasks, the $filter$ ranking metrics [18] is utilized for evaluation, which masks all other correct triples in train, valid or test dataset for a specific query. Specifically, the mean rank (MR), mean reciprocal rank (MRR) and Hits@k are utilized as performance evaluation metrics, which is consistent with the setting of mainstream KGC tasks. Hits@k represents the proportion of predicting ranks in the top k, where Hits@1, Hits@3 and Hits@10 are used. For the results, a lower MR, as well as a higher MRR or Hits@k, generally indicate better performance of the model.

5.2. Baselines and experiment setup

To verify the performance of our proposed FTL-LM on the KGC task, seventeen strong baselines of four types are selected for comparison. They can be categorized as follows:

- LM-based: using LMs to process the text of triples to complete KGC. Five methods are utilized: KG-BERT [22], MTL-KGC [23], MLMLM [24], StAR [25] and LP-BERT [26]. We regard these methods as the main comparison objects as they and our FTL-LM all use the same calculation tool, i.e., LMs.
- Fact-embedding: learning entities and relations embeddings for fact triples modeling. We benchmark with five methods: TransE [18], DistMult [19], ComplEx [31], RotatE [29] and QuatE [66].
- Topology-embedding: aggregating neighbor information using GCN to model topology structures. We benchmark with R-GCN [20], VR-GCN [32] and CompGCN [21].
- Rule-fused: reasoning by integrating logical rules. We benchmark with Neural LP [36]. DRUM [37], pLogicNet [39] and RNNLogic [40].

When conducting experiments, the pre-trained LM *RoBERTa-base* [44] is utilized, which is in line with the backbone of LP-BERT [26] (the strongest baseline in LM-based methods). This is because of the excellent performance of *RoBERTa-base* in NLP and less space complexity. In the heterogeneous random walk algorithm, the β is set to 1.2 and θ is set to 0.9. The min and max path length is 3 and 6. And the number of topology path is 100k. The text sequence of entities and relations is limited to a maximum of 64 tokens. For topology context learning, the number of negative paths is 8. The batch size is 8 and the maximum number of epochs for training is 20. For triple embedding, the number of negative samples and the batch size are set to 64 and 256 respectively. The contrastive temperature τ is set to 0.05. To obtain candidate rules, we first search for Horn rules with standard confidence greater than 0.4 and then generate hidden triples. The maximum rule length is set to 3 and 2 for WN18RR and FB15k-237 for efficiency. For rule distilling, we first train triple LM on observed triples for 20 epochs. Then we alternatively optimize triple LM and rule LM using the EM algorithm, where there are 2 epochs for each update and triple threshold η is sets 0.9. And in M-Step, we sample topology paths that are started by the valid hidden triples. The number is 20k and 40k for WN18RR and FB15k-237 respectively. For experiments on UMLS, the experimental hyperparameters are consistent with that of WN18RR. In all training procedures, AdamW [67] algorithm with the learning rate 2e-5 is utilized. To conduct experiments, the model is implemented using PyTorch[1] and the pre-trained LMs are taken from HuggingFace.[2] We utilized a GPU server for training and evaluating the model. It contains a 24G RTX3090 GPU and has 4 Intel(R) Xeon(R) Gold 6346 CPUs with 3.10 GHz and 16 CPU cores. In addition, it contains 256G memory space.

5.3. Comparison results

We compare the performance of our FTL-LM with the performance of the baselines, and the experimental results in the WN18RR and FB15k-237 datasets are shown in Table 3, where we have the following three observations:

Firstly, compared with the five LM-based baselines (the most direct baselines), FTL-LM yields better results. It exceeds the best LM-based baseline (LP-BERT) on 9 out 10 evaluation metrics over the two benchmarking datasets by the average gains of 0.070, 0.062, 0.026, and 0.050 on Hits@1, Hits@3, Hits@10, and MRR, respectively. LP-BERT only outperforms FTL-LM with a slight margin in MR (25) on FB15k-237. StAR presents better performance in MR. This is likely because it uses a stronger pre-trained LM, RoBERTa-large, while FTL-LM and LP-BERT use RoBERTa-base [25,26]. Nevertheless, the rest of evaluation metrics of StAR are much lower than FTL-LM. It shows that in the LM-based KGC learning paradigm, our proposed FTL-LM achieves the strongest results in most of evaluation metrics.

Secondly, compared with fact-embedding methods, FTL-LM achieves the best performance in 8 out of 10 metrics, where Hits@10 of FTL-LM is slight lower than that of RotatE by 0.012; MR of RotatE and QuatE is slight lower than that of FTL-LM by 2 and 3 respectively in the FB15k-237 dataset. Noticeably, in the WN18RR dataset, FTL-LM has achieved a comprehensive performance boost. For example, FTL-LM exceeds the state-of-the-art baseline QuatE in the fact-embedding cluster by 0.016, 0.137, 0.209, 3385 and 0.062 in Hits@1, Hits@3, Hits@10, MR and MRR, respectively. In the FB15k-237 dataset, it also achieves 0.032, 0.044 and 0.026 improvements in Hits@1, Hits@3 and Hits@10, compared with QuatE.

Thirdly, compared with topology-embedding and rule-fused methods, FTL-LM also delivers good performance. The performance of FTL-LM is much better than model R-GCN, VR-GCN, Neural LP and DRUM. Compared with CompGCN, pLogicNet and RNNLogic, our FTL-LM performance is competitive in the FB15k-237 dataset. However, in the WN18RR dataset, FTL-LM achieves great improvements, for example, 0.143/0.227, 0.226/0.242 and 0.140/0.215 increase on Hits@3/Hits@10 compared with these three strong baselines. CompGCN yields marginal improvements in the FB15k-237 dataset, probably because it enhances the expression effect of GCNs by introducing different aggregation operations. However, CompGCN and other topology embedding-based methods cannot handle the inductive setting of KGs, because they are designed for the static KGC. These approaches have a narrow application scope, because KGs are ever-evolving to increase entities and triples in the real world [33,34].

From the above observations, we can verify the effectiveness of our model FTL-LM. FTL-LM shows great potential to fuse topology contexts and logical rules in LMs for KGC. Such a feature allows FTL-LM to achieve great flexibility in continuously learning KGC from new entities and relations in a KG. The developments in pre-trained LMs can potentially empower FTL-LM to further improve its accuracy. Another noteworthy phenomenon is that FLM-LM achieves higher accuracy gains in the WN18RR dataset than that in the FB15k-237 dataset. This is likely due to the fact that the former has fewer relations (11 vs. 237) and the patterns of topology contexts and logical rules are relatively simple. In addition, the experimental results in the UMLS dataset are in Table 4. We can observe that the performance of our FTL-LM is comparable to LP-BERT and exceeds all other LM-based or fact-embedding baselines. It further demonstrates the superiority of the proposed FTL-LM.

In our model FTL-LM, a variational EM algorithm is introduced for soft logical rule distilling. Although it seems time-consuming, it is actually quite computationally efficient. Each training epoch of the E-step has an average runtime of 24 min in the WN18RR dataset, which is slightly shorter than the counterpart of LM-based models, e.g., StAR needs 28 min on our GPU server. Additionally, each training epoch of the M-step has an average runtime of 12 min. Thus, the consumption time of our model is 1.28 times that of the StAR model (36 min vs. 28 min). Considering that our model achieves 20.9%, 14.6%, 6.4% and 14.2% performance improvements on Hits@1, Hits@3, Hits@10 and MRR metric in the WN18RR dataset, we believe such time consumption is worthwhile. This is because efficiency is not a priority in our task. The KGC task means to accurately complete a known KG off-line, so that it can be better utilized on a downstream task. An accurate model is more valuable than a fast model in this domain, given a slight difference in time costs.

5.4. Ablation studies

To investigate the effectiveness of topology contexts and logical rules for KGC, we conduct ablation studies in both WN18RR and FB15k-237 dataset. The results are shown in Table 5, where *w/o TC* and *w/o LR* denote the ablation for topology contexts and logical rules, respectively. When conducting experiments for the *w/o TC* setting, the process of topology context learning is removed and the standard

[1] https://pytorch.org/.
[2] https://huggingface.co/models.

Table 3
Experimental results in the WN18RR and FB15k-237 datasets. The optimal and suboptimal values of each metric are marked in bold and underlined respectively. ♣ means the results are from [25] and ♠ are from [40]. Others are directly taken from the corresponding papers.

Category	Model	WN18RR					FB15k-237				
		Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR
LM-based	KG-BERT [22] ♣	0.041	0.302	0.524	97	0.216	–	–	0.420	153	–
	MTL-KGC [23]	0.203	0.383	0.597	89	0.331	0.172	0.298	0.458	132	0.267
	MLMLM [24]	0.439	0.542	0.611	1603	0.502	–	–	–	–	–
	StAR [25] ♣	0.243	0.491	0.709	**51**	0.401	0.205	0.322	0.482	**117**	0.296
	LP-BERT [26]	0.343	0.563	0.752	92	0.482	0.223	0.336	0.490	154	0.310
Fact-embedding	TransE [18] ♣	0.043	0.441	0.532	2300	0.243	0.198	0.367	0.441	323	0.279
	DistMult [19] ♣	0.412	0.470	0.504	7000	0.444	0.199	0.301	0.446	512	0.281
	ComplEx [31] ♣	0.409	0.469	0.530	7882	0.449	0.194	0.297	0.450	546	0.278
	RotatE [29] ♣	0.428	0.492	0.571	3340	0.476	0.241	0.375	0.533	177	0.338
	QuatE [66] ♣	0.436	0.500	0.564	3472	0.481	0.221	0.342	0.495	176	0.311
Topology-embedding	R-GCN [20]♠	0.080	0.137	0.207	6700	0.123	0.100	0.181	0.300	600	0.164
	VR-GCN [32]	–	–	–	–	–	0.159	0.272	0.432	–	0.248
	CompGCN [21]	0.443	0.494	0.546	3533	0.479	**0.264**	**0.390**	**0.535**	197	**0.355**
Rule-fused	Neural LP [36]♠	0.368	0.386	0.408	–	0.381	0.173	0.259	0.361	–	0.237
	DRUM [37] ♠	0.369	0.388	0.410	–	0.382	0.174	0.261	0.364	–	0.238
	pLogicNet [39]	0.015	0.411	0.531	3436	0.230	0.231	0.369	0.528	173	0.330
	RNNLogic [40] ♠	0.446	0.497	0.558	4615	0.483	0.252	0.380	0.530	232	0.344
Ours	FTL-LM	**0.452**	**0.637**	**0.773**	87	**0.543**	0.253	0.386	0.521	179	0.348

Table 4
Experimental results in the UMLS dataset. The optimal and suboptimal values of each metric are marked in bold and underlined respectively. The results of baselines are from [26].

Category	Model	UMLS	
		Hits@10	MR
LM-based	KG-BERT [22]	0.990	1.47
	StAR [25]	0.991	1.49
	LP-BERT [26]	**1.000**	**1.18**
Fact-embedding	TransE [18]	0.989	1.84
	DistMult [19]	0.846	5.52
	ComplEx [31]	0.967	2.59
Ours	FTL-LM	0.997	1.28

Table 5
The ablation results of FTL-LM.

Ablation	WN18RR		FB15k-237	
	Hits@1	Hits@10	Hits@1	Hits@10
FTL-LM	0.452	0.773	0.253	0.521
FTL-LM w/o TC	0.423	0.759	0.241	0.521
Δ	−0.029	−0.014	−0.012	0
FTL-LM w/o LR	0.395	0.751	0.239	0.502
Δ	−0.057	−0.022	−0.014	−0.019

confidence values of candidate rules are utilized for the rule distilling process. This means the rule confidence does not update throughout the whole FTL-LM process. As the table shows, Hits@1 and Hits@10 values all decrease in both WN18RR and FB15k-237 datasets except for the Hits@10 of FB15k-237 which remains generally unchanged. It indicates the topology contexts and logical rules are both effective for LMs to perform KGC tasks, which explicitly verifies the motivation of our FTL-LM. The ablation setting *w/o LR* has a greater effect than *w/o TC*. For example, the former decreases by 0.029 on Hits@1 of WN18RR, while the latter is correspondingly reduced by 0.057. This is likely because rules directly affect a KGC model by generating hidden triples, while path modeling is a relatively indirect way.

5.5. Parameter analysis

To demonstrate how hyperparameters affect the model performance, we carry out experiments for the parameter analysis. For the BFS weight θ in heterogeneous random-walk, the values in [0.5, 1.4] with stride 0.1 are utilized. The results on the Hits@10 metric are shown as Fig. 4, where there are two settings: only train topology contexts (T) and full process of FTL-LM (T+L). For the T setting, we can see that the model performance gradually rises and finally tends to be stable. It shows that the KGC task pays more attention to the feature construction of local graph topologies. For the T+L setting, the performance growth trend is nonidentical and fluctuated. Generally, the growth ranges of Hits@10 values under setting T and T+L are about 0.080/0.025 and 0.025/0.010, respectively. This is likely because after the triple training and rule distilling, the model tends to be stable. The influence of random walk is reduced.

Table 6
Parameter analysis of the attenuation coefficient θ and contrastive temperature τ in WN18RR dataset.

θ	Hits@1	Hits@10	τ	Hits@1	Hits@10
0.70	0.437	0.756	0.04	0.451	0.772
0.75	0.436	0.758	0.05	0.452	0.773
0.80	0.442	0.754	0.06	0.449	0.771
0.85	0.445	0.763	0.07	0.438	0.748
0.90	0.452	0.773	0.08	0.433	0.756
0.95	0.449	0.762	0.09	0.424	0.731

For the attenuation coefficient θ and contrastive temperature τ, we carry out analysis in the WN18RR dataset. The results are shown as Table 6. We can observe that although the impact of θ is limited, the performance metrics will be reduced when it takes a small value. It demonstrates that the heterogeneous random-walk should consider the influence of relations and take a larger attenuation coefficient. Another observation is that small τ values can achieve better performance. When the value of τ is about 0.05, the performance of the model reaches the optimum, which shows that LMs need strong contrastive learning for precise KGC tasks.

5.6. Case study for horn rules

To show the learned Horn rules by our model FTL-LM, we list several example rules of WN18RR and FB15k-237 dataset in Table 7, which provides intuitive evidence for the reasoning interpretability of the KGC task. The confidence, body and head of rules are all given in the table. All rule confidence is calculated by the language model LM using Eq. (8), which indicates that the LM has the ability to represent logical rule in KGs. By the soft rule distilling, the LM is updated with the supervision of the hidden triples validated by another LM. In this way, our model FTL-LM is implicitly fused with logical rules.

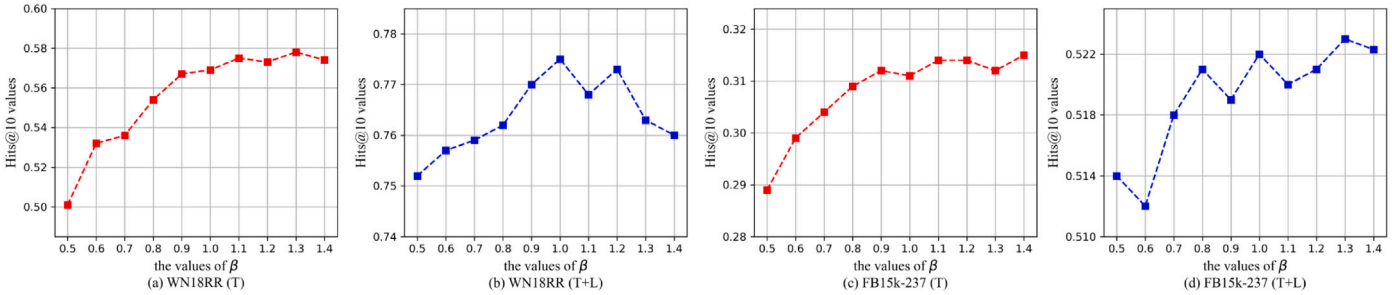

Fig. 4. Parameter analysis on the BFS weight β for knowledge graph competition.

Table 7
Example logical rules learned by FTL-LM. "−1" denotes the inverse of the original relation.

Dataset	No.	Rules of the form "$\epsilon\ body \Rightarrow head$"	
WN18RR	[1]	0.672	$memberMeronym(X,Y_1) \wedge instanceHypernym^{-1}(Y_1,Y_2) \wedge domainRegion(Y_2,Z) \Rightarrow memberMeronym(X,Z)$
	[2]	0.581	$memberMeronym(X,Y) \wedge verbGroup^{-1}(Y,Z) \Rightarrow memberMeronym(X,Z)$
	[3]	0.782	$topicOf(X,Y_1) \wedge relatedForm^{-1}(Y_1,Y_2) \wedge topicOf(Y_2,Z) \Rightarrow topicOf(X,Z)$
	[4]	0.586	$memberMeronym(X,Y_1) \wedge relatedForm^{-1}(Y_1,Y_2) \wedge hasPart(Y_2,Z) \Rightarrow hasPart(X,Z)$
	[5]	0.492	$relatedForm(X,Y_1) \wedge relatedForm^{-1}(Y_1,Y_2) \wedge domainRegion(Y_2,Z) \Rightarrow domainRegion(X,Z)$
FB15k-237	[6]	0.551	$locationState(X,Y) \wedge administrativeParent(Y,Z) \Rightarrow locationContains(X,Z)$
	[7]	0.716	$crewmember(X,Y) \wedge nationalityOf(Y,Z) \Rightarrow filmReleaseRegion(X,Z)$
	[8]	0.342	$divisionOf^{-1}(X,Y) \wedge bibsLocationState^{-1}(Y,Z) \Rightarrow bibsLocationCountry^{-1}(X,Z)$
	[9]	0.821	$locationCountry^{-1}(X,Y) \wedge divisionOf^{-1}(Y,Z) \Rightarrow locationContains(X,Z)$
	[10]	0.612	$producedBy^{-1}(X,Y) \wedge splitTo(Y,Z) \Rightarrow producedBy^{-1}(X,Z)$

6. Conclusion and future work

In this paper, we propose the FTL-LM framework to fuse topology contexts and logical rules in LMs for KGC. Since direct fusion is intractable to achieve, we adopt an indirect method. Specifically, a heterogeneous random-walk algorithm is introduced to generate topology paths. Then, the reasoning paths are obtained by the topology paths transformation. Through the mask language modeling and contrastive path learning, we fuse topology contexts in LMs. To fuse logical rules, two LMs, say a triple LM and a rule LM, are utilized in a variational EM algorithm and are optimized alternatively.

In summary, the main advantages and contributions of our model are in the following three folds: (1) Theoretically, since the current LM-based models only focus on modeling fact triples, we propose a unified framework FTL-LM to fuse topology contexts and logical rules in LMs. To our best knowledge, this is the first study that simultaneously integrates these two types of information in LMs. (2) Experimentally, our method FTL-LM surpasses all current LM-based methods on two common large KGC datasets, i.e., WN18RR and FB15k-237. For example, it achieves 2.1% and 3.1% improvement on Hits@10 metric over the state-of-the-art LM-based model LP-BERT, respectively. (3) Besides, our model has wider application potential. Compared with LM-based methods, our method of rule mining can be transferred to other intelligent application scenarios with high interpretability and reliability requirements, such as medical diagnosis [68] and investment strategy [69]. On the other hand, compared with other types of models, our model can handle tasks under KG inductive settings. It indicates that new entities or relations emerge in the test phase, which leads to the unavailability of fact-embedding and topology-embedding methods [34,70]. In the future, we will explore the application of our method to such scenarios and problems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research work is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046). This work was also supported by National Key Research and Development Program of China (2020AAA0108800), National Natural Science Foundation of China (62137002, 61937001, 62192781, 62176209, 62176207, 62106190, and 62250009), Innovative Research Group of the National Natural Science Foundation of China (61721002), Innovation Research Team of Ministry of Education (IRT_17R86), Consulting research project of Chinese academy of engineering "The Online and Offline Mixed Educational Service System for 'The Belt and Road' Training in MOOC China", "LENOVO-XJTU" Intelligent Industry Joint Laboratory Project, CCF-Lenovo Blue Ocean Research Fund, Project of China Knowledge Centre for Engineering Science and Technology, the Fundamental Research Funds for the Central Universities (xzy022021048, xhj032021013-02, xpt012022033).

References

[1] S. Ji, S. Pan, E. Cambria, P. Marttinen, P.S. Yu, A survey on knowledge graphs: Representation, acquisition and applications, IEEE Trans. Neural Netw. Learn. Syst. 33 (2) (2022) 494–514.
[2] E. Cambria, S. Ji, S. Pan, P. Yu, Guest editorial: Knowledge graph representation and reasoning, Neurocomputing 461 (2021) 494–496.
[3] J.M. Rozanec, B. Fortuna, D. Mladenic, Knowledge graph-based rich and confidentiality preserving Explainable Artificial Intelligence (XAI), Inf. Fusion 81 (2022) 91–102.
[4] N.D. Rodríguez, A. Lamas, J. Sanchez, G. Franchi, I. Donadello, S. Tabik, D. Filliat, P. Cruz, R. Montes, F. Herrera, EXplainable Neural-Symbolic Learning (*X-NeSyL*) methodology to fuse deep learning representations with expert knowledge graphs: The MonuMAI cultural heritage use case, Inf. Fusion 79 (2022) 58–83.
[5] Q. Lin, Y. Zhu, H. Lu, K. Shi, Z. Niu, Improving university faculty evaluations via multi-view knowledge graph, Future Gener. Comput. Syst. 117 (2021) 181–192.
[6] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, J. Leskovec, QA-GNN: Reasoning with language models and knowledge graphs for question answering, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), 2021, pp. 535–546.
[7] X. Zhang, A. Bosselut, M. Yasunaga, H. Ren, P. Liang, C.D. Manning, J. Leskovec, GreaseLM: Graph reasoning enhanced language models for question answering, in: International Conference on Learning Representations, ICLR, 2022.
[8] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, Q. Liu, ERNIE: Enhanced language representation with informative entities, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL, 2019, pp. 1441–1451.

[9] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, H. Wang, ERNIE 2.0: A continual pre-training framework for language understanding, in: AAAI, 2020, pp. 8968–8975.

[10] R. Mao, C. Lin, F. Guerin, Word embedding and WordNet based metaphor identification and interpretation, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Vol. 1, ACL, 2018, pp. 1222–1231.

[11] R. Mao, X. Li, M. Ge, E. Cambria, MetaPro: A computational metaphor processing model for text pre-processing, Inf. Fusion 86 (2022) 30–43.

[12] Y. Zhu, Q. Lin, H. Lu, K. Shi, D. Liu, J. Chambua, S. Wan, Z. Niu, Recommending learning objects through attentive heterogeneous graph convolution and operation-aware neural network, IEEE Trans. Knowl. Data Eng. (2021).

[13] X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu, X. He, T. Chua, Learning intents behind interactions with knowledge graph for recommendation, in: The Web Conference, WWW, 2021, pp. 878–887.

[14] E. Cambria, Q. Liu, S. Decherchi, F. Xing, K. Kwok, SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis, in: LREC, 2022, pp. 3829–3839.

[15] N. Ofek, S. Poria, L. Rokach, E. Cambria, A. Hussain, A. Shabtai, Unsupervised commonsense knowledge enrichment for domain-specific sentiment analysis, Cogn. Comput. 8 (3) (2016) 467–477.

[16] K. He, L. Yao, J. Zhang, Y. Li, C. Li, et al., Construction of genealogical knowledge graphs from obituaries: Multitask neural network extraction system, J. Med. Internet Res. 23 (8) (2021) e25670.

[17] I. Alhussien, E. Cambria, Z. NengSheng, Semantically enhanced models for commonsense knowledge acquisition, in: ICDM Workshops, 2018, pp. 1014–1021.

[18] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Advances in Neural Information Processing Systems (NeurIPS), 2013, pp. 2787–2795.

[19] B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: 3rd International Conference on Learning Representations, ICLR, 2015.

[20] M.S. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: European Semantic Web Conference, ESWC, 2018, pp. 593–607.

[21] S. Vashishth, S. Sanyal, V. Nitin, P.P. Talukdar, Composition-based multi-relational graph convolutional networks, in: International Conference on Learning Representations, ICLR, 2020.

[22] L. Yao, C. Mao, Y. Luo, KG-BERT: BERT for knowledge graph completion, 2019, CoRR abs/1909.03193. arXiv:1909.03193.

[23] B. Kim, T. Hong, Y. Ko, J. Seo, Multi-task learning for knowledge graph completion with pre-trained language models, in: Proceedings of the 28th International Conference on Computational Linguistics, COLING, 2020, pp. 1737–1743.

[24] L. Clouâtre, P. Trempe, A. Zouaq, S. Chandar, MLMLM: Link prediction with mean likelihood masked language model, in: Findings of the Association for Computational Linguistics (ACL/IJCNLP), 2021, pp. 4321–4331.

[25] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, Y. Chang, Structure-augmented text representation learning for efficient knowledge graph completion, in: The Web Conference (WWW)), 2021, pp. 1737–1748.

[26] D. Li, M. Yi, Y. He, LP-BERT: Multi-task pre-training knowledge graph BERT for link prediction, 2022, CoRR abs/2201.04843.

[27] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Trans. Knowl. Data Eng. (TKDE) 29 (12) (2017) 2724–2743.

[28] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI, 2014, pp. 1112–1119.

[29] Z. Sun, Z. Deng, J. Nie, J. Tang, RotatE: Knowledge graph embedding by relational rotation in complex space, in: 7th International Conference on Learning Representations, ICLR, 2019.

[30] M. Nickel, V. Tresp, H. Kriegel, A three-way model for collective learning on multi-relational data, in: Proceedings of the 28th International Conference on Machine Learning, ICML, 2011, pp. 809–816.

[31] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: Proceedings of the 33nd International Conference on Machine Learning, Vol. 48, ICML, 2016, pp. 2071–2080.

[32] R. Ye, X. Li, Y. Fang, H. Zang, M. Wang, A vectorized relational graph convolutional network for multi-relational network alignment, in: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI, 2019, pp. 4135–4141.

[33] K.K. Teru, E.G. Denis, W.L. Hamilton, Inductive relation prediction by subgraph reasoning, in: Proceedings of the 37th International Conference on Machine Learning, ICML, in: Proceedings of Machine Learning Research, vol. 119, 2020, pp. 9448–9457.

[34] Q. Lin, J. Liu, F. Xu, Y. Pan, Y. Zhu, L. Zhang, T. Zhao, Incorporating context graph with logical reasoning for inductive relation prediction, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR, 2022, pp. 893–903.

[35] B. Liang, H. Su, L. Gui, E. Cambria, R. Xu, Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks, Knowl.-Based Syst. 235 (107643) (2022).

[36] F. Yang, Z. Yang, W.W. Cohen, Differentiable learning of logical rules for knowledge base reasoning, in: Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 2319–2328.

[37] A. Sadeghian, M. Armandpour, P. Ding, D.Z. Wang, DRUM: End-to-end differentiable rule mining on knowledge graphs, in: Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 15321–15331.

[38] W. Li, R. Peng, Z. Li, Knowledge graph completion by jointly learning structural features and soft logical rules, IEEE Trans. Knowl. Data Eng. (TKDE) (2021).

[39] M. Qu, J. Tang, Probabilistic logic neural networks for reasoning, in: Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 7710–7720.

[40] M. Qu, J. Chen, L.A.C. Xhonneux, Y. Bengio, J. Tang, RNNLogic: Learning logic rules for reasoning on knowledge graphs, in: International Conference on Learning Representations, ICLR, 2021.

[41] Z. Wang, J. Li, Text-enhanced representation learning for knowledge graph, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI, 2016, pp. 1293–1299.

[42] R. Xie, Z. Liu, J. Jia, H. Luan, M. Sun, Representation learning of knowledge graphs with entity descriptions, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI, 2016, pp. 2659–2665.

[43] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019, pp. 4171–4186.

[44] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019, CoRR abs/1907.11692.

[45] R. Mao, X. Li, Bridging towers of multi-task learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI, 2021, pp. 13534–13542.

[46] M. Ge, R. Mao, E. Cambria, Explainable metaphor identification inspired by conceptual metaphor theory, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI, 2022, pp. 10681–10689.

[47] L. Drumond, S. Rendle, L. Schmidt-Thieme, Predicting RDF triples in incomplete knowledge bases with tensor factorization, in: Proceedings of the ACM Symposium on Applied Computing, 2012, pp. 326–331.

[48] Q. Wang, P. Huang, H. Wang, S. Dai, W. Jiang, J. Liu, Y. Lyu, Y. Zhu, H. Wu, CoKE: Contextualized knowledge graph embedding, 2019, CoRR abs/1911.02168.

[49] D. Poole, Probabilistic horn abduction and Bayesian networks, Artificial Intelligence 64 (1) (1993) 81–129.

[50] Q. Lin, J. Liu, Y. Pan, L. Zhang, X. Hu, J. Ma, Rule-enhanced iterative complementation for knowledge graph reasoning, Inform. Sci. 575 (2021) 66–79.

[51] K. Cheng, Z. Yang, M. Zhang, Y. Sun, UniKER: A unified framework for combining embedding and definite horn rule reasoning for knowledge graph inference, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2021, pp. 9753–9771.

[52] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.

[53] A. Grover, J. Leskovec, Node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 855–864.

[54] T. Gao, X. Yao, D. Chen, SimCSE: Simple contrastive learning of sentence embeddings, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, 2021, pp. 6894–6910.

[55] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, 2018, CoRR abs/1807.03748.

[56] Q. Lin, J. Liu, L. Zhang, Y. Pan, X. Hu, F. Xu, H. Zeng, Contrastive graph representations for logical formulas embedding, IEEE Trans. Knowl. Data Eng. (2021).

[57] M. Richardson, P. Domingos, Markov logic networks, Mach. Learn. 62 (1) (2006) 107–136.

[58] R.M. Neal, G.E. Hinton, A view of the EM algorithm that justifies incremental, sparse, and other variants, in: Learning in Graphical Models, Springer, 1998, pp. 355–368.

[59] M. Opper, D. Saad, Advanced Mean Field Methods: Theory and Practice, MIT Press, 2001.

[60] M. Qu, Y. Bengio, J. Tang, GMNN: Graph Markov neural networks, in: Proceedings of the 36th International Conference on Machine Learning, ICML, in: Proceedings of Machine Learning Research, vol. 97, 2019, pp. 5241–5250.

[61] L. Galárraga, C. Teflioudi, K. Hose, F.M. Suchanek, Fast rule mining in ontological knowledge bases with AMIE+, VLDB J. 24 (6) (2015) 707–730.

[62] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2D knowledge graph embeddings, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI, 2018, pp. 1811–1818.

[63] K. Toutanova, D. Chen, Observed versus latent features for knowledge base and text inference, in: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, 2015, pp. 57–66.

[64] G.A. Miller, WordNet: A lexical database for english, Commun. ACM 38 (11) (1995) 39–41.

[65] K.D. Bollacker, C. Evans, P.K. Paritosh, T. Sturge, J. Taylor, Freebase: A collaboratively created graph database for structuring human knowledge, in: Proceedings of the ACM International Conference on Management of Data, SIGMOD, ACM, 2008, pp. 1247–1250.

[66] S. Zhang, Y. Tay, L. Yao, Q. Liu, Quaternion knowledge graph embeddings, in: Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 2731–2741.
[67] I. Loshchilov, F. Hutter, Fixing weight decay regularization in adam, 2017, CoRR abs/1711.05101.
[68] I. Gadaras, L. Mikhailov, An interpretable fuzzy rule-based classification methodology for medical diagnosis, Artif. Intell. Med. 47 (1) (2009) 25–41.
[69] J. Wang, Y. Zhang, K. Tang, J. Wu, Z. Xiong, Alphastock: A buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1900–1908.
[70] H. Zha, Z. Chen, X. Yan, Inductive relation prediction by BERT, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI, 2022, pp. 5923–5931.