



Research article

PEJL: A path-enhanced joint learning approach for knowledge graph completion

Xinyu Lu, Lifang Wang, Zejun Jiang*, Shizhong Liu and Jiashi Lin

School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, China, 710072

* **Correspondence:** Email: claud@nwpu.edu.cn.

Abstract: Knowledge graphs (KGs) often suffer from incompleteness. Knowledge graph completion (KGC) is proposed to complete missing components in a KG. Most KGC methods focus on direct relations and fail to leverage rich semantic information in multi-hop paths. In contrast, path-based embedding methods can capture path information and utilize extra semantics to improve KGC. However, most path-based methods cannot take advantage of full multi-hop information and neglect to capture multiple semantic associations between single and multi-hop triples. To bridge the gap, we propose a novel path-enhanced joint learning approach called PEJL for KGC. Rather than learning multi-hop representations, PEJL can recover multi-hop embeddings by encoding full multi-hop components. Meanwhile, PEJL extends the definition of translation energy functions and generates new semantic representations for each multi-hop component, which is rarely considered in path-based methods. Specifically, we first use the path constraint resource allocation (PCRA) algorithm to extract multi-hop triples. Then we use an embedding recovering module consisting of a bidirectional gated recurrent unit (GRU) layer and a fully connected layer to obtain multi-hop embeddings. Next, we employ a KG modeling module to leverage various semantic information and model the whole knowledge graph based on translation methods. Finally, we define a joint learning approach to train our proposed PEJL. We evaluate our model on two KGC datasets: FB15K-237 and NELL-995. Experiments show the effectiveness and superiority of PEJL.

Keywords: knowledge graph completion; knowledge graphs; path; triples; multi-hop components

Mathematics Subject Classification: 68T07, 68T30

1. Introduction

Knowledge graphs [1] (KGs), e.g., Freebase [2] and Wordnet [3], describe abstract entities and their relations in a structured form, express them closer to the human cognitive world and provide a way to

store massive internet information. A typical KG is usually represented as factual triples (head entity, relation, tail entity) or (h, r, t) , indicating the relation (link) between the head entity and tail entity. KGs are the vital infrastructure of internet knowledge-driven intelligent applications and are widely used in solving problems such as recommendation [4], question-answering [5] and link prediction [6]. With big data and deep learning development, KGs have become a core driving force for developing the internet and artificial intelligence. However, KGs suffer from incompleteness. For example, 75% of person entities in Freebase [2] are stateless, and 60% of person entities in DBpedia [7] have no birthplace. Therefore, it is necessary to ensure the validity and completeness of KGs when applying them to various tasks. To improve the quality of KGs, knowledge graph completion [8] (KGC) is proposed to complete the missing components of KGs and preserve the inherent structure of KGs.

Several KGC methods have been developed to embed KG components in continuous vector spaces. Most methods can be categorized into two groups: translation algorithms and semantic matching algorithms. Translation models, such as TransE [9], TransH [10] and TransR [11], use distance-based scoring functions, which measure the plausibility of each triple by computing the distance between head entities and tail entities after translation operations. Semantic matching models, such as RESCAL [12], DistMult [13] and ComplEx [14], use similarity-based scoring functions to measure the plausibility of each triple by matching latent semantic information of KG components. Although the above methods have succeeded in KGC tasks, they only consider direct relations observed in the KGs and fail to leverage multi-hop paths. Studies [15–19] have shown that multi-hop paths can provide rich and additional semantic information between entity pairs. A multi-hop path consists of a sequence of multi-hop relations and intermediate entities, and multi-hop triples can be linked through multi-hop paths. Figure 1 is an example of multi-hop triples. In the Figure 1, two triples (*ElizabethOlsen*, *BornIn*, *California*) and (*California*, *LocatedIn*, *USA*) can form a multi-hop triple (*ElizabethOlsen*, *Nationality*, *USA*) via a path *ElizabethOlsen* \rightarrow *BornIn* \rightarrow *California* \rightarrow *LocatedIn* \rightarrow *USA*. The linked path can be used to infer the single relation *Nationality*, resulting in a single triple (*ElizabethOlsen*, *Nationality*, *USA*).

A growing number of academics are attempting to leverage multi-hop paths in given KGs because such additional information can help to improve the KGC tasks. Path ranking algorithm [20] (PRA) considers the multi-hop paths as features for a binary log-linear classifier to validate the target relation between the given entity pairs. PTransE [15] extends TransE and employs a path constraint resource allocation (PCRA) method to obtain the path embeddings by composing all the relations in each multi-hop path. RUGE [16] can learn KG representations from labeled triples, unlabeled triples and soft rules in an iterative process. MADLINK [17] encapsulates contextual information using a path selection approach before embedding structured knowledge and textual entity descriptions. PRM [18] is a path ranking algorithm that can complete KGs by combining latent semantic information with observable patterns. [19] is a semantic and data-driven path representation method that uses horn rules to generate reduced paths and advance KGC. PRN [1] creates multi-hop paths via PRA and applies the relation network in KGC to capture the semantics between triples and relations.

Despite the effectiveness of previous methods, incorporating multi-hop paths into the KGC task still encounters the following challenges:

- Limited semantic information is captured since some path-based models only consider relation components along multi-hop paths but neglect the semantic information implied in intermediate entities. Meanwhile, the heterogeneity of entities and relations in the multi-hop triples limits the

intermediate entities to be further leveraged to learn KG embeddings.

- Due to the fact that multi-hop triples can be used to infer single triples, there exist multiple associations between multi-hop and single triples. However, most path-based KGC approaches represent KGs solely utilizing entities and multi-hop paths, thus failing to explore multiple semantic associations and resulting in limited semantic accuracy of the KG embeddings.

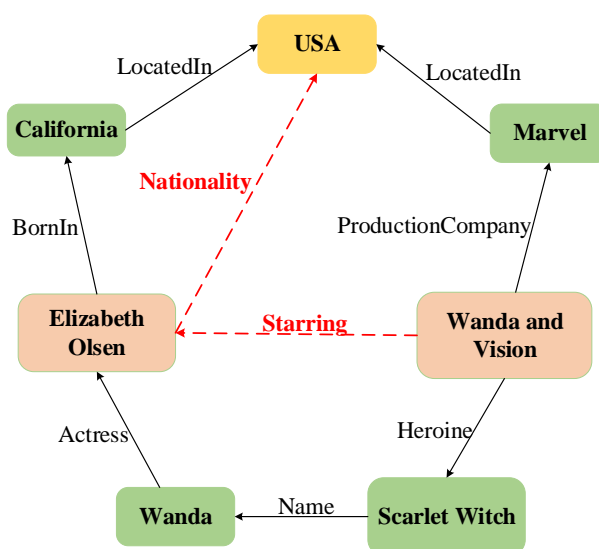


Figure 1. An example of multi-hop triples. Nodes (ellipses) represent entities, edges (arrows) represent relations, and dashed arrows represent these relations can be inferred by multi-hop paths.

In this paper, we explore how to utilize the full multi-hop components to complete knowledge graphs. This study focuses on entity prediction, which is the most prevalent task in KGC. The entity prediction task takes an incomplete triple as input and outputs a ranked list of candidate entities. To bridge the gap, we present PEJL, a path-enhanced joint learning approach that consists of an embedding recovering module and a KG modeling module. Specifically, we first employ the PCRA algorithm [15] to extract multi-hop triples. Then we use the embedding recovering module to recover the multi-hop embeddings from the KG embedding space. Significantly, our proposed embedding recovering module can leverage full path information, including multi-hop relations and intermediate entities. Next, we use the KG modeling module to capture multiple associations between single and multi-hop triples. Finally, we design a joint learning technique for multi-hop knowledge graph completion using the recovered multi-hop embeddings and the captured various semantic information. In summary, our main contributions are presented as follows:

- Instead of learning multi-hop embeddings, we provide an embedding recovering method that uses full multi-hop components to recover multi-hop embeddings from the KG embedding space.
- Unlike most KGC methods that only focus on the entities and multi-hop paths, we propose a KG modeling method that extends the definition of translation energy functions and generates new semantic representations to model the whole KG.
- We define a joint learning method that can simultaneously use single and multi-hop triples to improve the performance of knowledge graph completion.

- We test our model on two KG datasets and compare its performance to other state-of-the-art approaches. The results of entity prediction show that PEJL obtains promising performances, and extensive studies demonstrate its effectiveness and superiority.

The rest of this paper is organized as follows: Section 2 summarizes the related work of knowledge graph completion. Section 3 introduces some necessary notations and definitions in this paper. The proposed work is described in detail in Section 4. PEJL's extensive experiment findings are shown and analyzed in Section 5. Section 6 wraps up our paper and offers some suggestions for further work.

2. Related work

This section briefly reviews current studies on KGC tasks, such as knowledge graph embedding and path-based embedding approaches. Knowledge graph embedding approaches complete KGs by embedding entities and relations in a continuous low-dimensional vector space and then minimizing pairwise or logistic loss. Different from the knowledge graph embedding methods which only rely on triples, the path-based embedding approaches leverage multi-hop information to complete missing components in KGs.

2.1. Knowledge graph embedding methods

TransE [9], TransH [10] and TransR [11] are three typical translation-based KGC approaches proposed to represent entities and relations. Since the loss functions in the above algorithms are the same, they differ in translating entity components to relation components. TransE [9] treats the relation r as a translation between the head entity h and tail entity t to measure the plausibility of each triple; however, TransE may fail to capture the precise semantics of KG components. TransH [10] embeds each relation in a relation-specific hyperplane and projects entity components onto that hyperplane to deal with this flaw. TransH can distinguish semantic information of the same entity in different relations. Following the same idea as TransH, TransR introduces relation-specific spaces. TransR [11] projects entity components onto a relation-specific space employing two projection vectors and a matrix.

Unlike the translation models that compute the distance between $h + r$ and t , semantic matching methods match latent semantics of KG components in semantic spaces. RESCAL [12] associates each entity component with a vector to exploit latent semantic information and denotes each relation component as a matrix to achieve pairwise interactions. DistMult [13] simplifies RESCAL by restricting each full relation matrix to be diagonal. DistMult only can deal with symmetric relations, which is not pervasive for general KGs. To solve this problem, HoIE [21] uses cyclic-related operations to represent KG components. Since cyclic-related operations are not commutative, HoIE can model asymmetric relations. HoIE has both the efficiency of RESCAL and the simplicity of DistMult.

Besides, DSKG [22] applies an attention mechanism to model KGs through a KG-specific multi-layer recurrent neural network. HAKE [23] maps entities into polar coordinates and builds semantic hierarchies in KGs. Experiments show that HAKE can model entities at different levels in the semantic hierarchy as well as the same level. ConE [24] embeds entities as hyperbolic cones and generates hierarchical and non-hierarchical relation patterns via various cone transformations. DualE [25] models each relation as both rotation and translation based on the dual-quaternion multiplication. With a more intuitive physical and geometric understanding, this strategy extends the KG representation

space to the dual quaternion space. By aggregating information from surrounding nodes, KGEL [26] can generate two types of embedding vectors for each node, and then KGEL leverages the generated embeddings to model KGs. However, the above approaches disregard multi-hop paths, which can provide rich semantic information and help KGC perform better.

2.2. Path-based embedding methods

Path-based embedding algorithms can take advantage of multi-hop semantic information captured from paths rather than depending on length-1 paths. Path ranking algorithm [20] (PRA) is the first work to generate reliable multi-hop paths and represents path sequences to perform KGC. PTransE [15] is an extension of TransE that presents a way for learning representations for multi-hop paths. PTransE can be regarded as a typical compositional semantic problem that requires the semantic composition of the vectors of all relations to generate the path vectors. DPTransE [27] develops interactions between latent characteristics and graph features based on assuming that a single relation can be merged with multi-hop relations. DPTransE can enhance the discriminative power of multi-hop relations. However, the above algorithms represent the multi-hop paths via simple operations, which may lead to limited performance.

Other path-based embedding methods introduce reinforcement learning [28] (RL) or logic rules [19] in KGC. DeepPath [29] embeds multi-hop paths in RL and predicts reliable paths between given entity pairs. However, DeepPath generates reliable paths at the cost of searching time. Under the translation assumption, KALE [30] treats triples as atomic formulas and rules as complex formulae described by t-norm fuzzy logics. RUGE [16] can transfer logic rules into the learned embeddings by iteratively learning KG embeddings from labeled triples, unlabeled triples and soft rules. However, KALE and RUGE quantify the rules in the embedding phase, which may diminish the semantics of rules and further affect the performance of KGC.

Besides, MADLINK [17] considers both structured semantic data and textual entity descriptions and then encapsulates an entity's contextual information through a path selection method. PRM [18] is a path ranking algorithm that can complete KGs by combining latent semantic information with observable patterns. [19] is a data-driven and semantics-driven path representation method that injects horn rules into the framework to achieve condensed paths and progress KGC. PRN [1] uses PRA to generate multi-hop paths and applies the relation network to capture the semantics of triples.

Except for length-1 paths and multi-hop paths observed in KGs, some extra information can be exploited to improve the accuracy of KGC, such as entity types and multi-modal knowledge. By simultaneously embedding KG structured knowledge, textual descriptions and type information, TAPR [31] presents a type-aware attentive method for KGC. MMKRL [32] accomplishes KGC by integrating and reconstructing triples, textual descriptions and visual images. Although the above approaches could enhance the performance of KGC, it is not available for some KGs without specific information.

3. Problem statement

Notations : Given a KG $\mathcal{G} = (\mathbb{E}, \mathbb{R}, \mathbb{T})$, where \mathbb{E} is the set of entities, \mathbb{R} is the set of relations, and $\mathbb{T} = \{(h, r, t) \mid h, t \in \mathbb{E}, r \in \mathbb{R}\}$. We represent uppercase letters as matrices (e.g., W) and denote lowercase bold letters as vectors (e.g., \mathbf{x}). The operation $\|\mathbf{x}\|$ represents the norm of vector \mathbf{x} . The operation $[\ ;]$ represents the concatenation of column vectors. We define three kinds of embeddings for

each entity pair: single $\mathbf{h}_s, \mathbf{r}_s, \mathbf{t}_s \in R^k$, multi-hop $\mathbf{h}_p, \mathbf{r}_p, \mathbf{t}_p \in R^k$ and multiple $\mathbf{h}_m, \mathbf{r}_m, \mathbf{t}_m \in R^k$, where k is the corresponding embedding size.

(Definition 1) Multi-hop Triples Given a KG, a multi-hop triple can be expressed as $(h \xrightarrow{r_1} e_1 \xrightarrow{r_2} \dots e_i \xrightarrow{r_n} t)$, where $r_1 \rightarrow e_1 \rightarrow r_2 \rightarrow \dots e_i \rightarrow r_n$ denotes a multi-hop path p , e_i denotes i th immediate entity and r_n represents n th multi-hop relation. It is worth noting that all multi-hop triples can be iteratively inferred and formed into single triples. The inference chain between multi-hop and single triples can be described as follows: $r_p(h, t) \Leftarrow r_1(h, e_1) \wedge r_2(e_1, e_2) \dots \wedge r_n(e_i, t)$, where r_p denotes an inferred single relation by a multi-hop path.

(Definition 2) Knowledge Graph Completion Knowledge graph completion (KGC) attempts to predict the missing components of a triple. The KGC task can be defined by an energy function that assigns a score for each triple $(h, r, t) \in \mathbb{E} \times \mathbb{R} \times \mathbb{E}$. Rather than finding the best score triple, we rank the scores and provide a list of candidates in this study. It means high-scoring triples are more likely to be candidates than low-scoring triples.

(Definition 3) Entity Prediction Entity prediction is a KGC subtask that aims to predict an entity with a specific relation to another entity, i.e., completing missing entities in the knowledge graphs. For example, $(Wanda, Actress, ?)$ aims to predict Wanda's actress, whereas $(?, LocatedIn, USA)$ attempts to predict which city is located in the US.

4. Proposed work

This section extends the translation algorithms since most of them only take direct relations as path information and neglect to exploit rich multi-hop information to perform the KGC. The overall flow of PEJL is shown in Figure 2. PEJL first employs the PCRA algorithm to extract multi-hop triples. Then PEJL feeds knowledge graphs into the TransE model to obtain pre-trained embeddings. Next, PEJL provides an embedding recovering module that uses full path information, including multi-hop relations and intermediate entities, to recover multi-hop embeddings. In addition, a KG modeling module is proposed to learn various semantic information and model whole KGs. Finally, PEJL jointly learns KG representations and predicts candidates.

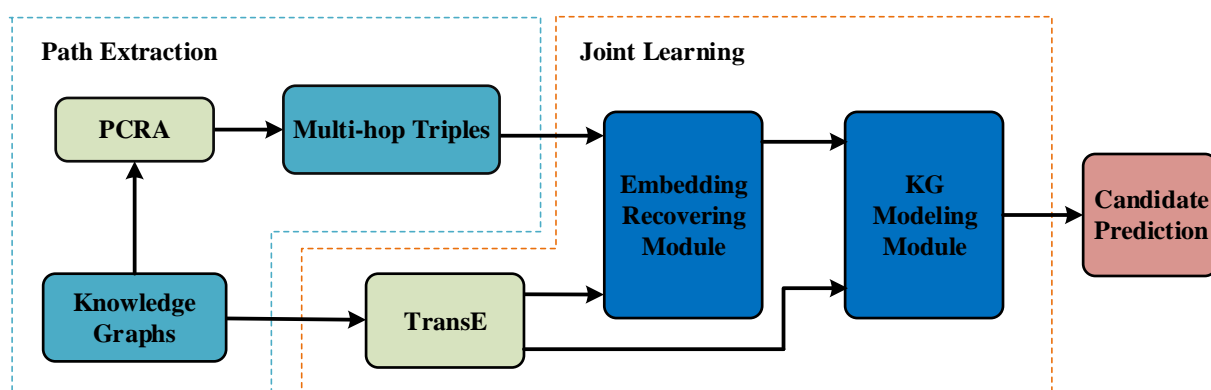


Figure 2. The overall flow of PEJL.

4.1. Extracting multi-hop triples

The path constraint resource allocation [15] (PCRA) algorithm is commonly used to evaluate the reliability of the multi-hop path p , which is essentially a resource allocation method that restricts the path. PTransE [15] first leverages PCRA to measure the reliability of multi-hop paths. The main principle behind PCRA is that a specific quantity of resources flow from the head entity h and flow down a particular path p ; the total amount of resources that finally flow to the tail entity t is used to determine the dependability of path p as a connecting path between h and t .

This paper employs the PCRA algorithm to extract multi-hop triples, consisting of a head entity h , a path p and a tail entity t . A path contains multi-hop relations and intermediate entities. Head entities, multi-hop relations, intermediate entities and tail entities comprise multi-hop components. The extracted multi-hop triples can be iteratively inferred and formed as single triples. As a result, there exist inference chains between multi-hop and single triples. For example, the single triple $(David, BornInState, California)$ can be inferred by the triple $(David, BornInCity, Angeles)$ and $(Angeles, CityInState, California)$. The chain of inference between 2-hop and single triple can be expressed as $BornInState(David, California) \Leftarrow BornInCity(David, Angeles) \wedge CityInState(Angeles, California)$. Following the PCRA reported from PTransE, we consider computational efficiency and limit the path length to a maximum of 3 hops.

4.2. Recovering multi-hop embeddings

Some approaches [1, 15, 20] focus on encoding relation components in order to learn multi-hop path embeddings, but they ignore semantic information embodied in intermediate entities. For example, for an inference chain $FilmLanguage(ThePursuitofHappiness, English) \Leftarrow CastActor(ThePursuitofHappiness, WillSmith) \wedge PersonLanguage(WillSmith, English)$, its 2-hop path $CastActor \rightarrow WillSmith \rightarrow PersonLanguage$ can be considered as a descriptive sentence to provide textual information for the single relation $FilmLanguage$. The semantic information provided by multi-hop paths will become increasingly sufficient as the number of multi-hop sequences grows. Nevertheless, the heterogeneity of relations and entities makes KGC methods difficult to learn representations of multi-hop paths.

To overcome the heterogeneity of KG components and make full use of multi-hop paths, we propose an embedding recovering module that recovers multi-hop embeddings from the KG embedding space. The idea of recovering KG representations is originated from question-answering [33], which is mainly used to recover representations of a subject entity and a predicate from each question (sentence). This paper applies this idea to obtain multi-hop embeddings in KGC. The full multi-hop components can roughly be regarded as a sentence providing semantic information and inferring the corresponding single triple. This point motivates us to propose the embedding recovering module, which attempts to build a method that takes a multi-hop triple as input and recovers multi-hop embeddings from the KG embedding space. The embedding recovering module is shown in Figure 3.

Given a multi-hop triple $(YaoMing \xrightarrow{BornIn} ShangHai \xrightarrow{LocatedIn} China)$, our goal is to subtly leverage all the multi-hop components and accurately recover corresponding semantic embeddings. The single triple $(YaoMing, Nationality, China)$ supervises the process of recovering and constructs the KG embedding space.

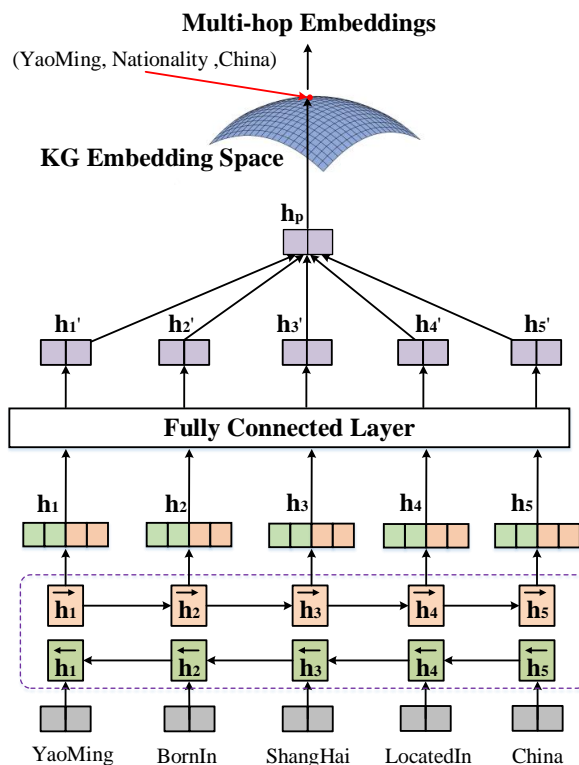


Figure 3. The architecture of embedding recovering module.

The primary architecture of the embedding recovering module consists of a bidirectional gated recurrent unit [34] (GRU) layer and a fully connected layer. We start by encoding the multi-hop sequences using a bidirectional GRU layer with reset and update gates. The reset gates decide whether the previous time step is to be reset. The update gates determine how the candidate hidden state should be output at this moment. For a multi-hop triple, we randomly initialize its tokens as embedding vectors $\{\mathbf{x}_j\}$, for $j = 1, \dots, N$. Then we learn forward hidden vectors $(\vec{\mathbf{h}}_1, \vec{\mathbf{h}}_2, \dots, \vec{\mathbf{h}}_N)$ and backward hidden vectors $(\overleftarrow{\mathbf{h}}_1, \overleftarrow{\mathbf{h}}_2, \dots, \overleftarrow{\mathbf{h}}_N)$. Taking the forward computation as an example, $\{\vec{\mathbf{h}}_j\}$ is computed through the following schemes.

$$\mathbf{r}_j = \text{sigmoid}(\mathbf{x}_j W_{xr} + \vec{\mathbf{h}}_{j-1} W_{hr} + b_r) \quad (4.1)$$

$$\mathbf{z}_j = \text{sigmoid}(\mathbf{x}_j W_{xz} + \vec{\mathbf{h}}_{j-1} W_{hz} + b_z) \quad (4.2)$$

$$\vec{\mathbf{h}}_j' = \tanh(\mathbf{x}_j W_{xh} + (\mathbf{r}_j \odot \vec{\mathbf{h}}_{j-1}) W_{hh} + b_h) \quad (4.3)$$

$$\vec{\mathbf{h}}_j = \mathbf{z}_j \odot \vec{\mathbf{h}}_{j-1} + (1 - \mathbf{z}_j) \odot \vec{\mathbf{h}}_j' \quad (4.4)$$

where \mathbf{r}_j denotes the reset gate; \mathbf{z}_j denotes the output gate; $\vec{\mathbf{h}}_j$ are the output vectors of GRU. \odot represents an element-wise operation. After learning the forward and backward vectors, we

concatenate these two vectors to obtain \mathbf{h}_j .

$$\mathbf{h}_j = \left[\vec{\mathbf{h}}_j; \overleftarrow{\mathbf{h}}_j \right] \quad (4.5)$$

Then we apply a fully connected layer to obtain \mathbf{h}'_j .

$$\mathbf{h}'_j = FCL(\mathbf{h}_j) \quad (4.6)$$

where FCL denotes a fully connected layer. Finally, all encoded multi-hop vectors are computed by a summing average operation to obtain recovered multi-hop embeddings \mathbf{h}_p . Using the same procedure, we can obtain other recovered multi-hop embeddings \mathbf{r}_p and \mathbf{t}_p . The recovered multi-hop embeddings carry rich semantic information because we encode the full multi-hop triples.

Training : In the embedding recovering module, we aim to find the closest multi-hop embeddings from the KG embeddings space. This paper uses the TransE model to construct the KG embeddings space and obtain pre-trained KG representations. We define an evaluated metric by summing the distance between the recovered multi-hop and the corresponding single embeddings. The distance metric can be defined as

$$\mathcal{L}_P = \|\mathbf{h}_p - \mathbf{h}_s\|_2 + \|\mathbf{r}_p - \mathbf{r}_s\|_2 + \|\mathbf{t}_p - \mathbf{t}_s\|_2 \quad (4.7)$$

where $\mathbf{h}_p, \mathbf{r}_p$ and \mathbf{t}_p denote the recovered embeddings. $\mathbf{h}_s, \mathbf{r}_s$ and \mathbf{t}_s are pre-trained embeddings from TransE. In \mathcal{L}_P , the pre-trained embeddings can be considered as a supervision to recover multi-hop KG components accurately.

4.3. Modeling knowledge graphs

Most path-based algorithms define energy functions with single triples and paths, neglecting semantic associations between single and multi-hop triples. For an inference chain $Nationality(YaoMing, China) \Leftarrow BornIn(YaoMing, ShangHai) \wedge LocatedIn(ShangHai, China)$, there must exist semantic associations between the 2-hop and single triple since 2-hop triple $(YaoMing \xrightarrow{BornIn} ShangHai \xrightarrow{LocatedIn} China)$ can be used to infer a single triple $(YaoMing, Nationality, China)$. This paper integrates single semantic information with multi-hop semantic information to obtain multiple semantic associations. To leverage various semantic information, we propose a KG modeling module and define new energy functions based on translation methods [9, 35]. The architecture of the KG embedding module is shown in Figure 4. After obtaining pre-trained representations and recovered multi-hop embeddings, we integrate them and define the overall energy for multi-hop triples.

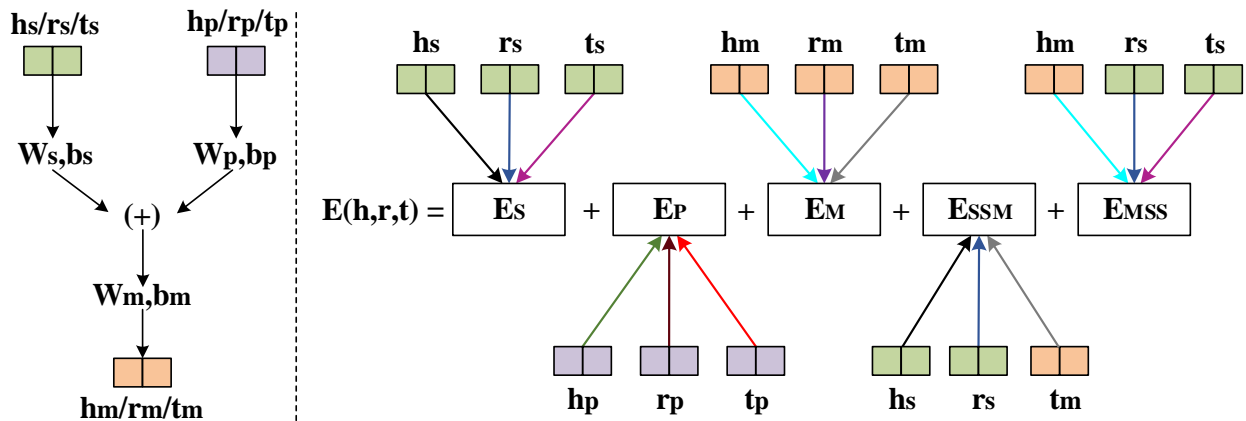


Figure 4. The architecture of KG modeling module.

Single: TransE [9] is one of the most representative translation approaches for KGC, which learns representations of KG components in the assumption of $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, i.e., the relation \mathbf{r} can be translated from the head entity \mathbf{h} to the tail entity \mathbf{t} . The intuition of TransE originates from Word2Vec [36], which obtains word embeddings to learn some semantic regularities such as *Japan – Tokyo* \approx *Germany – Berlin*. In TransE, such a semantic regularity holds since the relation *IsCapitalOf* and we can get *Tokyo + IsCapitalOf* \approx *Japan* and *Berlin + IsCapitalOf* \approx *Germany*. By assuming that head embeddings plus relation embeddings should be close to tail embeddings, the energy E_S of length-1 triples is defined as

$$E_S = \|\mathbf{h}_s + \mathbf{r}_s - \mathbf{t}_s\| \quad (4.8)$$

where \mathbf{h}_s , \mathbf{r}_s , \mathbf{t}_s denote the pre-trained embeddings obtained from TransE, respectively.

Multi-hop: TransE [9] achieves promising results with the translation assumption, leading to a series of translation-based models [10, 11]. We reconfirm that TransE is a robust baseline and follow the translation assumption to leverage multi-hop components for modeling KGs. To incorporate the multi-hop information under the translation assumption, we transfer the energy function of length-1 triples to the multi-hop components. The energy E_P of multi-hop triples is defined as

$$E_P = \|\mathbf{h}_p + \mathbf{r}_p - \mathbf{t}_p\| \quad (4.9)$$

where \mathbf{h}_p , \mathbf{r}_p , \mathbf{t}_p denote the recovered embeddings learned from the embedding recovering module, respectively. E_P represents that the multi-hop paths correspond to a translation operation between the the head entities and the tail entities.

Multiple: Since both single and multi-hop representations can provide different representations for the same entity or relation, we integrate the above two representations and mine multiple semantic representations. These new semantics can be considered a semantic enhancement for each KG component. Specifically, single and multi-hop triples are mapped by different weight matrices. Then we use a multiple weight matrix to combine the mapped length-1 and multi-hop embeddings. Taking the head components as an example, the construction of multiple head entities is as follows.

$$\mathbf{h}_s = f(W_s \times \mathbf{h}_s + b_s) \quad (4.10)$$

$$\mathbf{h}_p = f(W_p \times \mathbf{h}_p + b_p) \quad (4.11)$$

$$\mathbf{h}_m = f(W_m \times (\mathbf{h}_s + \mathbf{h}_p) + b_m) \quad (4.12)$$

where W_s , W_p and W_m denote the single weight matrix, multi-hop weight matrix and multiple weight matrix, respectively; b_s , b_p and b_m are the bias items; f is an activation function. Similar to the multiple head embeddings \mathbf{h}_m , we can obtain multiple relation embeddings \mathbf{r}_m and multiple tail embeddings \mathbf{t}_m . To incorporate the multiple information under the translation assumption, we transfer the energy function of length-1 triples to the multiple components. The energy E_M of multiple components is defined as

$$E_M = \|\mathbf{h}_m + \mathbf{r}_m - \mathbf{t}_m\| \quad (4.13)$$

E_M displays that the multiple semantics corresponds to a translation operation between multiple head semantics and multiple tail semantics.

To ensure that the length-1 and the multiple representations can interact in the same embedding space, we transfer the energy function of length-1 and define two kinds of energy functions under the translation assumption.

$$E_{SSM} = \|\mathbf{h}_s + \mathbf{r}_s - \mathbf{t}_m\| \quad (4.14)$$

$$E_{MSS} = \|\mathbf{h}_m + \mathbf{r}_s - \mathbf{t}_s\| \quad (4.15)$$

E_{SSM} and E_{MSS} enforce single and multiple representations in the same embedding space.

Following the assumption of translation approaches, we define the overall energy of a KG as

$$E(h, r, t) = E_S + E_P + E_M + E_{SSM} + E_{MSS} \quad (4.16)$$

where $E(h, r, t)$ will obtain less energy or score if a KG is correct.

Training: For the KG modeling module, we minimize a margin-based ranking loss [9] between the correct and the incorrect triples. The margin-based ranking loss is defined as

$$\mathcal{L}_O = \sum_{\substack{(h,r,t) \in \mathcal{G} \\ (h',r,t') \in \mathcal{G}'}} \max(\gamma + E(h, r, t) - E(h', r, t'), 0) \quad (4.17)$$

Here, γ is a margin to separate correct triples and incorrect triples, \mathcal{G} is a subset of correct triples and \mathcal{G}' is a subset of incorrect triples. Given correct triples, incorrect triples can be generated by randomly sampling either the head h or the tail t from the whole entity set \mathbb{E} , i.e.,

$$\begin{aligned} \mathcal{G}' = & \{(h', r, t) \mid h' \in \mathbb{E} \wedge h' \neq h \wedge (h, r, t) \in \mathcal{G}\} \\ & \cup \{(h, r, t') \mid t' \in \mathbb{E} \wedge t' \neq t \wedge (h, r, t) \in \mathcal{G}\} \end{aligned} \quad (4.18)$$

4.4. Objective formalization

To jointly train our model, we formalize the optimization objective as

$$\mathcal{L}_{PEJL} = \mathcal{L}_P + \mathcal{L}_O \quad (4.19)$$

where \mathcal{L}_P denotes a distance metric to recover multi-hop embeddings as much as possible, and \mathcal{L}_O is a pairwise ranking loss to drive the energies of correct triples higher than incorrect triples.

5. Experiments

5.1. Datasets

We adopt FB15K-237 [37] and NELL-995 [29] as the evaluated datasets in PEJL. Table 1 shows overview statistics for each dataset. FB15K-237 [37] is created from FB15K, which comes from the sizeable real-world knowledge base Freebase [2]. FB15K is not used as the benchmark for our evaluation because of the leakage problems [37] (test triples can be easily obtained by inverting training triples). FB15K-237 is more realistic and accurate when validating the performance of KGC models. FB15K-237 provides 14541 entities, 237 relations and 310116 triples. NELL-995 [29] is a subset of NELL generated from the 995th iteration of the system. NELL-995 contains 75492 entities, 200 relations and 154208 triples. This paper selects these two datasets with a large number of relations as the benchmarks and extracts multi-hop triples.

Table 1. Dataset statistics.

Dataset	Entities	Relations	Triples		
			Training	Valid	Test
FB15K-237	14541	237	272115	17535	20466
NELL-995	75492	200	123370	15000	15838

5.2. Evaluation protocol

According to the RUGE in [16], we select Mean Reciprocal Rank (MRR) and Hits@k as the evaluation metrics. We split each triple (h, r, t) into two sub-prediction tasks: tail entity prediction $(h, r, ?)$ and head entity prediction $(?, r, t)$. For each entity prediction task, we generate top-k entities as the candidates. MRR is calculated by the average reciprocal ranking of all predicted entities. Hits@k denotes the proportion of the correct entities in top-k candidates. Higher MRR and Hits@k indicate better performance. Based on the locally closed world assumption, we randomly remove either the head or the tail entities to generate incorrect triples to evaluate our model. Some generated triples should be considered correct since they may occur in the KGs. Thus we use the ‘filtered’ setting to make the evaluation more accurate. In the ‘filtered’ setting, all created triples found in the training, valid and test sets are removed from the generated triples.

5.3. Experimental settings

In experiments, we employed the PCRA [15] method to extract multi-hop triples and limit the maximum hop to 3. We used TransE [9] as a pre-trained model to learn KG representations and construct a knowledge representation space. We trained TransE with 200 dimensions for both datasets and used the same values for other hyperparameters provided by PTransE [15]. To train our proposed PEJL, we selected Adam as the optimizer, the learning rate λ among $\{0.1, 0.01, 0.001, 0.0001\}$, the batch size bs among $\{64, 128, 256\}$, the dimension k among $\{50, 100, 150, 200\}$ and the margin γ among $\{0.5, 1, 2, 4, 6, 8, 10\}$. The finally adopted optimal settings are $\lambda = 0.001$, $bs = 128$, $k = 200$, $\gamma = 10$ for FB15K-237 and NELL-995.

5.4. Comparison baselines

We compare several representative and state-of-the-art baselines, which can be categorized into two groups: (1) The models that only focus on single relations, including TransE [9], TransH [10], TransR [11], DSKG [22], HAKE [23], ModE [23], ConE [24], DualE [25] and KGEL [26]. (2) The models that introduce multi-hop paths from the KGs, including PTransE [15], RUGE [16], MADLINK [17], PRM [18] and [19]. Our proposed PEJL belongs to the second group since it leverages multi-hop paths.

- **TransE** [9] This is the most typical translation model, which embeds both entities and relations as vectors in the same space under the translation assumption.

- **TransH** [10] TransH assumes that entities and relations are in the same semantic space and makes each entity have different representations under different relations.

- **TransR** [11] TransR assumes different relations have different semantic spaces and projects each entity into the corresponding relation space.

- **DSKG** [22] This method employs a KG-specific multi-layer recurrent neural network and applies an attention mechanism to model KGs.

- **HAKE** [23] This method maps entities into the polar coordinate system and constructs the semantic hierarchies in KGs.

- **ModE** [23] is a simplified version of HAKE that uses only the modulus part but permits $[\mathbf{r}]_i < 0$.

- **ConE** [24] This method embeds entities as hyperbolic cones and uses different cone transformations to obtain hierarchical and non-hierarchical relation patterns.

- **DualE** [25] DualE can model each relation as both rotation and translation based on the dual-quaternion multiplication.

- **KGEL** [26] KGEL is an end-to-end learning framework that models a KG by aggregating information from the neighboring nodes to generate KG embeddings.

- **PTransE** [15] PTransE extends TransE and proposes a representation learning method for considering multi-hop paths.

- **RUGE** [16] RUGE learns entity and relation embeddings with iterative guidance from soft rules.

- **MADLINK** [17] This method incorporates contextual information and considers structured and textual semantic information to embed a KG.

- **PRM** [18] This method employs latent semantic information and observable patterns to perform entity prediction.

- [19] This method injects horn rules into the framework to obtain the condensed paths and advance KGC.

5.5. Results and discussion

Entity prediction results on FB15K-253 and NELL-995 are shown in Table 2. Experiment results show that our proposed PEJL outperforms the state-of-the-art models in most cases. Some detailed discussions are listed as follows:

Table 2. Entity prediction results on FB15K-237 and NELL-995. Experimental results marking * are taken from [19], marking † are taken from [24], marking ‡ are taken from [22], and '-' represents missing experimental results in the previous work. Each column's best and second-best results are **boldfaced** and underlined, respectively.

Model	FB15K-237		NELL-995	
	MRR	Hits@10	MRR	Hits@10
TransE [9]	0.294 †	0.465 †	0.219*	0.352*
TransH [10]	-	-	0.223*	0.358*
TransR [11]	0.199 ‡	0.382‡	0.232*	0.382*
DSKG [22]	0.339 ‡	0.521‡	-	-
ModE [23]	0.341	0.534	-	-
HAKE [23]	0.346	0.542	-	-
ConE [24]	0.345 †	0.540 †	-	-
DualE [25]	0.365	0.559	-	-
KGEL [26]	<u>0.414</u>	<u>0.593</u>	-	-
PTransE [15]	0.314‡	0.501 ‡	-	-
PTransE-RNN [15]	-	-	0.286*	0.423*
PTransE-ADD [15]	-	-	0.304*	0.437*
RUGE [16]	0.164	0.349	0.318	0.433
MADLINK [17]	0.347	0.529	-	-
PRM [18]	0.364	0.580	-	-
[19]	-	-	0.350*	<u>0.475*</u>
PEJL(our)	0.455	0.655	<u>0.346</u>	0.612

- Our proposed PEJL outperforms the other baselines on all metrics on two datasets apart from MRR on NELL-995. Compared to the second-best baselines KGEL [26] and [19], PEJL obtains the performance gains. For the FB15K-237 dataset, PEJL outperforms KGEL [26] with an improvement of $0.455-0.414=0.041$ (about a 10% relative improvement) in MRR and $0.655-0.593=0.062$ (about a 10% relative improvement) in Hits@10. It indicates that PEJL works well in the datasets without the reverse relations since our model leverages directed relations and does not consider the inverse patterns. For the NELL-995 dataset, PEJL obtains a relatively close MRR to the second-best result and outperforms [19] with an improvement of $0.612-0.475=0.137$ (about a 29% relative improvement) in Hits@10. Compared with [19], PEJL can obtain competitive results without introducing logic rules and entity type information. These experimental results highlight the effectiveness of our approach in terms of entity prediction.

- For the FB15K-237 dataset, PRM [18] can achieve the second-best result in path-based embedding models since PRM can exploit observable and latent semantic information in multi-hop paths. When compared to PRM [18], on FB15K-237, our model gains an improvement of $0.455-0.364=0.091$ (about a 25% relative improvement) in MRR and $0.655-0.580=0.075$ (about a 13% relative improvement) in Hits@10. This is because our method can capture and utilize extra multiple semantics from single and multi-hop triples, which is not considered in PRM.

- The NELL-995 dataset is more sparse than the FB15K-237 dataset because some of the KG

components in NELL-995 have relatively few triples or paths involving them. For the NELL-995 dataset, the path-based embedding models PTransE [15], RUGE [16, 19] and PEJL generally outperform the KG embedding models TransE [9], TransH [10] and TransR [11]. This result suggests that incorporating multi-hop paths in KGC approaches can help ease the sparse problem between entities and relations while also improving entity prediction accuracy.

- There are two main reasons our proposed PEJL can achieve competitive results in comparing KG embedding models and path-based embedding models. First, PEJL can recover multi-hop embeddings from the KG embedding space through an embedding recovering approach rather than learning multi-hop information. In particular, PEJL can leverage full multi-hop components, including head entities, full paths and tail entities. Second, unlike most KGC methods, our model can capture multiple semantics between single and multi-hop triples, potentially boosting KGC performance.

5.6. Ablation study

We perform ablation research to investigate how much each energy function in our proposed PEJL could donate. Specifically, we decrease the path length, remove multiple energy functions from our whole model and evaluate the performance of PEJL. As shown in Table 3, compared to our whole model (3-hop), the model that removes two enforcement energy functions ($-E_{SSM}-E_{MSS}$) drops 0.13 of MRR, 0.06 of Hits@1, 0.061 of Hits@3, 0.123 of Hits@5 and 0.134 of Hits@10 on FB15K-237, and 0.054 of MRR, 0.036 of Hits@1, 0.076 of Hits@3, 0.088 of Hits@5 and 0.089 of Hits@10 on NELL-995. The model that removes all multiple energy functions ($-E_M-E_{SSM}-E_{MSS}$) drops 0.171 of MRR, 0.095 of Hits@1, 0.115 of Hits@3, 0.176 of Hits@5 and 0.19 of Hits@10 on FB15K-237, and 0.1 of MRR, 0.072 of Hits@1, 0.13 of Hits@3, 0.146 of Hits@5 and 0.16 of Hits@10 on NELL-995. These results show that removing multiple energy functions will lead to performance degradation, especially when removing three multiple related energy functions. It indicates that exploiting multiple semantics can advance the performance of PEJL. The model that changes hop length to 2-hop drops 0.129 of MRR, 0.063 of Hits@1, 0.067 of Hits@3, 0.118 of Hits@5 and 0.124 of Hits@10 on FB15K-237, and 0.039 of MRR, 0.027 of Hits@1, 0.041 of Hits@3, 0.05 of Hits@5 and 0.065 of Hits@10 on NELL-995. This result emphasizes the significance of multi-hop paths leveraged in PEJL, which is consistent with our intuition, i.e., the longer the path length, the more semantic information the KGC model can capture.

Table 3. Ablation study of PEJL on FB15K-237 and NELL-995. The best results are **boldfaced**.

Model	FB15K-237					NELL-995				
	MRR	Hits@k				MRR	Hits@k			
		1	3	5	10		1	3	5	10
PEJL(3-hop)	0.455	0.288	0.464	0.549	0.655	0.346	0.218	0.435	0.482	0.612
$-E_{SSM}-E_{MSS}$	0.325	0.228	0.403	0.426	0.521	0.292	0.182	0.359	0.394	0.523
$-E_M-E_{SSM}-E_{MSS}$	0.284	0.193	0.349	0.373	0.465	0.246	0.146	0.305	0.336	0.452
PEJL(2-hop)	0.326	0.225	0.397	0.431	0.531	0.307	0.191	0.394	0.432	0.547

5.7. Further evaluations

For our model, two factors influence the performance: the margin γ and the multiple semantics. Therefore, in this section, we further compare the performance concerning these two factors in PEJL.

Effect of margin γ . Evaluation results are shown in Table 4. From the table, we observe that: (1) Margin values certainly affect the performance of PEJL in different datasets, and a suitable margin setting helps get the best performance of PEJL. (2) PEJL performs well when the margin $\gamma = 1/8/10$. When the margin value is 10, we get the best MRR evaluations; when the margin value is 1, we get most of the good Hit@k evaluations. It indicates that PEJL can learn more accurately semantic information in these settings than in other settings.

Table 4. Parameter sensitivity of margin. Each column's best is **boldfaced**.

γ	FB15K-237					NELL-995				
	MRR	Hits@k				MRR	Hits@k			
		1	3	5	10		1	3	5	10
0.5	0.407	0.286	0.456	0.541	0.652	0.308	0.191	0.386	0.424	0.547
1	0.413	0.289	0.463	0.549	0.668	0.318	0.322	0.471	0.590	0.526
2	0.391	0.275	0.437	0.518	0.628	0.299	0.169	0.355	0.406	0.527
4	0.367	0.252	0.413	0.491	0.603	0.329	0.250	0.408	0.457	0.592
6	0.356	0.246	0.398	0.472	0.580	0.325	0.200	0.409	0.455	0.587
8	0.357	0.248	0.397	0.471	0.581	0.336	0.209	0.423	0.471	0.612
10	0.455	0.288	0.464	0.549	0.655	0.346	0.218	0.435	0.482	0.612

Effect of multiple semantics. To analyze the impact of multiple semantics on PEJL, we set a threshold β to scale the total energy. The scaled energy can be defined as $E(h, r, t) = E_S + E_P + \beta(E_M + E_{SSM} + E_{MSS})$, where the larger the β , the richer the multiple semantics. Evaluation results are shown in Table 5. From the table, we observe that: (1) Overall, the MRR and Hits@k results of PEJL will increase with multiple semantics. (2) Compared with the case without multiple semantic information, PEJL gains marked improvements when $\beta = 0.2$. (3) When the scaling factor is more than 0.2, the performance improvement effect of PEJL gradually becomes flat. (4) The best evaluation of PEJL is obtained under the condition that the scaling factor is 1. (5) These findings show that having enough multiple semantic information can increase KG representation accuracy and advance the KGC.

Table 5. Parameter sensitivity of multiple semantic information. The performance degradation is marked by \downarrow and the best results are **boldfaced**.

β	FB15K-237					NELL-995				
	MRR	Hits@k				MRR	Hits@k			
		1	3	5	10		1	3	5	10
0	0.284	0.193	0.349	0.373	0.465	0.246	0.146	0.305	0.336	0.452
0.2	0.333	0.225	0.408	0.442	0.551	0.302	0.186	0.378	0.421	0.543
0.4	0.339	0.230	0.419	0.454	0.557	0.312	0.199	0.408	0.459	0.579
0.6	0.352	0.244	0.432	0.469	0.573	0.326	0.217	0.425	0.466	0.590
0.8	0.355	0.242 \downarrow	0.437	0.475	0.584	0.330	0.203 \downarrow	0.412 \downarrow	0.461 \downarrow	0.590
1	0.455	0.288	0.464	0.549	0.655	0.346	0.218	0.435	0.482	0.612

5.8. Analysis of performances in different conditions

PEJL is compared to TransE [9] and PTransE [15] in different training set proportions in this section. We also look at how the number of relations affects model performance in ModeE [23] and PEJL.

Analysis of performances of different training set proportions. TransE is the most typical KGC model, which only considers direct relations to perform KGC. PTransE and PEJL employ the PCRA algorithm to extract multi-hop paths and embed rich path information to complete KGs. Therefore, we compare these two models with PEJL to analyze model performance in different training set proportions. Figures 5 and 6 show the experimental results with different training set proportions on FB15K-237 and NELL-995. We observe that: (1) In most cases, especially when using only 20% training data, PEJL can obtain better results than TransE and PTransE on both datasets. It displays the robustness of PEJL since PEJL can make full use of limited data. (2) When the training dataset ratio is 20% and 50%, although PTransE can utilize the path information, the effect is lower than that of TransE. This may be because PTransE focuses on the multi-hop relations in the path and ignores the semantic information of intermediate entities. PEJL can simultaneously utilize multi-hop relations and intermediate entities to get better results in limited training data. (3) When the proportion of the training set exceeds 80%, the path-based model significantly outperforms TransE. This means that the amount of training data does affect the performance of the KGC models.

Analysis of performances of different kinds of relations. In FB15K-237 and NELL-995, the distribution of relations is different. To explore the influence of the number of relations on our model, we select 10 relations in both datasets. In descending order of quantity, Figure 7 shows the performance of PEJL in these different quantity distributions. For example, for FB15K-237, the largest number of relations sampled in Figure 7 is *PeoplePerson Profession*, abbreviated as “ppp”, and the least numbered relation is *TimeEventInstance OfRecurringEvent*, abbreviated as “teiore”; for NELL-995, the largest number of relations sampled in Figure 7 is *AtDate*, abbreviated as “ad”, and the least numbered relation is *PersonChargedWithCrime*, abbreviated as “pcwc”.

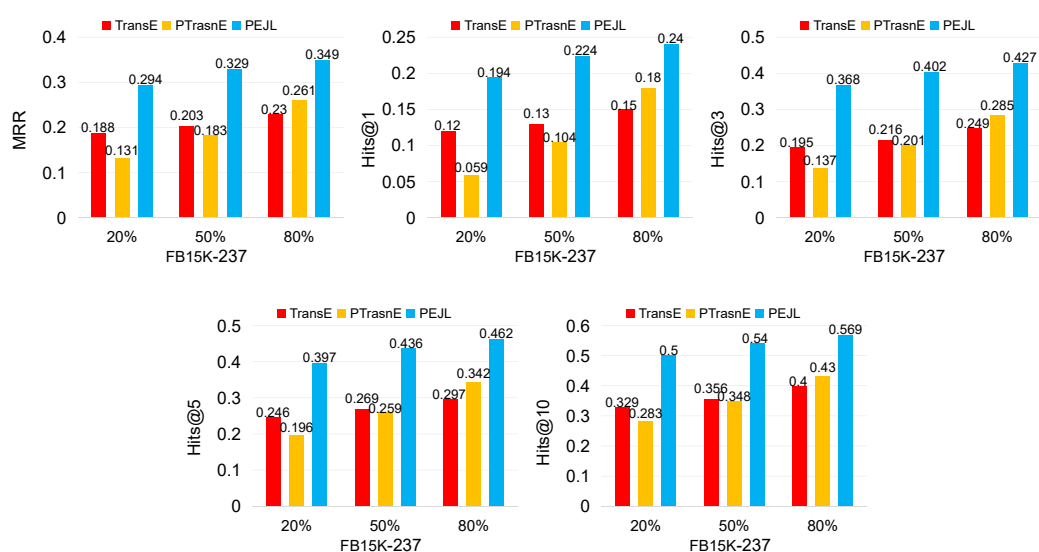


Figure 5. The performance of different training set proportions in FB15K-237.

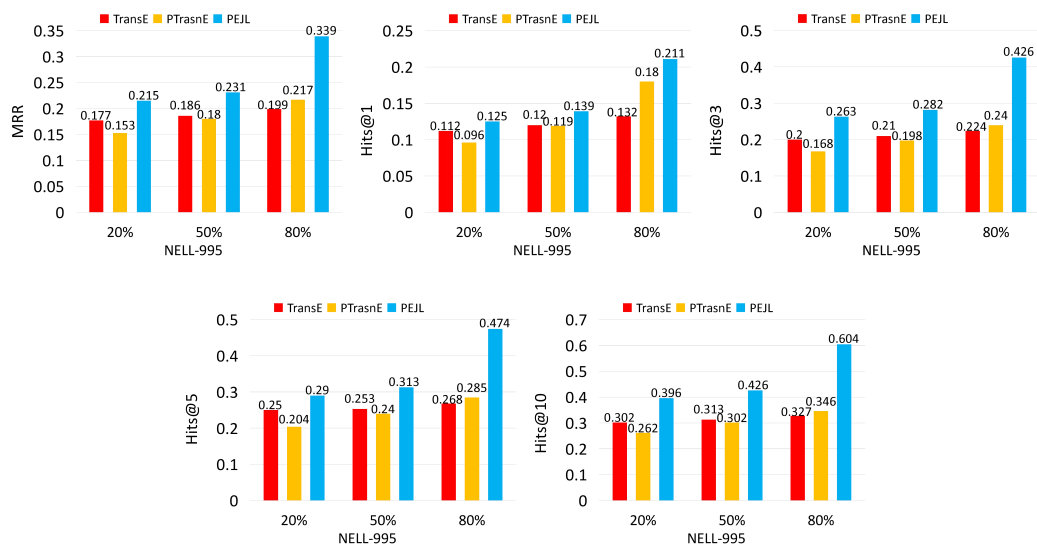


Figure 6. The performance of different training set proportions in NELL-995.

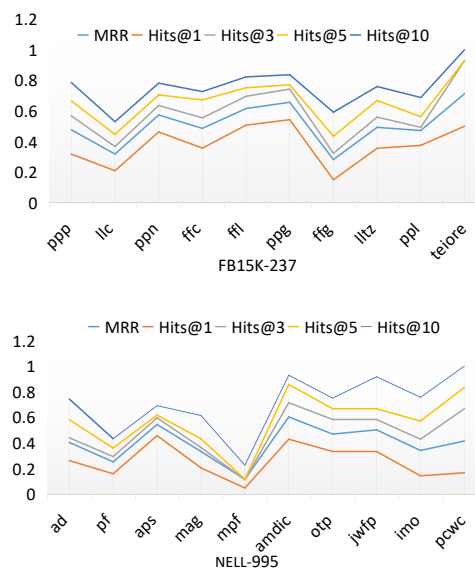


Figure 7. The performance of different relations in PEJL.

Specifically, the training set of PEJL is kept unchanged, and each type of relation is evaluated at the test phase. In Figure 7, for dataset FB15K-237, the number of relation “front” is the least but has the best performance (MRR: 0.713, Hits@1: 0.5, Hits@3: 0.929, Hits@5: 0.929, Hits@10: 1). In the NELL-995 dataset, the performance of the “mpf” with the fifth-ranked relation is much lower than that of the 5 relations with fewer than it. We can conclude that for a specific relation: (1) There is no necessary connection between the number of relations and the model’s performance in our model. (2) It is not that the greater the number of the relation, the better the model performance. (3) It is possible to obtain good performance with few relations, which indicates the superiority of our model.

5.9. Examples of tail entity prediction

To illustrate the prediction power of PEJL, Table 6 provides three examples of tail entity prediction on FB15K-237. We regard the tail prediction as a question-answering task containing queries (incomplete triples) and paths (sentences). Rather than returning a golden answer, PEJL aims to generate a list of candidates and rank all the candidates. Specifically, PEJL can match a reliable path and incorporate it to find a rank of golden answer for a query. From Table 6, we can observe that PEJL can achieve a good performance on question-answering because PEJL can accurately predict the ranks of golden answers. It once again demonstrates the effectiveness of our model.

Table 6. Examples of tail entity prediction in PEJL on FB15K-237.

FB15K-237
<p>Query: (/m/026lgs, /film/film/language, ?) Path: /film/film/country → /m/09c7w0 → /location/location/adjoin_s./location/adjoining_relationship/adjoins → /m/0d060g → /location/country/official_language Golden answer: /m/02h40lc Top-10 candidates: ['/m/02h40lc', '/m/06nm1', '/m/064_8sq', '/m/02bjrlw', '/m/04306rv', '/m/06b_j', '/m/0jzc', '/m/06mp7', '/m/05f_3', '/m/01r2l'] Rank: 1</p>
<p>Query: (/m/03k545, /people/person/places_lived./people/place_lived/location, ?) Path: /people/person/nationality → /m/02jx1 → /time/event/locations → /m/07ssc → /location/country/capital Golden answer: /m/04jpl Top-10 candidates: ['/m/0hyxv', '/m/04jpl', '/m/02_286', '/m/030qb3t', '/m/0cc56', '/m/0f2wj', '/m/07ssc', '/m/02jx1', '/m/0rh6k', '/m/01x73'] Rank: 2</p>
<p>Query: (/m/0grwj, /people/person/profession, ?) Path: /award/award_nominee/award_nominations./award/award_nomination/award_nominee → /m/05hj_k → /people/person/sibling_s./people/sibling_relationship/sibling → /m/06q8hf → /people/person/profession Golden answer: /m/03gjzk Top-10 candidates: ['/m/01d_h8', '/m/02hrh1q', '/m/03gjzk', '/m/0dxtg', '/m/02jknj', '/m/012t_z', '/m/0nbcg', '/m/0dz3r', '/m/016z4k', '/m/09jwl'] Rank: 3</p>

6. Conclusions

This paper proposes a novel path-enhanced joint learning approach called PEJL for knowledge graph completion. To take advantage of full multi-hop information, we offer an embedding recovering module that can not only leverage multi-hop relations and intermediate entities but also recover multi-hop embeddings from the KG embedding space. To discover extra semantic information, we propose

a KG modeling module to capture multiple semantics and model the whole knowledge graph. We train our model to perform entity prediction using a jointly learning strategy. Experiments show that PEJL can achieve state-of-the-art performance on most metrics. Further evaluations demonstrate that leveraging full multi-hop components and capturing multiple semantic associations indeed contribute to advance path-based KGC. Extensive experiments demonstrate the effectiveness and superiority of our proposed approach.

In future work, we aim to research the following directions to advance our model: (1) PEJL completes the knowledge graphs using single triples and full multi-hop paths. However, the process of obtaining KG representations lacks explainability since our model merely relies on neural networks. Logic rules are explainable and can provide a guidance to embed knowledge graphs. How can we increase the interpretability of PEJL with logic rules? (2) Real-life knowledge graphs continue to evolve by removing false facts and including new facts. In this way, KGs will continuously update the set of entities and relations. How can we overcome this challenge with dynamic learning?

Conflict of interest

All authors declare no conflicts of interest in this paper.

Use of AI tools declaration

All authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

References

1. W. Lee, W. Shin, B. Jagvaral, J. Roh, M. Kim, M. Lee, et al., A path-based relation networks model for knowledge graph completion, *Expert Syst. Appl.*, **182** (2021), 115273. <https://doi.org/10.1016/j.eswa.2021.115273>
2. 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 SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10–12, 2008* (Ed. J. T. Wang), ACM, 2008, 1247–1250.
3. G. A. Miller, Wordnet: A lexical database for english, *Commun. ACM*, **38** (1995), 39–41. <https://doi.org/10.1145/219717.219748>
4. M. Chen, T. Ma, X. Zhou, Cocnn: Co-occurrence CNN for recommendation, *Expert Syst. Appl.*, **195** (2022), 116595. <https://doi.org/10.1016/j.eswa.2022.116595>
5. Z. A. Guven, M. O. Ünalir, Natural language based analysis of squad: An analytical approach for BERT, *Expert Syst. Appl.*, **195** (2022), 116592. <https://doi.org/10.1016/j.eswa.2022.116592>
6. Z. Zhao, Z. Gou, Y. Du, J. Ma, T. Li, R. Zhang, A novel link prediction algorithm based on inductive matrix completion, *Expert Syst. Appl.*, **188** (2022), 116033. <https://doi.org/10.1016/j.eswa.2021.116033>

7. J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, et al., Dbpedia-A large-scale, multilingual knowledge base extracted from wikipedia, *Semant. Web*, **6** (2015), 167–195. <https://doi.org/10.1093/emph/eov017>
8. X. Chen, S. Jia, Y. Xiang, A review: Knowledge reasoning over knowledge graph, *Expert Syst. Appl.*, **141**.
9. 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 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States* (Eds. C. J. C. Burges, L. Bottou, Z. Ghahramani, K. Q. Weinberger), 2013, 2787–2795.
10. 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, July 27–31, 2014, Québec City, Québec, Canada* (Eds. C. E. Brodley, P. Stone), AAAI Press, 2014, 1112–1119.
11. Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25–30, 2015, Austin, Texas, USA* (Eds. B. Bonet, S. Koenig), AAAI Press, 2015, 2181–2187.
12. 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, Bellevue, Washington, USA, June 28–July 2, 2011* (Eeds. L. Getoor, T. Scheffer), Omnipress, 2011, 809–816.
13. 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, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings* (Eds. Y. Bengio, Y. LeCun), 2015.
14. T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, In: *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19–24, 2016* (Eds. M. Balcan, K. Q. Weinberger), vol. 48 of JMLR Workshop and Conference Proceedings, JMLR.org, 2016, 2071–2080.
15. Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, S. Liu, Modeling relation paths for representation learning of knowledge bases, In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17–21, 2015* (Eds. L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, Y. Marton), The Association for Computational Linguistics, 2015, 705–714.
16. S. Guo, Q. Wang, L. Wang, B. Wang, L. Guo, Knowledge graph embedding with iterative guidance from soft rules, In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2–7, 2018* (Eds. S. A. McIlraith, K. Q. Weinberger), AAAI Press, 2018, 4816–4823.
17. R. Biswas, M. Alam, H. Sack, Madlink: Attentive multihop and entity descriptions for link prediction in knowledge graphs, 2021.

18. X. Long, M. Yao, L. Zhuang, H. Li, S. Wang, Path ranking model for entity prediction, In: *2021 IEEE International Conference on Multimedia and Expo, ICME 2021, Shenzhen, China, July 5–9, 2021*, IEEE, 2021, 1–6.
19. G. Niu, B. Li, Y. Zhang, Y. Sheng, C. Shi, J. Li, et al., Joint semantics and data-driven path representation for knowledge graph reasoning, *Neurocomputing*, **483** (2022), 249–261. <https://doi.org/10.1016/j.neucom.2022.02.011>
20. N. Lao, T. M. Mitchell, W. W. Cohen, Random walk inference and learning in A large scale knowledge base, In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27–31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, ACL, 2011, 529–539.
21. M. Nickel, L. Rosasco, T. A. Poggio, Holographic embeddings of knowledge graphs, In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA* (Eds. D. Schuurmans, M. P. Wellman), AAAI Press, 2016, 1955–1961.
22. L. Guo, Q. Zhang, W. Ge, W. Hu, Y. Qu, DSKG: A deep sequential model for knowledge graph completion, In: *Knowledge Graph and Semantic Computing. Knowledge Computing and Language Understanding-Third China Conference, CCKS 2018, Tianjin, China, August 14–17, 2018, Revised Selected Papers*, (Eds. J. Zhao, F. van Harmelen, J. Tang, X. Han, Q. Wang, X. Li), vol. 957 of Communications in Computer and Information Science, Springer, 2018, 65–77.
23. Z. Zhang, J. Cai, Y. Zhang, J. Wang, Learning hierarchy-aware knowledge graph embeddings for link prediction, In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020*, AAAI Press, 2020, 3065–3072.
24. Y. Bai, Z. Ying, H. Ren, J. Leskovec, Modeling heterogeneous hierarchies with relation-specific hyperbolic cones, *Advances in Neural Information Processing Systems*, **34**.
25. Z. Cao, Q. Xu, Z. Yang, X. Cao, Q. Huang, Dual quaternion knowledge graph embeddings, In: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2–9, 2021*, AAAI Press, 2021, 6894–6902.
26. A. Zeb, A. U. Haq, D. Zhang, J. Chen, Z. Gong, KGEL: A novel end-to-end embedding learning framework for knowledge graph completion, *Expert Syst. Appl.*, **167** (2021), 114164. <https://doi.org/10.1016/j.eswa.2020.114164>
27. M. Zhang, Q. Wang, W. Xu, W. Li, S. Sun, Discriminative path-based knowledge graph embedding for precise link prediction, In: *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26–29, 2018, Proceedings* (Eds. G. Pasi, B. Piwowarski, L. Azzopardi, A. Hanbury), vol. 10772 of Lecture Notes in Computer Science, Springer, 2018, 276–288.
28. M. Taghian, A. Asadi, R. Safabakhsh, Learning financial asset-specific trading rules via deep reinforcement learning, *Expert Syst. Appl.*, **195** (2022), 116523. <https://doi.org/10.1016/j.eswa.2022.116523>

29. W. Xiong, T. Hoang, W. Y. Wang, Deeppath: A reinforcement learning method for knowledge graph reasoning, In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9–11, 2017* (Eds. M. Palmer, R. Hwa, S. Riedel), Association for Computational Linguistics, 2017, 564–573.
30. S. Guo, Q. Wang, L. Wang, B. Wang, L. Guo, Jointly embedding knowledge graphs and logical rules, In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1–4, 2016* (eds. J. Su, X. Carreras, K. Duh), The Association for Computational Linguistics, 2016, 192–202.
31. Y. Shen, N. Ding, H. Zheng, Y. Li, M. Yang, Modeling relation paths for knowledge graph completion, *IEEE Trans. Knowl. Data Eng.*, **33** (2021), 3607–3617.
32. X. Lu, L. Wang, Z. Jiang, S. He, S. Liu, MMKRL: A robust embedding approach for multi-modal knowledge graph representation learning, *Appl. Intell.*, **52** (2022), 7480–7497. <https://doi.org/10.1007/s10489-021-02693-9>
33. X. Huang, J. Zhang, D. Li, P. Li, Knowledge graph embedding based question answering, In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11–15, 2019* (Eds. J. S. Culpepper, A. Moffat, P. N. Bennett, K. Lerman), ACM, 2019, 105–113.
34. K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, et al., Learning phrase representations using RNN encoder-decoder for statistical machine translation, In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL* (Eds. A. Moschitti, B. Pang, W. Daelemans), ACL, 2014, 1724–1734.
35. R. Xie, S. Heinrich, Z. Liu, C. Weber, Y. Yao, S. Wermter, et al., Integrating image-based and knowledge-based representation learning, *IEEE Trans. Cogn. Dev. Syst.*, **12** (2020), 169–178. <https://doi.org/10.1109/TCDS.2019.2906685>
36. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States* (Eds. C. J. C. Burges, L. Bottou, Z. Ghahramani, K. Q. Weinberger), 2013, 3111–3119.
37. Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, *IEEE Trans. Knowl. Data Eng.*, **29** (2017), 2724–2743.



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)