

Research Article

Structure Enhanced Path Reasoning for Knowledge Graph Completion

Yilin Wang ¹, Zhen Huang ¹, Minghao Hu ², Dongsheng Li ¹, Xicheng Lu,¹ Wei Luo,²
and Dong Yang ¹

¹School of Computer, National University of Defense Technology, Changsha, China

²Information Research Center of Military Science, Academy of Military Science, Beijing, China

Correspondence should be addressed to Minghao Hu; huminghao16@gmail.com

Received 7 November 2022; Revised 19 March 2023; Accepted 30 March 2023; Published 17 May 2023

Academic Editor: Tao Li

Copyright © 2023 Yilin Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Knowledge graphs are crucial foundations for building intelligent systems, such as question answering and recommendation. However, their performance is hampered by the incompleteness of KGs, so the knowledge graph completion arises to infer whether a triple of the form (head entity, relation, tail entity) is a missing fact. The path-based approach that encodes paths from the head entity to the tail entity for reasoning achieves good performance. Previous work suggests that entity type is beneficial for learning path representations. Nevertheless, the semantics of entities are not captured accurately, as many entities are not typed or loosely typed. In addition, previous methods tend to model paths only from the forward direction but fail to capture new path patterns from the reverse direction (i.e., tail entity to head entity). In this paper, we introduce a structure enhanced path reasoning (SPR) framework to address the above-given problems. First, the model utilizes the structure of entities, i.e., their relational contexts (the relations linked from the given entity), to obtain a reliable path representation that captures correct entity semantics. This information is accessible to all nonisolated entities in all KGs, so that it can compensate the semantics for entities or KGs that have no type available. Second, we leverage the structure of paths to derive their reverse paths, so as to enhance the path representation by additionally encoding the new patterns embedded in them through a dual path encoding method. In order to verify the effectiveness of the proposed methods, we design different architectures based on LSTM and Transformer, respectively. Experimental results on two benchmark datasets, WN18RR, and FB15k-237, show that our approach apparently outperforms state-of-the-art methods on fact prediction task and relation prediction task. Furthermore, extensive experiments illustrate the benefits of enhancing path reasoning by exploiting structure information from entity relational contexts and the dual path encoding method.

1. Introduction

Knowledge graphs (KGs) contain millions of structured facts represented as triples, where each of them is stored as (head entity, relation, tail entity). As an effective way to store and search knowledge, KGs are critical for many enterprises to construct intelligent systems, such as web search [1, 2], question answering [3–5], and recommendation [6, 7]. Although many KGs have been built and published, such as WordNet [8], DBpedia [9], and Freebase [10], they are generally incomplete as a large number of facts are missing [11], which hinders the performance of intelligent systems.

Therefore, knowledge graph completion (KGC) is extremely crucial in improving the quality of KGs by inferring missing relations between entities.

Generally, prior methods of KGC can be divided into three categories: embedding-based, rule-based, and path-based models. (1) Embedding-based models [12–15] efficiently learn semantic connections between head entity, relation, and tail entity by mapping them to a continuous vector space, but they have long been criticized for lacking explainability. (2) Rule-based models [16–18] mine rules composed of relations and variable entities, where each rule is assigned with a confidence score indicating the probability

the rule holds. Despite their explainable strengths, they suffer from mining a limited form of rules dictated only by relations, resulting in poor discrimination. (3) Path-based models [19, 20] focus on modeling the path information between entities to interpretively predict their missing relation, which can learn not only regular relation patterns but also the semantics of entities on the path.

Due to the advantage that path-based methods can derive explainable inference results based on the explicit and interpretable paths between entities, they have been widely studied. The core of path-based methods is to learn the representations of paths, where each path consists of nodes representing entities and edges representing relations, as the dark green path shown in the left part of Figure 1. Earlier work only learns relation patterns of paths, by obtaining the path representation from the probability of executing random walk between two entities [19, 21], or by encoding the features of relation sequence on the path through RNN [22]. Since entities on the path also play an essential role in inference, later work [23, 24] takes entity information, such as entity itself and entity types, together with relations on the path to get the path representation by applying RNN or LSTM [25]. Although exploiting entity type information can simultaneously improve discrimination and generalization of path-based methods [24], there are still two problems that impede the performance of KGC.

First, these works only consider entities or entity types, many of which are not typed or loosely typed, leading to inaccurate and inadequate path representations. We argue that the local structure of an entity, by which we mean its neighboring relations, namely, relational contexts [26], can provide valuable contextual information. All nonisolated entities have relational contexts, and entities with different semantics own different relational contexts. Taking Figure 1 as an example, entity Jackie Chan which is an actor has relation playedInFilm, while entity Hong Kong which is a location links with relation locationLanguage. This suggests that entities linked with playedInFilm are likely to be actors, not locations. Furthermore, given a query relation, relational contexts should not receive equal attention; only those that are semantically similar to the query are important. For instance, when predicting the profession of Jackie Chan, relational contexts such as playedInFilm and directedFilm should be emphasized, while others such as gender and nationality should be disregarded.

Second, existing methods usually learn path representations only from the forward direction but fail to capture new patterns from the reverse structure of paths, i.e., reverse paths. The reverse path from tail entity to head entity contains new relation patterns along a “new” path composed of new relations. As shown in Figure 1, the relation pattern of path $\langle \text{Jackie Chan} \% \% \xrightarrow{\text{placeOfBirth}} \text{Hong Kong} \% \% \xrightarrow{\text{locationLanguage}} \text{cantonese} \rangle$ for predicting the query relation personLanguage is “placeOfBirth \wedge locationLanguage $\xrightarrow{\text{personLanguage}}$.” While its reverse path $\langle \text{cantonese} \% \% \xrightarrow{\text{locationSpeakTheLanguage}} \text{Hong Kong} \% \% \xrightarrow{\text{personBornInTheLocation}} \text{Jackie Chan} \rangle$, which infers the

reverse query relation personSpeakTheLanguage, contains different relation patterns “locationSpeakTheLanguage \wedge personBornInTheLocation $\xrightarrow{\text{personSpeakTheLanguage}}$.” So that the prediction results may differ when the encoding of reverse paths is also taken into account.

To tackle these problems, we propose a structure enhanced path reasoning (SPR) framework for KGC through leveraging the structure information of entities and paths, i.e., relational contexts and reverse paths, respectively. First, a multiperspective path encoder is applied to encode relations and entities information on a path to obtain the path representation. To get an accurate and adequate path representation that captures correct entity semantics, we propose to utilize multiperspective entity information, i.e., entity relational contexts and entity types for each entity, whose importance is considered differently through attention mechanism to be aggregated into entity features. Then, an attentive path aggregator is employed to fuse path features of all paths between the entity pair. Finally, to obtain path representations that are enriched by new relation patterns contained in reverse paths for final prediction, a dual path encoding method is proposed to combine path features of forward paths with reverse ones. Moreover, to verify the validity of the proposed approach, we not only implement the path encoder based on LSTM as in most previous work, but also apply Transformer [27] as the second architecture.

We evaluate our approach for both fact prediction task and relation prediction task on two benchmark datasets, WN18RR [14] and FB15k-237 [28]. Experimental results show that SPR outperforms state-of-the-art KGC approaches and achieves an absolute MAP gain for fact prediction over the best path-based baseline of 3.63% on WN18RR and 4.52% on FB15k-237. For relation prediction, SPR also scores the best, with MRR and hits@1 at 99.2% and 98.7% on WN18RR, respectively. Extensive experiments illustrate the effectiveness of utilizing entity relational contexts and dual path encoding. The code and datasets are available at <https://github.com/wylResearch/SPR>.

Our main contributions could be summarized as follows:

- (i) We introduce relational contexts of entities to learn accurate and sufficient path representations that capture correct entity semantics
- (ii) We propose to learn path representations by encoding relations, entity relational contexts and entity types on the path and additionally encoding reverse paths in a dual encoding manner, which enriches path representations and improves prediction performance
- (iii) The proposed method achieves the state-of-the-art for fact prediction task and relation prediction task on WN18RR and FB15k-237, and quantitative and qualitative analyses demonstrate the effectiveness of entity relational contexts and dual path encoding method for enhancing path representations

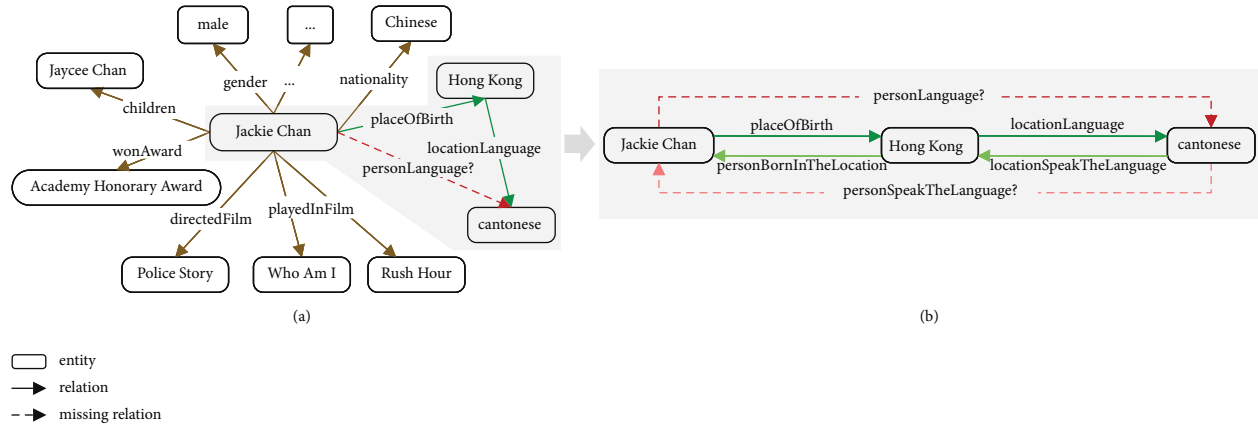


FIGURE 1: An example KG, where nodes are entities and solid edges are relations. (a) Shows some relational contexts (gold edges) of Jackie Chan. (b) shows a path (dark green edges) from Jackie Chan to cantonese for reasoning the missing fact (Jackie Chan, personLanguage, cantonese), where personLanguage is query relation (dark red dashed edges). Its reverse path (light green edges) can be seen as inferring the reverse query relation personSpeakTheLanguage (light red dashed edges).

The remainder of this article is organized as follows. Section 2 provides a brief description of related work, and Section 3 gives a formal definition of the KGC task. A detailed description of the proposed framework SPR is reported in Section 4. The experimental settings and results are discussed in Sections 5 and 6, respectively. Finally, main conclusions and future works are summarized in Section 7.

2. Related Work

2.1. Knowledge Graph Completion. There are mainly three categories of approaches to implement KGC: (1) embedding-based models learn low-dimensional distributed embeddings of entities and relations by designing a score function. The approach can be further classified into: distance-based models, such as TransE [12], TransH [29], RotatE [30], and HAKE [31]; similarity-based models, such as ComplEx [13] and ANALOGY [32]; neural network models, such as NTN [33], ConvE [14], and R-GCN [15]; models with additional information, such as DKRL [34], TKRL [35], KALE [36], and PTransE [37]. (2) Rule-based models mine interpretable rules from the knowledge graph to fill in missing facts based on existing knowledge, where a rule is defined in the form of head \leftarrow body, in which the head is an atom, i.e., a fact with variable entities, and body is a conjunction of atoms. Early approaches, such as AMIE [16] and RULES [38], mine the structure of rules in discrete spaces while learning their confidence in continuous spaces. Recent approaches, such as NeuralLP [17], DRUM [18], and NLIL [39], tend to employ neural networks to simultaneously learn the structure and confidence of rules in a continuous space. (3) Path-based models generally extract paths between an entity pair and learn the path features to make predictions. The approach falls into two major categories: path reasoning methods, such as PRA [19] and Chains [23], and reinforcement-based path finding methods, such as DeepPath [20] and MINERVA [40]. The former pays more attention to the path

feature encoding process, while the latter focuses on extracting an effective inference path. In this paper, we focus on the path reasoning methods.

2.2. Path Reasoning Methods in KGC. Paths between entities are beneficial to the explainability of reasoning in KGC. Early path reasoning approaches [19, 41] utilize the probability of walking between entities following relation sequences to get the path features. Later on, neural networks are employed to encode the path. Neelakantan et al. [22] apply RNN to encode the relation sequence as path representations. Chains [23] encodes alternate sequences of entity information and relations by RNN. APR [24] separately encodes relation sequence and entity types by LSTM and merges them into the path feature. However, the above-given methods only encode paths in the forward direction but ignore capturing new patterns from reverse paths. Although Cor-PRA [21], which is based on PRA, alleviates this problem by conducting a bidirectional random walk to get the probability as path feature, it neither learns the semantics of relations on the path nor considers encoding entity information. Different from it, we not only model entity information along with its relational contexts to enhance path representations but also capture new patterns from a novel dual path encoding method.

2.3. Entity Information in KGC. Entities are critical for predicting missing knowledge in KGs. To enhance entity embedding learned solely from facts, some embedding-based methods apply additional knowledge to enrich entity semantics, such as entity descriptions [34, 42], entity types [35, 43], and entity neighbors [15, 44]. Recently, PathCon [26] proposes to fuse relational contexts into entity representation and further perform message passing over relation graphs for relation prediction. However, PathCon represents an entity by summing up the representations of

its relational contexts, without considering their different contributions. Unlike the above approach, a query relation-guided portrait attention and the self-attention are considered in our LSTM-based and Transformer-based architectures, respectively, to capture the importance of relational contexts for each entity, so that various relational contexts contribute differently.

Among path-based methods, the entity is not considered by the earlier work [19, 22, 41], where only relations are used. In later approaches, the entities themselves and entity types are taken as entity information for learning path representation [23, 24]. Chains [23] simply sums entity type representations and combines it with entity embedding to represent an entity. APR [24] employs an attention mechanism on type hierarchies as entity semantics. While we further leverage entity relational contexts to capture reliable entity semantics, especially for entities that are not typed or loosely typed in KGs.

3. Problem Formulation

3.1. Knowledge Graph. Let $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$ be a KG, where \mathcal{E} is the set of entities, \mathcal{R} is the set of relations and \mathcal{F} is the set of facts. Each fact in \mathcal{F} can be defined as (h, r, t) , where $h, t \in \mathcal{E}$, $r \in \mathcal{R}$. For each fact (h, r, t) , we take its reverse form (t, r', h) into account, where r' denotes the reverse relation of r .

3.2. Paths. The paths from head entity h to tail entity t are denoted as $P_{h,t}$. A path $p \in P_{h,t}$ with length M can be stated as $p = \langle e_1, r_1, e_2, \dots, r_M, e_{M+1} \rangle$, where $e_i \in \mathcal{E}$, $1 \leq i \leq M+1$, $e_1 = h$, $e_{M+1} = t$, $r_j \in \mathcal{R}$, $1 \leq j \leq M$, and $(e_i, r_i, e_{i+1}) \in \mathcal{F}$. We take p as the reverse path of p , so that $p' = \langle e_{M+1}, r'_M, \dots, e_2, r'_1, e_1 \rangle \in P_{h,t}$, where $P_{h,t}$ is the set of reverse paths from t to h corresponding to $P_{h,t}$.

3.3. Relational Contexts and Entity Types. For entity e_i , we define its relational contexts C_i as the relations in facts with e_i as the head entity, that is $C_i = \{r_j | (e_i, r_j, e_k) \in \mathcal{F}, r_j \in \mathcal{R}, e_k \in \mathcal{E}\}$. To distinguish it from relations on paths, $c_{i,u}$ is used to denote one of the relational contexts in C_i , where $1 \leq u \leq |C_i|$. Let L_i denote the types of entity e_i , and $l_{i,v}$ is one of the types in L_i , where $1 \leq v \leq |L_i|$. An example path with relational contexts and types shown for each entity is displayed in Figure 2.

Given a knowledge graph \mathcal{G} , a query relation $r_q \in \mathcal{R}$, a head entity $h \in \mathcal{E}$, a tail entity $t \in \mathcal{E}$, and some paths $P_{h,t}$ extracted between the entity pair, the goal of KGC is to infer whether the triple (h, r_q, t) is true, i.e., the probability that (h, r_q, t) holds. Important symbols are summarized in Table 1, including those representations that will be described later. In addition, we use the bold letter to represent a vector, e.g., \mathbf{r} denotes the embedding of the relation r .

4. Approach

We propose a structure enhanced path-reasoning framework, namely, SPR, which enhances path representations by

incorporating relational contexts into entity semantics and encoding reverse paths in a dual path encoding manner, to perform KGC. The overall architecture is presented in Figure 3, which consists of three components: (1) the multiperspective path encoder (MPE) encodes relations and multiperspective entity information, i.e., relational contexts and types of each entity, to learn accurate and adequate path representations. We explore different architectures that are based on LSTM and Transformer, which are described in Sections 4.1 and 4.2, respectively. (2) The attentive path aggregator fuses the representations of paths, where important and useful paths are highly weighted (Section 4.3). (3) The dual path encoding combines the path representations of forward paths and reverse ones to enhance the final prediction (Section 4.4).

4.1. LSTM-Based Multiperspective Path Encoder. The multiperspective path encoder encodes relations, entity relational contexts and entity types on a path p to get its path representation \mathbf{h}^p , i.e., $f^{\text{path}}(p) = \mathbf{h}^p$, where f^{path} is the function for encoding a path. We present the architecture based on LSTM to implement f^{path} in this section, termed LSTM-MPE, as shown in Figure 4. It consists of a LSTM-based relation encoder and a LSTM-based entity encoder to encode the path information separately.

4.1.1. Relation Encoder. For a path $p \in P_{h,t}$, its relation sequence $p^r = \langle r_1, \dots, r_j, \dots, r_M \rangle$ is the key to inferring r_q between (h, t) . The relation encoder applies a LSTM, namely LSTM_r, to encode p^r , and the encoding step j can be expressed as

$$\mathbf{h}_j^r = \text{LSTM}_r(\mathbf{h}_{j-1}^r, \mathbf{r}_j), \quad (1)$$

where \mathbf{h}_j^r is the hidden state of LSTM_r at step j , and \mathbf{r}_j is the randomly initialized embedding of relation r_j . The last hidden state, formulated as \mathbf{h}^r , is used to denote the relation representation of path p .

4.1.2. Entity Encoder. Entity information plays a crucial role during the inference, which can lead to different path representations even when the relation sequences of paths are the same. Prior path-based work only considers entity or entity types as entity information, which leads to unreliable path representations that capture inaccurate and insufficient entity semantics. To compensate for the entity semantics obtained solely from entity types, we introduce relational contexts as additional entity information.

For entity sequence $p^e = \langle e_1, \dots, e_i, \dots, e_{M+1} \rangle$ of path p , its entity information includes two perspectives: entity relational contexts and entity types, namely, relational context sequence $p^C = \langle C_1, \dots, C_i, \dots, C_{M+1} \rangle$ and entity type sequence $p^L = \langle L_1, \dots, L_i, \dots, L_{M+1} \rangle$. Encoding each of them comprises two modules: a Portrait Attention module and a Sequence Encoding module. Since the two kinds of entity information are encoded in the same way, we will use the relational context sequence as an example for a detailed description.

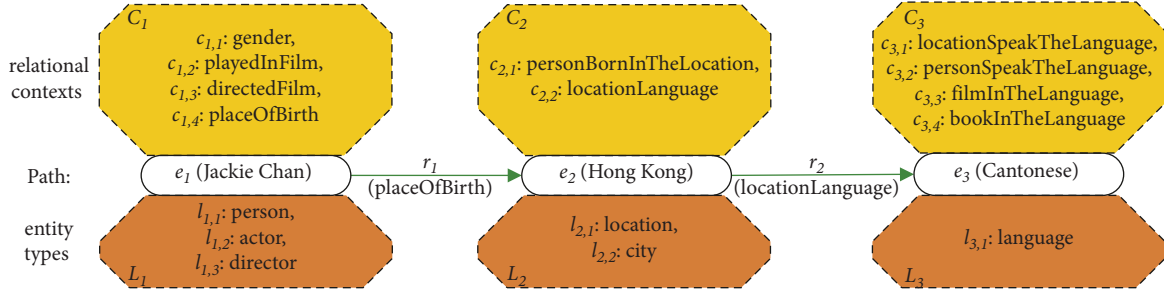


FIGURE 2: An example path $p = \langle e_1, r_1, e_2, r_2, e_3 \rangle$ with length $M = 2$, where example relational contexts and types of each entity are shown above and below the path, respectively. C_i : the relational contexts of the i -th entity e_i . $c_{i,j}$: the j -th relational context of C_i . L_i : the types of the i -th entity e_i . $l_{i,j}$: the j -th entity type of L_i .

(1) *Relational Context Portrait Attention*. This module aims to fuse the relational contexts of each entity of the path, i.e., for step i of p^e , fuse $C_i = \{c_{i,1}, \dots, c_{i,u}, \dots, c_{i,|C_i|}\}$ to obtain the feature vector $\hat{\mathbf{e}}_i^C$ of entity e_i about relational contexts. Inspired by APR [24], we use the current path features to lead the selection of important relational contexts of an entity. Furthermore, we consider that when predicting different query relations, the relational contexts of the same entity should be given different emphases. Therefore, the query relation is employed to guide the selection. Formally, for path p , the attention to relational context $c_{i,u}$ is guided by the query relation r_q and the path feature about the relational contexts selected in the previous steps. Denote the path feature about relational contexts before step i as \mathbf{h}_{i-1}^C , which we will illustrate the detail later. This historical feature and the embedding of query relation r_q are first concatenated and then feed into a single-layer feed-forward neural network f_{guide}^C to obtain the guidance vector \mathbf{g}_{i-1}^C :

$$\mathbf{g}_{i-1}^C = f_{\text{guide}}^C([\mathbf{h}_{i-1}^C, \mathbf{r}_q]), \quad (2)$$

where \mathbf{r}_q is the embedding of query relation r_q . Then, \mathbf{g}_{i-1}^C is combined with the embedding of relational context $c_{i,u}$ to compute its attention weight $\text{att}_{i,u}^C$:

$$\text{att}_{i,u}^C = f_{\text{att}}^C(\text{ReLU}(f_g^C(\mathbf{g}_{i-1}^C) + f_{\text{emb}}^C(\mathbf{c}_{i,u}))), \quad (3)$$

where $\mathbf{c}_{i,u}$ denotes the randomly initialized embedding of relational context $c_{i,u}$, and f_g^C , f_{emb}^C , and f_{att}^C are all single-layer feed-forward neural networks. Next, the normalized attention weight $\alpha_{i,u}^C$ of relational context $c_{i,u}$ is calculated as

$$\alpha_{i,u}^C = \frac{\exp(\text{att}_{i,u}^C)}{\sum_{w=1}^{|C_i|} \exp(\text{att}_{i,w}^C)}. \quad (4)$$

Now, we can compute the fused relational context representation for entity e_i , i.e., $\hat{\mathbf{e}}_i^C$, which is the weighted sum of all the transformed features of relational contexts in C_i :

$$\hat{\mathbf{e}}_i^C = \sum_{u=1}^{|C_i|} \alpha_{i,u}^C * f_{\text{emb}}^C(\mathbf{c}_{i,u}). \quad (5)$$

(2) *Relational Context Sequence Encoding*. The goal of this module is to obtain the relational context representation of the whole path, namely, \mathbf{h}^C , from relational context sequence $p^C = \langle C_1, \dots, C_i, \dots, C_{M+1} \rangle$ of p^e . We employ a LSTM, denoted as LSTM_C , to encode the features on the sequence. Let \mathbf{h}_{i-1}^C denote its hidden state of step $i-1$, which is the historical feature mentioned in equation (2). It is used to obtain the fused feature vector $\hat{\mathbf{e}}_i^C$ (equation (5)) of C_i , for input to LSTM_C at step i . Formally, the encoding step i of relational context sequence of p^e can be expressed as

$$\mathbf{h}_i^C = \text{LSTM}_C(\mathbf{h}_{i-1}^C, \hat{\mathbf{e}}_i^C). \quad (6)$$

The initialization of hidden state \mathbf{h}_0^C and cell state \mathbf{a}_0^C for LSTM_C of entity sequence p^e is guided by the relation representation \mathbf{h}^r of the corresponding relation sequence p^r :

$$\mathbf{h}_0^C = f_{\text{init}_h}^C(\mathbf{h}^r), \mathbf{a}_0^C = f_{\text{init}_c}^C(\mathbf{h}^r), \quad (7)$$

where both $f_{\text{init}_h}^C$ and $f_{\text{init}_c}^C$ are single-layer feed-forward neural networks. Let \mathbf{h}^C be the last hidden state of LSTM_C for p^e , to represent the relational context representation of the path.

(3) *Entity Type Portrait Attention and Entity Type Sequence Encoding*. The method for encoding entity type sequence $p^L = \langle L_1, \dots, L_i, \dots, L_{M+1} \rangle$ has the same structure as that for encoding relational context sequence, but with different parameters. Denote the LSTM that encodes entity type sequence as LSTM_L , then its last hidden state, notated as \mathbf{h}^L , which represents the entity type representation of path p^e can be obtained similarly.

Then, the relational context representation \mathbf{h}^C and entity type representation \mathbf{h}^L are summed to represent the entity representation \mathbf{h}^e of path p , formally written as

$$\mathbf{h}^e = \mathbf{h}^C + \mathbf{h}^L. \quad (8)$$

The relation representation and entity representation are concatenated as the path representation \mathbf{h}^p as follows, where $[,]$ means concatenation:

$$\mathbf{h}^p = [\mathbf{h}^r, \mathbf{h}^e]. \quad (9)$$

TABLE 1: Important notations and descriptions.

Groups	Notation	Description
Knowledge graph	e	An entity
	r	A relation
	r'	The reverse relation of r
Paths	P	A path, $P = \langle e_1, r_1, e_2, \dots, r_M, e_{M+1} \rangle$, where M is the length of path
	p	The reverse path corresponding to P , $p' = \langle e_{M+1}, r_M, \dots, e_2, r_1, e_1 \rangle$
	$P_{h,t}$	Paths between entity pair (h, t) , $P_{h,t} = \{p_1, p_2, \dots, p_N\}$, where N is the number of paths
	$P'_{h,t}$	The reverse paths corresponding to $P_{h,t}$, $P'_{h,t} = \{p'_1, p'_2, \dots, p'_N\}$
Relational contexts and entity types	C_i	Relational contexts of entity e_i
	$c_{i,u}$	The u -th relational context of entity e_i , $c_{i,u} \in C_i$
	L_i	Types of entity e_i
	$L_{i,v}$	The v -th type of entity e_i , $L_{i,v} \in L_i$
Sequences on path	p^r	Relation sequence of path p , $p^r = \langle r_1, \dots, r_M \rangle$
	p^e	Entity sequence of path p , $p^e = \langle e_1, \dots, e_{M+1} \rangle$
	p^C	Entity relational context sequence of path p , $p^C = \langle C_1, \dots, C_{M+1} \rangle$
	p^L	Entity type sequence of path p , $p^L = \langle L_1, \dots, L_{M+1} \rangle$
Representations	h^r	Relation representation of path p
	h^e	Entity representation of path p
	h^C	Relational context representation of path p
	h^L	Entity type representation of path p
	h^p	Path representation of path p

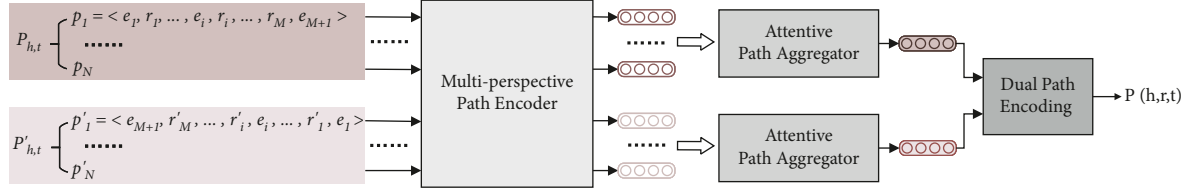


FIGURE 3: Overview of SPR, which consists of three components: multiperspectivie path encoder, attentive path aggregator, and dual path encoding. The box in the upper left shows the set of forward paths $P_{h,t}$ between entity pair (h, t) , while the lower left box means the corresponding reverse paths $P'_{h,t}$, where $e_1 = h$, $e_{M+1} = t$.

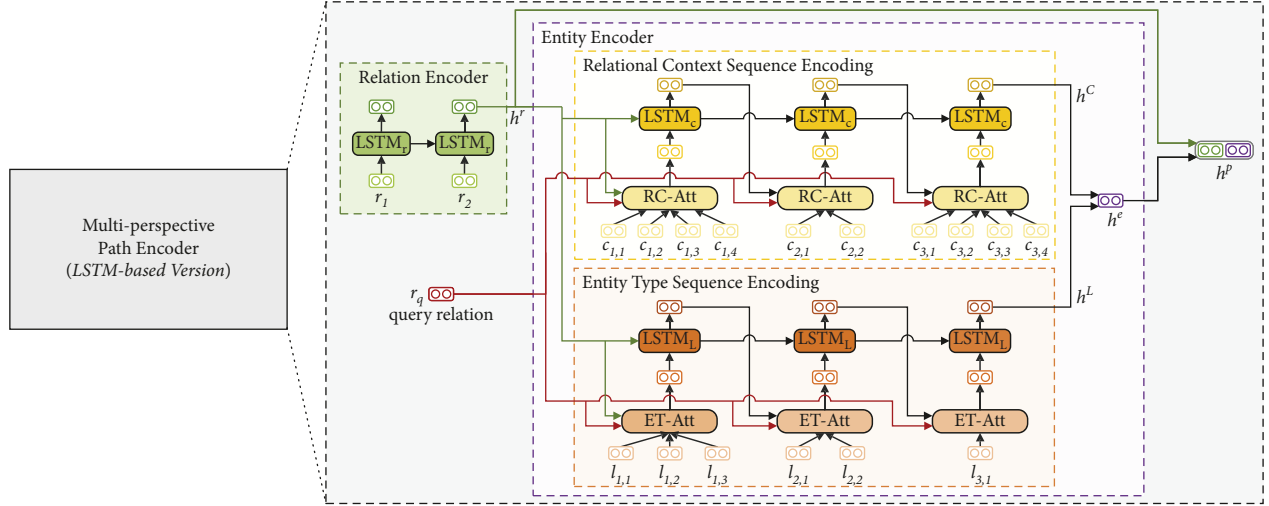


FIGURE 4: The architecture of LSTM-based multiperspectivie path encoder (LSTM-MPE) for encoding relations and multiperspectivie entity information, namely, entity relational contexts and entity types, of the forward example path p in Figure 2. It consists of a LSTM-based relation encoder and a LSTM-based entity encoder, where “RC-Att” refers to “relational context portrait attention” and “ET-Att” refers to “entity type portrait attention.”

4.2. Transformer-Based Multiperspectivie Path Encoder. Transformer-based networks show great power in fields such as natural language processing [45, 46] and computer vision [47, 48]. Although some methods apply it to solve the knowledge graph completion problem, they do not use it to encode both relation and multiperspectivie entity information on a path [49, 50]. Therefore, in this section, we try to investigate its abilities to encode path information and propose a Transformer-based multiperspectivie path encoder, notated as Transformer-MPE, to implement $f^{\text{path}}(p) = \mathbf{h}^p$.

For a path p , its relation sequence $p^r = \langle r_1, \dots, r_j, \dots, r_M \rangle$, entity relational context sequence $p^c = \langle C_{1,1}, \dots, C_{i,1}, \dots, C_{M+1,1} \rangle$, and entity type sequence $p^l = \langle L_1, \dots, L_j, \dots, L_{M+1} \rangle$ are processed into three separate input sequences, namely, the relation input sequence S^r , entity relational context input sequence S^c , and entity type input sequence S^l . Inspired by pretrained language models [45], we add “[CLS]” token and “[SEP]” token to segment these sequences. As illustrated in Figure 5, “[CLS]” and “[SEP]” are placed at the beginning and the end of a sequence, respectively. The “[SEP]” token is also used to separate the relational contexts or types of different entities in the entity information sequence. The embeddings of these two special tokens are randomly initialized.

Then, these input sequences, i.e., S^r , S^c , and S^l , are fed into three separate Transformer encoders, namely, Transformer_r , Transformer_c , and Transformer_l , respectively. Since different kinds of information on the path are separately put into different encoders, we only consider the input embeddings as the sum of token embeddings and position embeddings, without the segment embeddings. Moreover, considering the self-attention mechanism in Transformer, instead of designing additional attention for the relational contexts or types within each entity, we treat them as a sequence with the same position embeddings. Hence, the overall semantics of all entities at different positions on the path can be observed, and important relational contexts or entity types can be identified for inference.

The output hidden states of “[CLS]” token of these encoders are taken as relation representation \mathbf{h}^r , relational context representation \mathbf{h}^c , and entity type representation \mathbf{h}^l of path p , respectively, which can be written as

$$\begin{aligned} \mathbf{h}^r &= \text{Transformer}_r(s^r), \\ \mathbf{h}^c &= \text{Transformer}_c(s^c), \\ \mathbf{h}^l &= \text{Transformer}_l(s^l). \end{aligned} \quad (10)$$

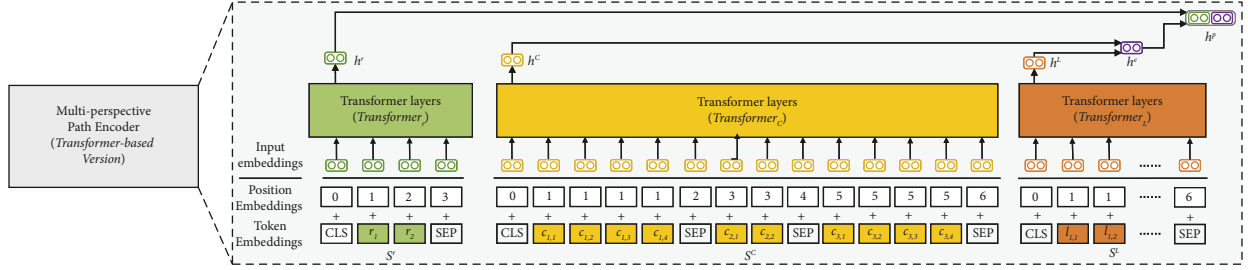


FIGURE 5: The architecture of the Transformer-based multiperspective path encoder (Transformer-MPE) for encoding the example path p in Figure 2. The path is processed into three sequences, i.e., relation input sequence S^r , relational context input sequence S^c and entity type input sequence S^l , which are fed into Transformer_r , Transformer_c , and Transformer_l , respectively. The processing of S^l omitted in the figure is the same as for S^c .

Next, \mathbf{h}^c is summed with \mathbf{h}^l as the entity representation \mathbf{h}^e , which is next concatenated with \mathbf{h}^r to get the path representation \mathbf{h}^p of path p , as in LSTM-based path encoder (equations (8) and (9)). Note that, different from the LSTM-MPE, r_q is not considered in the Transformer-MPE which we will leave as future work.

4.3. Attentive Path Aggregator. The paths between an entity pair are usually extracted randomly, and most of them are useless or even noisy for inference. It is therefore necessary to select important and useful ones among these paths. The attentive path aggregator is applied to weigh the paths and fuse their features. The weight att_n^p of path $p_n \in P_{h,t}$ is determined directly by its path feature \mathbf{h}_n^p , which can be formulated as

$$\text{att}_n^p = f_{\text{att}}^p(\text{ReLU}(f_p(\mathbf{h}_n^p))), \quad (11)$$

where f_{att}^p and f_p are single-layer feed-forward neural networks. Then, the normalized attention weight β_n of p_n is

$$\beta_n = \frac{\exp(\text{att}_n^p)}{\sum_{w=1}^N \exp(\text{att}_w^p)}, \quad (12)$$

where N is the number of paths in $P_{h,t}$. Finally, the pooled representation of all paths in $P_{h,t}$, i.e., $\hat{\mathbf{h}}^p$, is computed as the weighted sum of these path representations:

$$\hat{\mathbf{h}}^p = \sum_{n=1}^N \beta_n * \mathbf{h}_n^p. \quad (13)$$

4.4. Dual Path Encoding. Prior approaches only use feature $\hat{\mathbf{h}}^p$ of paths $P_{h,t}$ to predict whether (h, t) is connected by r_q , i.e., $h \xrightarrow{r_q} t$. However, it overlooks the role of reverse paths for reasoning $t \xrightarrow{r'_q} h$ in assisting the prediction. Although the inverse sequence $\tilde{p} = \langle e_{M+1}, r_M, \dots, e_2, r_1, e_1 \rangle$ can be obtained from the path $p = \langle e_1, r_1, e_2, \dots, r_M, e_{M+1} \rangle$, it is unable to reason r'_q . The reason is that \tilde{p} would break the directionality of relations. For example, take $M = 2$, then in sequence \tilde{p} there is $e_3 \xrightarrow{r_2} e_2 \xrightarrow{r_1} e_1$, i.e., $e_1 \xleftarrow{r_1} e_2 \xleftarrow{r_2} e_3$, while in path p there is $e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} e_3$, which is contradictory. In order to enhance the inference about (h, r_q, t) using

features about (t, r'_q, h) , the dual path encoding module additionally models representations of reverse paths $P'_{h,t}$. The reverse path $p' = \langle e_{M+1}, r'_M, \dots, e_2, r'_1, e_1 \rangle$ maintains the relation directionality and thus can be used to predict (t, r'_q, h) , which means (h, r_q, t) but with different path features than p . This is because the reverse path contains new relation patterns combined by new relations $\{r'_M, \dots, r'_1\}$, which will result in new path features.

This module learns the representation of reverse paths $P'_{h,t}$ in the same way as $P_{h,t}$. That is, for a reverse path p' , the multiperspective path encoder encodes it as the path representation \mathbf{h}'^p . Then, paths in $P'_{h,t}$ are fused into $\hat{\mathbf{h}}'^p$ by the attentive path aggregator. Finally, the dual-path representations $\hat{\mathbf{h}}^p$ and $\hat{\mathbf{h}}'^p$ are utilized by simply applying a two-layer feed-forward neural network to predict the probability $\mathbb{P}(h, r_q, t)$:

$$\mathbb{P}(h, r_q, t) = \sigma\left(f_2\left(\text{ReLU}\left(f_1\left(\left[\hat{\mathbf{h}}^p, \hat{\mathbf{h}}'^p\right]\right)\right)\right)\right), \quad (14)$$

where σ is the sigmoid function.

Note that, encoding reverse paths share the parameters of the multiperspective path encoder with encoding forward ones. Two attentive path aggregators are applied separately to fuse features of paths in $P_{h,t}$ and of reverse paths in $P'_{h,t}$. In LSTM-MPE, both relational context portrait attention and entity type portrait attention for reverse path p' are led by reverse query relation r'_q in equation (2), and the initialization of LSTM_c in relational context sequence encoding and LSTM_l in entity type sequence encoding are guided by relation representation \mathbf{h}^r of reverse path p' in equation (7).

4.5. Training. Let $\mathbb{R}_Q = \{r_1, \dots, r_q, \dots, r_Q\}$ denote the set of query relations. For each task (query relation) r_q , the true triples in the KG with r_q as relation are regarded as positive samples Δ_q^+ while the unobservable triples with r_q as relation are treated as negative samples Δ_q^- . The ground truth y is 1 for positive sample and 0 for negative sample. The parameters of the model for each r_q can be trained end to end by minimizing the loss between predictions and ground truths:

$$L = \sum_{(e_i, r_q, e_k) \in \Delta_q} J(\mathbb{P}(e_i, r_q, e_k), y), \quad (15)$$

Input: KG \mathcal{G} , query relations $\mathbb{R}_Q = \{r_1, \dots, r_q, \dots, r_Q\}$, training examples $\Delta = \{\Delta_1, \dots, \Delta_q, \dots, \Delta_Q\}$, validation examples $\Delta^V = \{\Delta_1^V, \dots, \Delta_q^V, \dots, \Delta_Q^V\}$, paths between the entity pair of each example, types of each entity.
Output: The set of model parameters for each query relation, i.e., $\{\theta_1, \dots, \theta_q, \dots, \theta_Q\}$.

- (1) **for** each query relation r_q in \mathbb{R}_Q **do**
- (2) Initialize model SPR with parameters θ_q ;
- (3) **for** $epoch = 1 \rightarrow \max_epoch_num$ **do**
- (4) **for** $step = 1 \rightarrow \max_step_num$ **do**
- (5) $b \leftarrow$ sample a batch of training examples from $\Delta_q = \{\Delta_q^+, \Delta_q^-\}$;
- (6) **for** (e_i, r_q, e_k) in b **do**
- (7) Compute the prediction probability by equation (14);
- (8) Compute the loss by equation (15);
- (9) Update θ_q by minimizing the batch loss;
- (10) Validate using validation examples Δ_q^V ;

ALGORITHM 1: Training algorithm of SPR.

where Δ_q is the training samples about r_q , i.e., $\Delta_q = \{\Delta_q^+, \Delta_q^-\}$, and $J(\cdot)$ is the binary cross-entropy loss. The training process of SPR is presented in Algorithm 1.

5. Experiments

5.1. Datasets. We evaluate our model using the data released in APR [24] for experiments, which are based on two common KGs datasets: WN18RR [14] and FB15k-237 [28]. The statistics are shown in Table 2. The data contains positive and negative examples, as well as paths extracted by bi-directional breadth first search between the entities in each example for each query relation. Reverse relations have already been added to augment these knowledge graphs for path extraction. The 11 relations in WN18RR are all set as query relations, and 10 relations are sampled as queries out of 237 relations in FB15k-237. The maximum length of a path is 6 for WN18RR and 4 for FB15k-237. The maximum number of paths between each entity pair is 200, which means that if more paths can be extracted, then 200 paths of them are randomly sampled. The type hierarchies of each entity are provided in both datasets. Specifically, the type data of entities for WN18RR is extracted from inherited hypernyms available in WordNet [8], and for FB15K-237, it is those released by Xie et al. [35]. The average and maximum numbers of entity types are 4.6 and 14 for WN18RR, 6.4 and 7 for FB15k-237. For more details, please refer to APR [24]. As for the relational contexts used in the proposed method, we count them for each entity in both KGs, where the average and maximum numbers of relational contexts are 2.6 and 9 for WN18RR, 10.6 and 59 for FB15k-237.

5.2. Experimental Settings. We present the settings according to the different components of the proposed SPR. For LSTM-MPE, its hidden dimension of all the LSTMs (i.e., $LSTM_r$, $LSTM_C$, and $LSTM_L$) is set to 150 for both datasets. The output dimensions of the fully connected layers for encoding relational context sequence, i.e., f_g^C , f_{emb}^C , f_{att}^C , $f_{init_h}^C$, and $f_{init_e}^C$, are set to 50, 50, 1, 150, and 150, respectively, and the same for the fully connected layers that encode entity type sequence. For Transformer-MPE, the number of

TABLE 2: Statistics about the datasets, including KGs, samples of tasks, paths between entity pairs, types of entities, and the relational contexts of entities.

Dataset	WN18RR	FB15k-237
#Entities	40,943	13,545
#Relations	11	237
#Facts	67,360	254,290
#Tasks (#query relations)	11	10
Avg. #train sample/task	35,747	16,596
Avg. #valid sample/task	8,936	4,152
Avg. #test sample/task	11,173	5,188
Avg. #paths/sample	88.6	154.8
Max path length	6	4
Avg. Path length	5.3	3.5
#Entity types	8,092	1,029
Avg. #types/entity	4.6	6.4
Max number of types	14	7
#Relational contexts	11	237
Avg. #relational contexts/entity	2.6	10.6
Max number of relational contexts	9	59

Transformer layers and the number of heads per layer of $Transformer_r$, $Transformer_C$, and $Transformer_L$ are all set as 3 and 2 on both datasets. The dimension of the feed-forward network in these Transformer layers is set to 150. For attentive path aggregator and dual path encoding, the output dimensions of f_p , f_{att}^p , and f_2 are set to 100, 1, and 1, and the output dimension of f_1 is set to be the same as $\hat{\mathbf{h}}_p^p$. Since SPR can be built on different MPE, we denote the overall model based on LSTM-MPE and Transformer-MPE by SPR_LSTM and SPR_Transformer, respectively.

For both architectures, the embedding matrix of relations is randomly initialized for both datasets. The embedding matrix of entity types for WN18RR is initialized with vectors provided by APR [24], which is mapped from a pretrained Google News word2vec model. The embedding matrix of entity types for FB15k-237 is randomly initialized. As for relational contexts, another random initialized embedding matrix different from relations is applied. Let D_r , D_C , and D_L be the embedding dimensions of relations, relational contexts, and entity types, respectively, whose

TABLE 3: Hyperparameter settings on two datasets.

	D_r	D_C	D_L	K_C	K_L	SPR_LSTM		SPR_transformer	
						bs	lr	bs	lr
WN18RR	150	300	300	2	14	16	$e-3$	4	$e-5$
FB15k-237	150	50	50	12	7	16	$e-3$	8	$e-4$

D_r , embedding dimension of relations; D_C , embedding dimension of relational contexts; D_L , embedding dimension of entity types; K_C , the maximum number of relational contexts used for each entity; K_L , the maximum number of types used for each entity; bs , batch size; lr , learning rate; SPR, structure enhanced path reasoning; SPR_Transformer, Transformer-based SPR; SPR_LSTM, LSTM-based SPR.

values are listed in Table 3. The embedding matrices of position in SPR_Transformer are randomly initialized, with the same dimensions as the corresponding input tokens (relations, relational contexts, or entity types). Table 3 also reports the maximum number of relational contexts and types used for each entity in the model, i.e., K_C and K_L , as well as the batch size bs and learning rate lr . Note that, we set K_L to the maximum number of types of the entity, i.e., we use all the types contained in it. However, for relational contexts, the number of them for some entities is so large, e.g., up to 59 in FB15k-237, so that we use only K_C instead of all of them. Due to the limitation of computational resources, SPR_Transformer is set to a smaller batch size than SPR_LSTM, and the learning rate is reduced accordingly. We employ the self-adaptive optimization method Adam [51] for all trainings, and the model is trained fully to 30 epochs, where the best one is chosen based on validation sets. All experiments are conducted with one Tesla V100 GPU.

5.3. Evaluation Metrics. We evaluate the performance of SPR for implementing KGC on two tasks, namely, fact prediction and relation prediction. For fact prediction task, where the system needs to, given a missing triple (h, r, t) , identify whether it is true or not. We utilize the classical mean average precision (MAP) as the evaluation metric following previous path-based models. It measures the model performance on all categories (here on all query relations), which is the mean of the average precision (AP) for each category. As for relation prediction task, the goal is to predict the missing relation between the entity pair (h, t) . We adopt typical ranking metrics including mean rank (MR), mean reciprocal rank (MRR), and Hits@n.

5.4. Baselines. We select several state-of-the-art models with released code as baselines:

- (1) Embedding-based models: TransE [12], PTransE [37], RotatE [30], HAKE [31], PairRE [52], and PathCon [26]. Among them, paths are additionally considered in PTranse and PathCon. We test these embedding-based models by their released codes with corresponding optimal parameters, and the hidden dimension is set to 500 on WN18RR and 1000 on FB15k-237. Following Xiong et al. [20], test triples with the same relation are clustered together to calculate the AP and finally obtain the MAP. Note that, since PathCon focuses on relation prediction

task by calculating the probability distribution of relations given an entity pair, we take the probability of the query relation for an entity pair as its triple score to calculate MAP for fact prediction task.

- (2) Path-based models: PRA [19], SFE [41], Path-RNN [22], Chains [23], and APR [24], where their details are described as follows:
 - (a) PRA [19]: adopts the probabilities of performing random walks between entities following relation sequences as path features.
 - (b) SFE [41]: improves PRA [19] by regarding relation paths as binary features without calculating the random walk probabilities.
 - (c) Path-RNN [22]: encodes the relation sequence on the path through RNN as path features for prediction.
 - (d) Chains^A [23]: encodes relations and entity types on the path by RNN to jointly create the path representation and consider multiple paths for inference, where the features of multiple types of an entity are averaged.
 - (e) Chains^B [23]: goes a step further than Chains^A by combining the features of entities themselves to obtain the path representation, in addition to the relations and the averaged entity type features.
 - (f) APR [24]: utilizes two LSTMs to separately encode relations and weighted entity type hierarchies on the path and combines them into path representations which are aggregated by an attention mechanism.

The first two approaches, i.e., PRA [19] and SFE [41], are tested with code released by Gardner and Mitchell [41]; Path-RNN [22], Chains^A [23], and Chains^B [23] are tested with code released by Das et al. [23], which use LogSumExp method proposed in Chain [23] to pool scores of paths; APR [24] are tested with code released by Liu et al. [24]. We use their fact prediction results reported by Liu et al. [24].

6. Results and Analysis

In this section, we first analyze the main results of SPR against the baselines for fact prediction task in Section 6.1. Then, we additionally build a unified version of SPR and report its results for fact prediction task in Section 6.2. The results for relation prediction task are presented in Section 6.3. To explore the effects of using entity, entity relational contexts, entity types, and dual path encoding, we design several variants of SPR_LSTM in Section 6.4. Next, the

contribution of each component is described in Section 6.5. We also illustrate the influences of different strategies for choosing relational contexts, as well as the effects of the value of K_C in Section 6.6. At last, to better understand the effectiveness of the proposed approach, several case studies are shown in Section 6.7.

6.1. Main Results of Fact Prediction. The fact prediction results of SPR against various baselines are shown in Table 4. We can observe that both LSTM-based and Transformer-based SPR outperforms all the baselines in both datasets. (1) Compared to path-based models, the results suggest the effectiveness of SPR in enhancing path features by utilizing relational contexts and dual path encoding, where SPR_LSTM achieves 3.63% and 4.52% improvements over the best on WN18RR and FB15k-237, respectively. SPR_Transformer is not as good as SPR_LSTM, probably due to the fact that the information of the query relation r_q is not fused in MPE-Transformer, which can refer to Section 6.5 for more analysis. Insufficient training data may also be the reason that hinders the great power of Transformer, since each query relation has its own training data and model parameters (more detailed analysis can be found in Section 6.2). (2) The superiority of SPR is displayed on WN18RR with a MAP higher than 87%, which beats the embedding-based approach by a large margin where the MAP is no higher than 55%. This is probably because SPR does not use the embedding of entity as most embedding-based methods do, but uses relational contexts and types to represent the entity. When there are a huge number of entities in a KG, the use of entity embedding may introduce noise and lead to performance degradation as discussed in Section 6.4. In addition, it may also benefit from the ability of SPR to efficiently utilize information on the path for prediction, where the results in Section 6.4 show that different kinds of information on paths play essential roles in performance improvement. (3) On FB15k-237, even though the recent embedding-based method HAKE with 58.11% MAP shows slightly better performance than the path-based methods APR with 57.35% MAP, SPR beats both of them. However, compared with the embedding-based approach, the effectiveness of SPR on FB15k-237 is not as obvious as on WN18RR, which may be caused by the higher number of relations on FB15k-237. This will lead to a great number of combinations of relational patterns on the paths, thus making it more difficult for SPR to learn knowledge from the paths and make predictions. (4) SPR beats PathCon on both datasets, even though it also utilizes relational contexts and paths between entity pairs, indicating the validity of SPR by treating paths as sequences and identifying different contributions of multiple relational contexts for each entity.

6.2. Results of Unified Version

6.2.1. SPR_Unified. Recall that SPR treats different query relations as different tasks for training in Section 4.5. In order to obtain a model that takes all relations as one task for inference, we modify the proposed SPR as a unified

TABLE 4: MAP% of fact prediction results about embedding-based models (the first portion), path-based models (the second portion), and the proposed SPR method (the third portion, including SPR_Transformer and SPR_LSTM).

Models	Use paths	WN18RR	FB15k-237
TransE [12]	—	28.59	57.68
PTransE [37]	✓	39.52	58.04
RotatE [30]	—	50.58	57.56
HAKE [31]	—	53.84	58.11
PairRE [52]	—	44.43	57.55
PathCon [26]	✓	52.48	41.24
PRA [19]	✓	38.85	34.33
SFE [41]	✓	30.75	36.79
Path-RNN [22]	✓	67.16	45.64
Chains ^A [23]	✓	50.82	51.23
Chains ^B [23]	✓	51.08	52.17
APR [24]	✓	84.91	57.35
SPR_Transformer	✓	87.53	61.01
SPR_LSTM	✓	88.54	61.87

Note. The best result is shown in bold, and the previous best is in italics. MAP, mean average precision; SPR, structure enhanced path reasoning; SPR_Transformer, Transformer-based SPR; SPR_LSTM, LSTM-based SPR.

model, namely, SPR_Unified. Given an entity pair (h, t) , instead of predicting the probability of a fact (h, r_q, t) consisting of a certain query relation r_q , SPR_Unified predicts a vector $\hat{\mathbf{r}}$ representing the relation features between h and t , so that the score of (h, r_q, t) can be generated by matching $\hat{\mathbf{r}}$ against the embedding of r_q via dot product. Specifically, it replaces the prediction in equation (14) by the following formulas:

$$\hat{\mathbf{r}} = f_2\left(\text{ReLU}\left(f_1\left(\left[\hat{\mathbf{h}}^p, \hat{\mathbf{h}}'^p\right]\right)\right)\right), \quad (16)$$

$$\mathbb{P}(h, r_q, t) = \sigma(\hat{\mathbf{r}}^\top \mathbf{r}_q), \quad (17)$$

where the output dimension of the fully connected layer f_2 is changed to be the same as the dimension of relation embeddings, \mathbf{r}_q is the relation embedding of query relation r_q , and σ is the sigmoid function.

6.2.2. Training. For training SPR_Unified, positive samples Δ^{u+} are composed of all the positive instances corresponding to each query relations in SPR, i.e., $\Delta^{u+} = \{\Delta_q^+ | r_q \in \mathbb{R}_Q\}$. And, negative samples Δ^{u-} are obtained by corrupting the relation of each positive triple in Δ^{u+} . That is, for a positive triple (h, r_q, t) in Δ^{u+} , we obtain its negative samples $\Delta_{(h, r_q, t)}^{u-}$ by corrupting its relation r_q by all relations that do not exist between h and t , i.e., $\Delta_{(h, r_q, t)}^{u-} = \{(h, r, t) | (h, r, t) \notin \mathcal{F}, r \in \mathcal{R}\}$. The ground truth y is 1 for positive sample and 0 for negative one. SPR_Unified is trained by minimizing the loss between predictions and ground truths over the training samples Δ^u :

$$L = \sum_{(e_i, r_q, e_k) \in \Delta^u} J(\mathbb{P}(e_i, r_q, e_k), y), \quad (18)$$

where $\Delta^u = \{\Delta^{u+}, \Delta^{u-}\}$, and $J(\cdot)$ is the binary cross-entropy loss. The training process is shown in Algorithm 2.

Input: KG \mathcal{G} , query relations $\mathbb{R}_Q = \{r_1, \dots, r_q, \dots, r_Q\}$, training examples Δ^u for all query relations, validation examples Δ^{uV} for all query relations, paths between the entity pair of each example, types of each entity.

Output: The model parameters for all query relations, i.e., θ .

- (1) Initialize model SPR_Unified with parameters θ ;
- (2) **for** $epoch = 1 \rightarrow \max_epoch_num$ **do**
- (3) **for** $step = 1 \rightarrow \max_step_num$ **do**
- (4) $b \leftarrow$ sample a batch of training examples from $\Delta^u = \{\Delta^{u+}, \Delta^{u-}\}$;
- (5) **for** (e_i, r_q, e_k) in b **do**
- (6) Compute the prediction probability by equation (17);
- (7) Compute the loss by equation (18);
- (8) Update θ by minimizing the batch loss;
- (9) Validate using validation examples Δ^{uV} ;

ALGORITHM 2: Training algorithm of SPR_Unified.

6.2.3. *Evaluation.* Note that, query relation r_q is encoded in SPR_LSTM, so we remove the guide of r_q in equation (2) to get its unified version, noted as SPR_Unified_LSTM. And we denote SPR_Unified_Transformer as the unified SPR_Transformer. For comparison, we also modify APR [24] to APR_Unified in the same way. We evaluate these unified models for fact prediction and relation prediction task on WN18RR. It is worth noting that we do not evaluate on FB15k-237, since this dataset released by Liu et al. [24] contains only samples of 10 query relations out of 237 relations and cannot perform complete training on the unified model for all relations.

Results for fact prediction are listed in Table 5, we find that: (1) the unified version SPR_Unified_Transformer achieves the best result of 81.32% MAP, which gains an absolute improvement of 27.48% and 6.7% compared to the previous optimal embedding-based model HAKE and path-based model APR_Unified, respectively. (2) Comparing SPR_Unified_Transformer with SPR_Unified_LSTM, the Transformer-based architecture is clearly superior to the LSTM-based one with a 9.49% higher MAP, which illustrates the ability of Transformer to handle large-scale and complex data. (3) Compared to APR_Unified, which is also a path-based method built on LSTM, SPR_Unified_LSTM shows a performance degradation. This can be explained by the inability of LSTM to handle complex data and learn useful information when considering relational contexts and dual path encoding method in the unified version, resulting in a MAP decline of SPR_Unified_LSTM. (4) The performances of unified versions, including APR_Unified and SPR_Unified, are reduced compared to the corresponding original models. This is because in the unified version, there is only one set of model parameters to be trained, rather than one set for each query relation. That is, the total number of parameters of the unified model is reduced compared to the original one, while the amount of data for training it is increased. As a result, the unified version fails to learn and remember all the knowledge, leading to reduced performance. However, the decline in our SPR_Unified_Transformer is the least, with 6.21% lower than SPR_Transformer, while the drop of APR_Unified is 10.29% lower than APR [24]. This once

TABLE 5: Fact prediction results on WN18RR of SPR_Unified, including SPR_Unified_LSTM and SPR_Unified_Transformer.

Models	MAP%
PathCon [26]	52.48
HAKE [31]	53.84
APR_Unified	74.62
SPR_Unified_LSTM	71.83
SPR_Unified_Transformer	81.32

Note. The best result is shown in bold, and the previous best is in italics. MAP, mean average precision; SPR, structure enhanced path reasoning; SPR_Unified, unified version of SPR; APR_Unified, unified version of APR [24]; SPR_Unified_LSTM, unified version of LSTM-based SPR; SPR_Unified_Transformer, unified version of Transformer-based SPR.

again indicates the ability of the Transformer-based architecture to deal with complex and massive data.

6.3. *Main Results of Relation Prediction.* The relation prediction results of SPR and its unified version on WN18RR are listed in Table 6. As mentioned above, the dataset FB15k-237 released by Liu et al. [24] contains only samples for predicting 10 query relations, which is not suitable for relation prediction task of predicting all relations, so we only evaluate on WN18RR.

We can conclude that: (1) SPR_Unified achieves the best results in several metrics such as MR, MRR, hits@1, and hits@5. Although the hits@3 of SPR_Unified is slightly worse than APR_Unified by about 0.1% to 0.2%, it obtains the best MRR and MR since the hits@1 is at least 0.5% higher. (2) SPR_Unified_LSTM and SPR_Unified_Transformer obtains comparable performance, with the former having a higher hits@3 and the latter having a higher hits@1. However, compared to their fact prediction performance in Table 5, SPR_Unified_Transformer outperforms SPR_Unified_LSTM by a large margin, indicating that the MAP for fact prediction task is a more rigorous evaluation metric. (3) It is not surprising that the unified version SPR_Unified shows better performance than SPR for relation prediction task, while the opposite conclusion is reached for fact prediction. It is mainly because SPR_Unified learns probability distributions over all relations in one model, while SPR gets different probability distributions for each query

TABLE 6: Relation prediction results on WN18RR of the proposed SPR and SPR_Unified.

Models	MR↓	MRR↑	hits@ 1↑	hits@ 3↑	hits@ 5↑
TransE [12]	2.20	0.639	0.385	0.850	0.943
RotatE [30]	1.48	0.903	0.857	0.934	0.960
HAKE [31]	1.56	0.887	0.832	0.935	0.953
PathCon [26]	1.50	0.867	0.786	0.942	0.975
APR [24]	1.32	0.937	0.910	0.953	0.969
APR_Unified	1.03	<i>0.989</i>	0.981	0.999	0.999
SPR_LSTM	<i>1.15</i>	0.958	0.935	0.977	0.992
SPR_Unified_LSTM	1.03	0.992	<i>0.986</i>	<i>0.998</i>	0.999
SPR_Transformer	1.23	0.948	0.923	0.962	0.983
SPR_Unified_Transformer	1.03	0.992	0.987	0.997	0.999

Note. The best result is shown in bold, and the second best is in italics. MR, mean rank; MRR, mean reciprocal rank; APR_Unified, unified version of APR [24]; SPR, structure enhanced path reasoning; SPR_Unified, unified version of SPR; SPR_LSTM, LSTM-based SPR; SPR_Unified_LSTM, unified version of LSTM-based SPR; SPR_Transformer, Transformer-based SPR; SPR_Unified_Transformer, unified version of Transformer-based SPR.

relation which has separate model parameters. In SPR, a query relation has a high triple score does not mean that the query relation will rank top among all relations.

6.4. Variants Analysis. Several variants of SPR_LSTM are explored for comparison, to investigate the effects of using entity, entity relational contexts and entity types, as well as dual path encoding: (1) Path_LSTM^r does not encode entity information, which suggests that the path representation in equation (9) is obtained only from the relation representation, i.e., $\mathbf{h}^p = \mathbf{h}^r$, $\mathbf{h}^{p'} = \mathbf{h}^{r'}$; (2) Path_LSTM^e encodes entity information by directly encoding entity sequences (p^e and p'^e), i.e., it uses entity embeddings as input to the LSTM in entity encoder, rather than encoding sequences of relational contexts and entity types; (3) Path_BiLSTM does not apply the dual path encoding method, i.e., it uses only $P_{h,t}$ but conducts bidirectional encoding, where the backward encoding of a path p is equivalent to encoding its inverse sequence $\tilde{p} = \langle e_{M+1}, r_M, \dots, e_2, r_1, e_1 \rangle$, rather than its reverse path p' . Specifically, Path_BiLSTM encodes relation sequence p^r by Bi-LSTM to obtain \mathbf{h}^r and $\mathbf{h}^{r'}$ and encodes the sequences of the relational contexts and entity types of p^e by LSTM_C and LSTM_L forward and backward, to obtain \mathbf{h}^C , \mathbf{h}^L , $\mathbf{h}^{C'}$, and $\mathbf{h}^{L'}$.

Table 7 shows the results of SPR_LSTM against its variants, from which we can find that: (1) learning path representation with the dual path encoding method by taking relational contexts, entity types as entity information, achieves the best results. (2) Directly encoding entities may not bring benefits. Compared to Path_LSTM^r, although the results of Path_LSTM^e increase in FB15k-237 (from 51.88% to 55.32%), they deteriorate in WN18RR (from 82.64% to 78.21%), which confirms that entity embedding may lead to noise. This may be due to the larger number of entities and the smaller number of facts in WN18RR, which highlights the drawbacks of embedding-based methods that learn entity representations directly. (3) Compared with the

Path_BiLSTM, which encodes the relation sequence and entity information sequence directly forward and backward, the results of SPR_LSTM gain 1.69% and 2.81% on WN18RR and FB15k-237 respectively, indicating that the dual path encoding method in SPR_LSTM works better than encoding forward paths and their inverse sequences in Path_BiLSTM. It suggests the indispensable contribution of the reverse paths that comprise new relation patterns.

6.5. Ablation Study. We conduct ablation studies to investigate the effectiveness of different components of SPR: (1)-DualPath: removing the dual path encoding method, which means we only encode forward paths $P_{h,t}$ to get $\tilde{\mathbf{h}}^p$ and use it to calculate $\mathbb{P}(h, r_q, t)$ in equation (14); (2)-Rel_ctx: removing entity relational contexts for paths $P_{h,t}$ and $P'_{h,t}$, which implies that the entity representation in equation (8) is obtained only from the entity type representation, i.e., $\mathbf{h}^e = \mathbf{h}^L$ and $\mathbf{h}^{e'} = \mathbf{h}^{L'}$; (3)-Type: removing entity types for paths $P_{h,t}$ and $P'_{h,t}$, where the entity representation in equation (8) is obtained only from the relational context representation, i.e., $\mathbf{h}^e = \mathbf{h}^C$ and $\mathbf{h}^{e'} = \mathbf{h}^{C'}$; (4)- $\mathbf{r}_q, \mathbf{r}'_q$: removing the guide of query relation in relational context portrait attention and entity type portrait attention in SPR_LSTM, i.e., the attention for relational contexts and types is only guided by historical path feature in equation (2).

From the results shown in Tables 8 and 9, it can be observed that: (1) removing the dual path encoding (-DualPath) results in performance degradation in both datasets, which demonstrates that the learned features of reverse paths are different from those of forward paths and can help predict the query relation. (2) Using only entity relational contexts (-Type) or entity types (-Rel_ctx) degrades performance. This proves that these two kinds of information are complementary to the semantic portrayal of entities, thus together they can enhance the representation of entities. Besides, the performance drop is more pronounced when relational contexts are not used, which indicates its greater contribution than types. (3) For SPR_LSTM, the results decline when without the guide of query relation ($-r_q, r'_q$), indicating that query relation can help to spotlight essential relational contexts and types of the entity. Compared to SPR_Transformer, neither of which uses information of query relation, its performance is slightly lower (0.05%) on FB15k-237, suggesting that the guidance of query relation is more important on FB15k-237. (4) For SPR_LSTM, the removal of dual path encoding (-DualPath) has the most noticeable decreases, which is 3.91% and 4.62% in WN18RR and FB15k-237, respectively. For SPR_Transformer, the most influential setting is the removal of relational contexts (-Rel_ctx), with drops of 1.73% and 3.83% in WN18RR and FB15k-237, respectively. These imply that relational contexts and dual path encoding are critical for improving the ability to path reasoning.

6.6. Influence of Entity Relational Contexts. We examine the impact of relational contexts in this section. First, three strategies are set to rank the relational context of an entity based on how often it appears in the facts with the entity as

TABLE 7: MAP% of fact prediction results against the variants of SPR_LSTM.

Models	Path info	Path encoding	Entity info	WN18RR	FB15k-237
Path_LSTM ^r	$P_{h,t}, P'_{h,t}$	Dual	—	82.64	51.88
Path_LSTM ^e	$P_{h,t}, P'_{h,t}$	Dual	Entity	78.21	55.32
Path_BiLSTM	$P_{h,t}$	Bi-direction	Relational context, entity type	86.85	59.06
SPR_LSTM	$P_{h,t}, P'_{h,t}$	Dual	Relational context, entity type	88.54	61.87

Note. The best result is shown in bold. MAP, mean average precision; info, information; SPR, structure enhanced path reasoning; SPR_LSTM, LSTM-based SPR; $P_{h,t}$, paths from head entity h to tail entity t ; and $P'_{h,t}$, reverse paths of $P_{h,t}$.

TABLE 8: Ablation study of SPR_LSTM for fact prediction task (MAP%).

Models	WN18RR	FB15k-237
SPR_LSTM	88.54	61.87
-DualPath	84.63	57.61
-Rel_ctx	86.03	58.09
-Type	86.62	58.99
- r_q, r'_q	87.69	60.96

MAP, mean average precision; SPR, structure enhanced path reasoning; SPR_LSTM, LSTM-based SPR; -DualPath, removing dual path encoding; -Rel_ctx: removing relational contexts; -Type: removing entity types; - r_q, r'_q : removing the guide of query relation.

TABLE 9: Ablation study of SPR_Transformer for fact prediction task (MAP%).

Models	WN18RR	FB15k-237
SPR_Transformer	87.53	61.01
-DualPath	85.86	58.66
-Rel_ctx	85.80	57.18
-Type	86.02	57.23

MAP, mean average precision; SPR, structure enhanced path reasoning; SPR_Transformer, Transformer-based SPR; -DualPath, removing dual path encoding; -Rel_ctx: removing relational contexts; -Type: removing entity types.

the head: (1) frequent_sort: sorts the relational contexts in descending order of frequency; (2) infrequent_sort: sorts the relational contexts in ascending order of frequency; (3) random_sort: sorts the relational contexts randomly. Second, we investigate the influence of K_C , which is the maximum number of relational contexts used for each entity. That is, if there are more than K_C relational contexts of an entity, then only the top K_C of them are used.

The results based on SPR_LSTM are displayed in Figure 6. We can notice that: (1) although the performance on the two datasets displays different trends, the results are relatively high when K_C is close to the average number of relational contexts per entity. (2) On WN18RR, regardless of the value of K_C , the performance is approximately the same for all three sorting scenarios. This may be because the average number and the kinds of relational contexts are both small, i.e., 2.6 and 11, respectively. Thus, when K_C is smaller than the average number of 2.6, most entities select the same relational contexts under all three strategies. Even when K_C is greater than 2.6, the number of combinations of entity relational contexts on the path is not large, therefore the strategies make little difference. (3) On FB15k-237, the prediction performance gets the best apparently with the

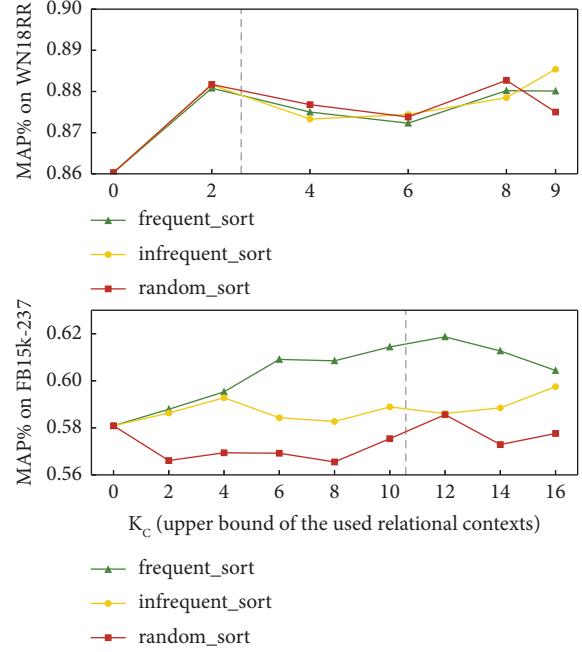


FIGURE 6: Influences of the sorting strategies for entity relational contexts on the proposed SPR_LSTM for fact prediction task. The gray dashed vertical line shows the average number of relational contexts per entity in the corresponding KG.

frequent_sort, while it shows erratic performance as K_C increases under the infrequent_sort and random_sort settings. This indicates that frequently occurring relational contexts are more representative of the semantics often expressed by the entity. While using infrequent relational contexts would be inconsistent with the semantics the entity intends to represent in most cases.

6.7. Case Study. We analyze the weights of relational contexts and types of entities on the paths. Figure 7 gives a path of a positive example in FB15K-237 learned by SPR_LSTM. We can observe that new knowledge can be learned from relational contexts, so as to enrich the semantics of entities. For instance, most of the learned highlight types of entity RazzieAward shown in the first column are meaningless pad rather than meaningful award category. While the relational contexts nomineeWorkOfAward and winningWorkOfAward of this entity RazzieAward are given higher weights, which fills a gap in the entity information learned from types. Moreover, relational contexts can capture what kind of semantics an entity is concerned with on a path. As the

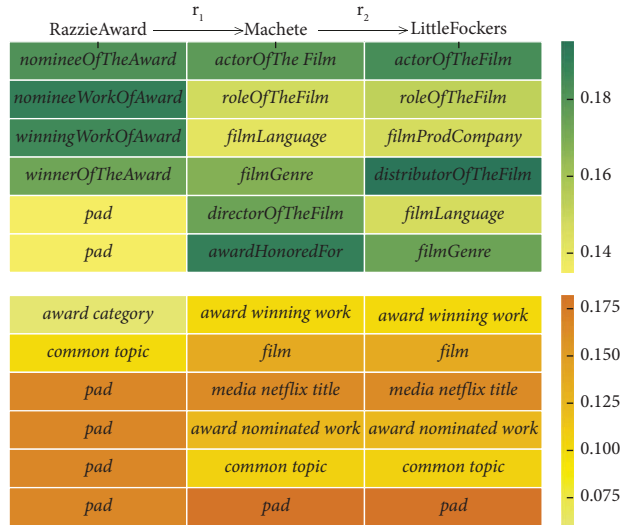


FIGURE 7: Attention weights for relational contexts (top) and types (bottom) of each entity on path \langle RazzieAward, nomineeWorkOfAward (r_1), Machete, nominatedFor (r_2), LittleFockers \rangle for predicting (RazzieAward, nomineeWorkOfAward, LittleFockers) in FB15K-237. Each column shows the top six relational contexts and types with the highest weights, which is listed from top to bottom by frequency. If an entity has less than six relationship contexts or types, it is filled in with “pad.” Note that, we abbreviate the name of relational contexts.

TABLE 10: Examples of important relational contexts listed for head entities and tail entities of several query relations in FB15k-237, where “ r ” refers to the reverse relation of r . Note that, we abbreviated the name of relational contexts.

Query relations	Entity pairs	Important relational contexts
personProfession	Head entity: person Tail entity: profession	athleteSportTeam, personProfession, filmDesignby', musicMemberRole jobtitleCompany, filmCrewRole', personProfession, professionSpecialized'
teamOfSportsPosition	Head entity: position Tail entity: sport team	icehockeyPos', basketballPos', baseballPos', professionSpecialized soccerPos, footballPos, sportsDraft, basketballTournament
filmGenre	Head entity: film Tail entity: genre	filmGenre, filmCostumeDesignby, directorOfFilm', dataSplitto' filmGenre, dataSplitto', tvGenre', filmOfTheSubject

TABLE 11: Prediction probability of positive samples with only one path available between the entity pair for inferring query relation hasPart in WN18RR, learned by Path_LSTM^t and SPR_LSTM, respectively.

Missing facts and the forward path between the entity pair	Path_LSTM ^t	SPR_LSTM
Fact: (custard apple tree, hasPart, custard apple) Path: custard apple tree $\xrightarrow{\text{hypernym}'}$ soursop tree $\xrightarrow{\text{haspart}}$ soursop $\xrightarrow{\text{hypernym}}$ custard apple	0.22	0.88
Fact: (open fireplace, hasPart, mantelpiece) Path: open fireplace $\xrightarrow{\text{haspart}}$ hearth $\xrightarrow{\text{hypernym}}$ home $\xrightarrow{\text{haspart}}$ dining room $\xrightarrow{\text{haspart}}$ sideboard $\xrightarrow{\text{haspart}}$ shelf $\xrightarrow{\text{hypernym}'}$ mantelpiece	0.35	0.97

SPR, structure enhanced path reasoning; SPR_LSTM, LSTM-based SPR; Path_LSTM^t, variant of SPR_LSTM.

entity Machete shown in the second column, its relational context awardHonoredFor which is about award is emphasized for inferring the query relation nomineeWorkOfAward, while filmLanguage is neglected.

We also count the important relational contexts learned for head entities and tail entities of several query relations, where the results are in Table 10. As can be seen, most of the

emphasized relational contexts are logically meaningful for predicting the query relation. For example, when inferring facts with query relation personProfession, i.e., predicting the profession (tail entity) of a person (head entity), athleteSportTeam and filmDesignby' are highly valued, which can infer professions like “athlete” and “actor,” respectively. However, relational contexts such as nationality and gender,

which are owned by most entities in the KG, are ignored, suggesting that the model can capture important relational contexts associated with the query relation.

In addition, Table 11 shows the prediction probability for inferring query relation *hasPart*, where only one forward path is available for the listed positive examples. The reverse path, and the relational contexts and types of entities are not displayed. Model Path_LSTM^t is a variant that removes dual path encoding, entity relational contexts, and query relation in entity information attention from SPR_LSTM, which means it only encodes relations and entity types of forward paths as the path representations. From the comparison with Path_LSTM^t, it is clear that SPR_LSTM predicts correctly, which demonstrates the contribution of relational contexts and dual path encoding.

7. Conclusion

In this paper, we propose a path-based framework named SPR for knowledge graph completion by enhancing the path representation through the structure information of each entity and each path, i.e., relational context and reverse path, respectively. We utilize relational contexts of entities to obtain a reliable path representation that captures accurate and sufficient entity semantics, where different weights of relational contexts are taken into account. Moreover, a dual path encoding method is used to enrich the path representation by capturing new path patterns contained in reverse paths. Different architectures based on LSTM and Transformer are designed to encode the information on the path, including relations, entity relational contexts, and entity types. Experimental results on fact prediction task and relation prediction task show that SPR outperforms the state-of-the-art models. Quantitative and qualitative experiments demonstrate the effectiveness of enhancing path reasoning through utilizing entity relational contexts and dual path encoding method.

It will be part of our future work to focus on designing a Transformer-based architecture, which fuses the query relation to guide the important selection of entity relational contexts and entity types. In addition, leveraging textual descriptions encoded by pretrained language models to complement the semantics of entities may be beneficial for path reasoning. Therefore, there is a potential for further additional research on how to integrate the textual semantic information with structure information on the path for final prediction.

Data Availability

The datasets supporting this research are from previously reported studies, which are open-sourced and have been cited in the paper. The codes used to support the findings of this study are open-sourced at <https://github.com/wylResearch/SPR>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Yilin Wang and Zhen Huang are equally contributing authors.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 62006243).

References

- [1] S. Amit, "Introducing the knowledge graph: things, not strings," 2012, <https://www.blog.google/products/search/introducing-knowledge-graph-things-not/>.%202020-11-13.
- [2] R. Qian, "Understand your world with bing," 2013, <https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/>.
- [3] Y. Chen, L. Wu, and J. Mohammed, "Zaki. Bidirectional attentive memory networks for question answering over knowledge bases," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 2913–2923, Association for Computational Linguistics, Minneapolis, Minnesota, June 2019, <https://aclanthology.org/N19-1299>.
- [4] A. Saxena, A. Tripathi, and P. Talukdar, "Improving multi-hop question answering over knowledge graphs using knowledge base embeddings," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, July 2020, <https://aclanthology.org/2020.acl-main.412>.
- [5] Q. Guo, S. Cao, and Z. Yi, "A medical question answering system using large language models and knowledge graphs," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 8548–8564, 2022, <https://onlinelibrary.wiley.com/doi/abs/10.1002/int.22955>.
- [6] E. Palumbo, G. Rizzo, and R. Troncy, "Entity2rec: learning user-item relatedness from knowledge graphs for top-n item recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys '17*, pp. 32–36, New York, NY, USA, August 2017.
- [7] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, Honolulu Hawaii USA, January 2019.
- [8] G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, nov 1995.
- [9] S. . Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: a nucleus for a web of open data," in *Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, M. Peter, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, Eds., pp. 722–735, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007*.
- [10] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. T. Freebase, "A collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of*

- Data*, SIGMOD '08, pp. 1247–1250, New York, NY, USA, June 2008.
- [11] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin, “Knowledge base completion via search-based question answering,” in *Proceedings of the 23rd International Conference on World Wide Web, WWW '14*, pp. 515–526, New York, NY, USA, April 2014.
- [12] B. Antoine, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Proceedings of the 26th International Conference on Neural Information Processing Systems NIPS'13*, vol. 2, pp. 2787–2795, Red Hook, NY, USA, December 2013.
- [13] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, “Complex embeddings for simple link prediction,” in *Proceedings of the 33rd International Conference on International Conference on Machine Learning ICML'16*, vol. 48, pp. 2071–2080, New York NY USA, June 2016.
- [14] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, “Convolutional 2d knowledge graph embeddings,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, New Orleans, Louisiana, USA, February 2018.
- [15] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” in *The Semantic Web*, A. Gangemi, R. Navigli, M.-E. Vidal et al., Eds., Springer International Publishing, Berlin, Germany, 2018.
- [16] L. Antonio Galárraga, C. Teflioudi, K. Hose, and F. Suchanek, “Amie: association rule mining under incomplete evidence in ontological knowledge bases,” in *Proceedings of the 22nd International Conference on World Wide Web, WWW '13*, pp. 413–422, New York, NY, USA, May 2013.
- [17] Y. Fan, Z. Yang, and W. W. Cohen, “Differentiable learning of logical rules for knowledge base reasoning,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, pp. 2316–2325, Red Hook, NY, USA, December 2017.
- [18] S. Ali, M. Armandpour, P. Ding, and D. Z. Wang, “Drum: end-to-end differentiable rule mining on knowledge graphs,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc, Red Hook, NY, USA, 2019, <https://proceedings.neurips.cc/paper/2019/file/0c72cb7ee1512f800abe27823a792d03-Paper.pdf>.
- [19] L. Ni, T. Mitchell, and W. W. Cohen, “Random walk inference and learning in a large scale knowledge base,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11*, pp. 529–539, Edinburgh, Scotland, UK, July 2011.
- [20] W. Xiong, T. Hoang, and 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*, pp. 564–573, <https://aclanthology.org/D17-1060>, Copenhagen, Denmark, September 2017.
- [21] L. Ni, E. Minkov, and W. Cohen, “Learning relational features with backward random walks,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 666–675, Beijing, China, July 2015, <https://aclanthology.org/P15-1065>.
- [22] A. Neelakantan, B. Roth, and A. McCallum, “Compositional vector space models for knowledge base completion,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 1, pp. 156–166, Beijing, China, July 2015.
- [23] R. Das, A. Neelakantan, D. Belanger, and A. McCallum, “Chains of reasoning over entities, relations, and text using recurrent neural networks,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 1, pp. 132–141, April 2017, <https://aclanthology.org/E17-1013>.
- [24] W. Liu, A. Daruna, Z. Kira, and S. Chernova, “Path ranking with attention to type hierarchies,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 03, pp. 2893–2900.
- [25] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [26] H. Wang, H. Ren, and J. Leskovec, “Relational message passing for knowledge graph completion,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1697–1707, Singapore, August 2021.
- [27] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, pp. 6000–6010, Red Hook, NY, USA, December 2017.
- [28] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, “Representing text for joint embedding of text and knowledge bases,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, September 2015, <https://aclanthology.org/D15-1174>.
- [29] Z. Wang, J. Zhang, J. Feng, and Z. Chen, “Knowledge graph embedding by translating on hyperplanes,” in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI'14*, pp. 1112–1119, Québec, Canada, July 2014.
- [30] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, “Rotate: knowledge graph embedding by relational rotation in complex space,” in *Proceedings of the International Conference on Learning Representations*, New Orleans, Louisiana, US, May 2019, <https://openreview.net/forum?id=HkgEQnRqYQ>.
- [31] Z. Zhang, J. Cai, Y. Zhang, and J. Wang, “Learning hierarchy-aware knowledge graph embeddings for link prediction,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 3065–3072, New York Midtown, USA, February 2020.
- [32] H. Liu, Y. Wu, and Y. Yang, “Analogical inference for multi-relational embeddings,” in *Proceedings of the 34th International Conference on Machine Learning ICML'17*, vol. 70, pp. 2168–2178, Sydney NSW Australia, August 2017.
- [33] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, “Reasoning with neural tensor networks for knowledge base completion,” in *Proceedings of the 26th International Conference on Neural Information Processing Systems NIPS'13*, vol. 1, pp. 926–934, Red Hook, NY, USA, December 2013.
- [34] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, “Representation learning of knowledge graphs with entity descriptions,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence AAAI'16*, pp. 2965–2971, Phoenix Arizona, February 2016.
- [35] R. Xie, Z. Liu, and M. Sun, “Representation learning of knowledge graphs with hierarchical types,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence IJCAI'16*, pp. 2965–2971, New York, USA, July 2016.

- [36] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Jointly embedding knowledge graphs and logical rules," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 192–202, <https://aclanthology.org/D16-1019>, Austin, Texas, November 2016.
- [37] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 705–714, <https://aclanthology.org/D16-1019>, Lisbon, Portugal, September 2015.
- [38] V. T. Ho, D. Stepanova, M. H. Gad-Elrab, E. Kharlamov, and G. Weikum, "Rule learning from knowledge graphs guided by embedding models," in *Denny Vrandečić, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl*, pp. 72–90, Springer International Publishing, Berlin, Germany, 2018.
- [39] D. Lei, G. Jiang, X. Gu, K. Sun, Y. Mao, and X. Ren, "Learning collaborative agents with rule guidance for knowledge graph reasoning," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8541–8547, November 2020, <https://aclanthology.org/2020.emnlp-main.688>.
- [40] R. Das, S. Dhuliawala, M. Zaheer et al., "Go for a walk and arrive at the answer: reasoning over paths in knowledge bases using reinforcement learning," in *Proceedings of the International Conference on Learning Representations*, Vancouver, Canada, April 2018.
- [41] M. Gardner and T. Mitchell, "Efficient and expressive knowledge base completion using subgraph feature extraction," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1488–1498, Lisbon, Portugal, September 2015, <https://aclanthology.org/D15-1173>.
- [42] X. Han, M. Huang, L. Meng, and X. Zhu, "Ssp: semantic space projection for knowledge graph embedding with text descriptions," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17*, pp. 3104–3110, San Francisco California USA, February 2017.
- [43] Z. Cui, P. Kapanipathi, K. Talamadupula, T. Gao, and Q. Ji, "Type-augmented relation prediction in knowledge graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, pp. 7151–7159, Palo Alto, California, February 2021.
- [44] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, "End-to-end structure-aware convolutional networks for knowledge base completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3060–3067, Hawaii, January 2019.
- [45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 4171–4186, Minneapolis, Minnesota, June 2019, <https://aclanthology.org/N19-1423>.
- [46] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018, <https://paperswithcode.com/paper/improving-language-understanding-by>.
- [47] A. Dosovitskiy, L. Beyer, K. Alexander et al., "An image is worth 16x16 words: transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations*, May 2021.
- [48] Z. Liu, Y. Lin, Y. Cao et al., "Swin transformer: hierarchical vision transformer using shifted windows," in *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, Montreal, BC, Canada, October 2021.
- [49] Q. Wang, P. Huang, H. Wang et al., "Coke: contextualized knowledge graph embedding," 2019, <https://arxiv.org/abs/1911.02168>.
- [50] B. He, D. Zhou, J. Xie, J. Xiao, X. Jiang, and Q. Liu, "Ppke: knowledge representation learning by path-based pre-training," 2020, <https://arxiv.org/abs/2012.03573>.
- [51] P. Diederik, "Kingma and Jimmy Ba. Adam: a method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, Banff, AB, Canada, April 2014.
- [52] L. Chao, J. He, T. Wang, and W. Chu, "PairRE: knowledge graph embeddings via paired relation vectors," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, vol. 1, pp. 4360–4369, August 2021.