Reliable Knowledge Graph Path Representation Learning

SEUNGMIN SEO*, BYUNGKOOK OH*, AND KYONG-HO LEE*
Department of Computer Science, Yonsei University, Seoul 03722, South Korea
Corresponding author: Kyong-Ho Lee (khlee89@yonsei.ac.kr)
This work was supported by the Korea Electric Power Corporation under Grant R18XA05.

ABSTRACT Knowledge graphs, which have been widely utilized in various intelligent applications, are highly incomplete. Many valid facts can be inferred from existing facts in knowledge graphs. A promising approach for this task is a knowledge graph representation learning, which aims to represent entities and relations into low-dimensional vector spaces. Most of the existing methods mainly focus on direct relationships between entities and do not reflect the semantics of multi-hop relation paths. Although a few methods have studied the problem of multi-hop path-based representation learning, they fail to distinguish reliable relation paths among a majority of meaningless relation paths. In this paper, we propose a reliable path-based knowledge graph representation learning method, called RKRL. Specifically, we combine the representations of intermediate entities and relations on relation paths to learn more meaningful knowledge representations. Also, we present a reliable knowledge graph path ranking method to avoid the unnecessary computation of unreliable paths and find semantically valid relation paths. Experimental results on benchmark datasets show that our method achieves consistent improvement on typical evaluation tasks for knowledge representations, compared with the classical and state-of-the-art representation learning baselines.

INDEX TERMS Knowledge graph embedding, representation learning, path reasoning, link prediction.

I. INTRODUCTION
Knowledge graphs, such as Freebase, WordNet, DBpedia, NELL, Google’s Knowledge Graph and YAGO, have become an important resource for intelligent applications including question answering, relation extraction from text, named entity recognition, and knowledge inference. A typical knowledge graph consists of facts in the form of RDF triples (subject, predicate/property, object). Recently, RDF triples are also represented as (headentity, relation, tailentity), abridged as \((h, r, t)\) [3]. Since triple facts are symbolic, it is required to represent the triple facts as low-dimensional vector representations, in order to apply knowledge graphs to various applications. Although knowledge graphs have broad applicability, knowledge graphs are still far from complete to reflect the real-world knowledge because automatically generated knowledge graphs include invalid triples and do not contain crucial facts.

A knowledge graph representation learning, also called a knowledge graph embedding, is a promising approach to learn representations from data while preserving the inherent meaning and structure of the original graph. For each triple, a scoring function is defined to measure its plausibility, and then the representations are learned by optimizing a margin-based loss function. TransE [3] is one of the most typical representation learning models, and its extensions [20], [34] have been widely developed in recent years. However, most typical representation learning models only deal with direct relationships between two entities.

In this context, multi-step path-based representation learning methods, such as PTransE [21], RTransE [11], RPE, [19], CKRL [39], and PaSKoGE [24], have been proposed to exploit a sequence of relations to infer knowledge, where the relation paths are denoted as the sequences of relations \(p = (r_1, r_2, \cdots, r_n)\) between the entity pair \((h, t)\). For example, if the knowledge graph does not contain a fact like LeonardoDiCaprio actIn theRevenant, then we can infer the missing relation using the relation path LeonardoDiCaprio play HughGlass,
and HughGlass characterIn theRevenant. Fig.1 shows the brief illustration of the path-based knowledge representation learning.

Although path-based knowledge representation learning methods perform better on real applications such as entity prediction and relation prediction, they fail to capture the whole meanings of relation paths. Since the existing methods only employ the intermediate relations in the relation paths, they cannot reflect the meaning of the intermediate entities. Also, it is critical to find and exploit semantically reliable relation paths for path-based representation learning, but the existing methods do not present a valid solution since they cannot distinguish meaningful paths and meaningless paths.

In this paper, we propose a reliable path-based knowledge graph representation learning method, called RKRL, to find reliable relation paths and reflect the implicit meanings of relation paths in the knowledge representations. Specifically, we present a reliable knowledge graph path ranking algorithm to filter a huge number of invalid relation paths in knowledge graphs and find semantically valid relation paths. For example, there is often a relation path in a knowledge graph, e.g. LeonardoDicaprio bornIn USA hasCapital WashingtonDC, but this kind of paths is meaningless for inferring missing relations between the entity pair (LeonardoDicaprio, WashingtonDC). We consider the degree of the importance of relations for the specific entity by measuring the frequency of relations directly linked to that entity, along with the flow of resource amount from head entity to tail entity [21]. We take advantage of the fact that some relations might be more critical than other relations for a specific entity. Moreover, we present new path representations and score functions to translate the exact meanings of multi-hop relation paths. Our proposed method incorporates the representations of intermediate entities and relations together in order to produce semantically discriminable path representations.

We check the validity of our idea before describing the detailed methodology and evaluate our RKRL method on publicly available benchmark datasets. We compare the performance of our method on typical evaluation tasks, including entity prediction, relation prediction, triple classification, and knowledge graph noise detection. The experimental results show that our method generates better knowledge representations and performs better than the existing methods on all evaluation tasks.

The rest of this paper is organized as follows. We first review the related works in Section 2 and then describe a detailed description of our method in Section 3. Section 4 reports the experimental results, and Section 5 concludes the paper.

II. RELATED WORK

A. TRANSLATION-BASED KNOWLEDGE REPRESENTATION LEARNING

Translation-based KRL models represent entities and relations of knowledge graphs as low-dimensional vectors. TransE [3], the most typical translation-based method, assumes the equation \( h + r = t \) where the boldface characters denote the embedding vectors, e.g., \( h \) is the embedding vectors of head entity \( h \). For a given triple fact \( (h, r, t) \) in a knowledge graph, TransE defines the score function as follows:

\[
f(h, r, t) = ||h + r - t||
\]

(1)

The translation assumption of TransE is quite straightforward and effective, but cannot address the N-to-1, 1-to-N, and N-to-N relations well. TransH [34] overcomes this problem by projecting entities into a relation-dependent hyperplane, whereas TransR [20] projects entities and relations into separate relation-specific hyperplanes.

Moreover, there are many extended methods to overcome the limitation of translation-based KRL models by utilizing additional information. The authors of [33] incorporates physical and logical rules to improve the inference accuracy of KRL models. It formulates inference as an integer linear programming (ILP) problem. TransG [35] handles a problem that a relation in a knowledge graph may have different meanings depending on the linked entities. They focus on discovering the semantic cluster of a relation by designing a Bayesian non-parametric infinite mixture KRL model. To create compositional representations, HoIE [26] uses the correlation of entity embeddings. TKRL [38] takes hierarchical entity types into considerations and makes projection matrices for entities with type encoders. The authors of [40] present a role-specific projection to preserve the logical properties of relations. SSE [14] puts additional semantic information to impose constraints on the geometric structure of an embedding space. PuTransE [31] generates multiple embedding spaces via their structure-aware triplet selection, which estimates an energy score non-parametrically. To adaptively determine the margin
over different knowledge graphs, TransA [42], [43] proposes a loss function with flexible margins.

Although these classic approaches employ the simple translation assumption and achieve impressive results on link prediction and triple classification tasks, most of them fail to model multi-hop relation paths, which imply more complex meanings and indispensable for multi-hop reasoning tasks.

B. MULTI-HOP PATH-BASED KNOWLEDGE REPRESENTATION LEARNING

To tackle the multi-hop reasoning problem, various path-based KRL models have been proposed. RTransE [11] composes length-2 relation paths by simply adding translation vectors. The approach of [15] similarly generates new auxiliary triples for compositional training. However, constructing multi-hop path vectors through simply composing relation vectors does not help for embedding more longer relation paths. PTransE [21] takes the reliability of relation paths into account by adapting the path-constraint resource allocation algorithm. The author of [30] proposed the dynamic programming algorithm that supports the modeling of all possible relation paths. Since it builds the sums of all path representations, it performs an irrelevant calculation on unreliable paths. In order to support more path-specific knowledge representation learning, RPE [19] exploits path-specific projection and extends the relation-specific type constraints. To detect possible noises in a knowledge graph, CKRL [39] employs triple confidence for each triple fact. Also, there is an approach for adaptively finding optimal margin-based loss function for path-based representation learning [24].

Although those path-based KRL methods introduced new score functions to take advantage of the semantics of relation paths, they do not consider the intermediate entities on the path when embedding relation paths. They fail to capture the fact that although the intermediate relations in the paths are the same, the meaning of the relation paths can vary depending on the intermediate entities (see Fig. 2). Also, PRA [18] and PCRA(Path-specific Resource Allocation) [21] have been proposed to find semantically meaningful relation paths, but they do not fully reflect structural information in a knowledge graph, which makes it challenging to explore meaningful and valid relation paths.

III. METHODOLOGY

A. BACKGROUND

In this section, we introduce the basic idea for reliable knowledge graph path representation learning (RKRL). Firstly, we observe whether the intermediate entities on relation paths have a beneficial influence on clarifying the meaning of relation paths. Typically, prior path-based knowledge graph representation learning methods only combines the meanings of intermediate relations in order to interpret the semantics of relation paths. Thus, they cannot grasp the exact meanings of relation paths, which might be different according to the intermediate entities.

Figure 2 shows the motivating examples of the importance of intermediate entities when combining the components of relation paths. Two relation paths a) and b) have the same head entity, target entity, and intermediate relations, but they have different intermediate entities, TheWolfOfWallStreet on relation path a) and TheRevenant on relation path b). Given the semantics of the relation path a), we would like to infer both beAwarded and beNominated relations between an entity pair (LeonardoDiCaprio, BestActor). On the other hand, we would like to infer only the beNominated relation concerning the semantics of the relation path b). However,
the existing path-based knowledge graph representation learning methods incorrectly infer the `beAwarded` relation given the relation path b), because the intermediate relations are the same in both relation path a) and b). We would introduce the reliable knowledge graph path representation learning model in section 3.2 to overcome this challenge.

Another challenge for learning better knowledge representations is finding and exploiting the reliable relation paths in a knowledge graph. There are a tremendous number of relation paths in a knowledge graph, but most of them are meaningless. For example, there is an unreliable and meaningless relation path c) in Fig.2, which is composed of the facts LeonardoDiCaprio `bornIn` USA `hold` AcademyAward `award` BestActor. In order to employ only reliable relation paths for learning knowledge graph representations, we follow the assumption of path-constraint resource allocation (PCRA) [21]. The PCRA assumes that the reliability of relation paths is equal to the resource amount allocated from \( h \) to \( t \) along the relation paths. However, since the assumption of PCRA is insufficient to find reliable relation paths in consideration of the distinct characteristics between intermediate entities and relations, we add an extra feature to measure the path reliability. We consider the degree of the importance of entities and relations, we add an extra feature to measure the consideration of the distinct characteristics between intermediate entities and relations. For the specific entity by measuring the frequency of relations directly linked to that entity. In other words, we assume that the more frequent the relation \( r_i \) appears with an entity \( e \), \( r_i \) carries the relatively important meaning than average relation of an entity \( e \) in the relation paths. For example, there are many `actIn` relations, whereas there is only one `bornIn` relation for the LeonardoDiCaprio entity. From that fact, we can infer that the LeonardoDiCaprio entity might be `person/actor` entity type and the `actIn` relation would connect with a variety of meaningful relation paths for that type of entities.

B. RELIABLE PATH REPRESENTATION LEARNING

We first introduce the notations used in this paper. Given a triple \((h, r, t)\) in the knowledge graph, we denote the head and tail entities \(h, t \in E\), and the relation \( r \in R\), where \( E \) and \( R \) are the sets of entities and relations. The relation path \( \pi \in \Pi \) from \( h \) to \( t \) is denoted as \((h, \pi, t)\), where \( \Pi \) is the set of relation paths between \( h \) and \( t \). Each relation path has an arbitrary length and consists of intermediate entities and relations. We can also represent the relation path as follows:

\[
(h, \pi, t) = (h, r_1, e_1, \cdots, r_n, e_n, t),
\]

where \( e_i \) and \( r_i \) denote intermediate entities and relations. To exploit the motivating idea, we propose a new score function as follows:

\[
g(h, r, t) = f(h, r, t) + f(h, \pi, t),
\]

where \( f(h, r, t) \) denotes plausibility between entities with direct relationships as defined in (1), and \( f(h, \pi, t) \) denotes the plausibility of multi-step relation paths starting from \( h \) to \( t \). The score will be smaller if the relation path \( \pi \) and relation \( r \) are the correct. \( f(h, \Pi, t) \) is defined as follows:

\[
f(h, \Pi, t) = \frac{1}{Z} \sum_{\pi_i \in \Pi} f(h, \pi_i, t) \cdot R(\pi_i|h, t),
\]

where \( R(\pi_i|h, t) \) denotes the reliability of path \( \pi_i \) from \( h \) to \( t \), which will be described in the next subsection, and \( Z = \sum R(\pi_i|h, t) \) is a normalization vector. The score function of \( f(h, \pi_i, t) \) of a triple \((h, \pi_i, t)\) is computed by:

\[
f(h, \pi_i, t) = ||h + \pi_i - t||.
\]

For the representation of a path \( \pi_i \in \Pi \), we define the weighted relation path representation \( \pi_i \) as follows:

\[
\pi_i = W_{r_1} \cdot r_1 + \cdots + W_{r_n} \cdot r_n
\]

C. RELIABLE KNOWLEDGE GRAPH PATH RANKING

To find and utilize the reliable relation paths in a knowledge graph, we should filter a tremendous number of meaningless relation paths. We use two features to measure the reliability of relation paths. The first one is the resource amount from head entity \( h \) to tail entity \( t \) presented in [21]. As we have seen before, it is not enough to judge whether the given path is semantically meaningful or meaningless by the resource amount alone. Thus, we exploit the appearance ratio of the relation for the entity to reflect the distinguishing characteristics of entities and relations.

Moreover, we apply the semantic constraints of the knowledge graphs to the reliable relation path selection. Practically, entities in schema-based knowledge graphs are divided into groups called classes (or types), which are semantically predefined in an RDF schema, data-modeling vocabulary. For example, the LeonardoDiCaprio entity is assigned to the person class. An RDF schema also provides the predicates `rdfs:domain` and `rdfs:range`. They indicate that a relation-type should be related to specific classes. A domain is used to restrict the class of a head entity, i.e., subject in an RDF triple, and a range restricts the type of a tail entity, i.e., object in an RDF triple. For instance, the domain and range of relation-type `actIn` might be `person` and `film`, respectively.

Formally, given a path \( \pi_i = r_1, e_1, \cdots, r_{i-1}, e_{i-1}, r_i \) and entity pair \((h, t)\), the reliability of relation path is defined as follows:

\[
R_{\pi_i}(e) = \sum_{e' \in E_{i-1}(\cdot, e)} \frac{R_{\pi_i}(e')}{E_i(e', \cdot)} \cdot Q(r_i),
\]

where \( R_{\pi_i}(e) \) is the reliability of path with given entity pair \((h, e)\), \( E_{i-1}(\cdot, e) \) is the direct predecessors of \( e \) via \( r_i \) which belong to the `rdfs:domain`, and \( E_i(e', \cdot) \) is the direct successors of \( e' \) via \( r_i \) which belong to the `rdfs:range`. \( Q(r_i) \) represents the importance of relation \( r_i \) specialized in an entity \( e' \) and can be calculated as follows:

\[
Q(r_i) = |r_i| \cdot \frac{|r|}{|m|},
\]
The optimization objective is defined as:

\[
L = \sum_{(h,r,t) \in T} L(h, r, t) + \frac{1}{2} \sum_{\pi_i \in \Pi} \mathbb{E}(L(h, \pi_i, t) \cdot R(\pi_i | h, t)),
\]

and the loss of triple and relation paths are as follows:

\[
L(h, r, t) = \sum_{(h', r', t') \in T'} \max(0, \gamma_1 + f(h, r, t) - f(h', r', t')) + \gamma_2 + f(h, \pi_i, t)
\]
\[
L(h, \pi_i, t) = \sum_{(h', r', t') \in T'} \max(0, \gamma_2 + f(h, \pi_i, t) - f(h', \pi_i, t')) \tag{12}
\]

where \(\gamma_1\) and \(\gamma_2\) are the hyperparameters of margin, and \(T'\) represents the set of corrupted triples. Since knowledge graphs do not provide the negative triples, we sample the negative triples with incorporating the type constraints into the optimization as follows:

\[
T' = \{(h', r, t) | h' \in E \} \cup \{(h, r, t') | t' \in E \} \cup \{(h, r', t) | r' \in R \}, (h, r, t) \in T. \tag{13}
\]

where \(|r_i|\) is the number of relations linked to the entity \(e'\) via relation \(r_i\), \(|r|\) is the number of all relations linked to the entity \(e'\), and \(|m|\) is the number of distinct relation types linked to the entity \(e'\). The value will be larger if the relation \(r_i\) frequently appears than the average number of occurrences of other relations of an entity \(e'\). In other words, we follow the assumption that the more frequent the relation \(r_i\) appears, the more frequent the relation \(r_i\) appears than average relation of an entity \(e'\). Furthermore, we follow the local closed-world assumption (LCWA) [16] to reduce the influence of the missing and invalid facts of knowledge graphs. Thus, we can obtain the domain and range constraints of the targeted relation for observed triples.

Moreover, we add an inverse relation for each relation in knowledge graphs to enrich the relation paths which may not be presented in knowledge graphs. For each triple \((h, r, t)\), we generate its reverse triple \((t, r^{-1}, h)\). For example, given the relation path LeonardoDicaprio \(\text{actIn}^{-1}\) TheRevenant beNominate LeonardoDicaprio. In order to exploit the multi-hop relation paths in the representation learning, we should first find and select the meaningful relation paths. Theoretically, our model could find arbitrary long-tailed relation paths, but the meaning of relation paths will fade away when the length of relation paths grows exponentially. Thus, we limit the maximum relation path length to at most 3 in the implementation.

### Table 1. Number of parameters and computational complexity.

| Methods        | Model Parameters | Time Complexity |
|----------------|------------------|-----------------|
| TransE         | \(O(d \cdot n_e + d \cdot n_r)\) | \(O(n_{triple})\) |
| TransH         | \(O(d \cdot n_e + d \cdot n_r)\) | \(O(2 \cdot d \cdot n_{triple})\) |
| TransR         | \(O(d \cdot n_e + d \cdot n_r + d^2 \cdot n_r)\) | \(O(2 \cdot d^2 \cdot n_{triple})\) |
| PTransE(ADD, 2-hop) | \(O(d \cdot n_e + d \cdot n_r)\) | \(O(p \cdot l \cdot n_{triple})\) |
| PTransE(MUL, 2-hop) | \(O(d \cdot n_e + d \cdot n_r)\) | \(O(p \cdot l \cdot n_{triple})\) |
| TransA         | \(O(d \cdot n_e + d \cdot n_r)\) | \(O(xxxx)\) |
| RPE            | \(O(d \cdot n_e + d \cdot n_r + d^2 \cdot n_r)\) | \(O(2 \cdot d^2 \cdot p \cdot n_{triple})\) |
| PaSKoGE        | \(O(d \cdot n_e + d \cdot n_r)\) | \(O(p \cdot l \cdot n_{triple})\) |
| CKRL           | \(O(d \cdot n_e + d \cdot n_r)\) | \(O(p \cdot l \cdot n_{triple})\) |
| RKRL (our method) | \(O(d \cdot n_e + d \cdot n_r)\) | \(O(p \cdot l \cdot n_{triple})\) |

### Table 2. The statistics of the datasets.

| Data     | #Rel | #Ent | #Train | #Valid | #Test |
|----------|------|------|--------|--------|-------|
| FB15K    | 1,345| 14,951| 483,142| 50,000 | 59,071|
| WN18     | 18   | 40,943| 141,442| 5,000  | 5,000 |
widely used to evaluate the complexity of knowledge graph representation learning methods [3], [19], [24], [26], [34]. We denote \( n_e \) and \( n_r \) as the number of entities and relations, and \( n_{\text{triple}} \) is the number of triples. \( d \) is the embedding dimension, and \( p \) is the expected number of relation paths between entity pairs, and \( l \) is the expected length of relation paths. As shown in Table 3, the model parameter size of RKRL is one of the smallest numbers \( O(d \cdot n_e + d \cdot n_r) \), which is the same as simple embedding methods. When comparing the computational complexity per each iteration in optimization, RKRL has the same complexity of other path-based knowledge representation learning model, \( O(p \cdot l \cdot n_{\text{triple}}) \).

TABLE 3. Distribution of the relations of FB15K (%).

| Categories     | 1-to-1 | 1-to-N | N-to-1 | N-to-N |
|----------------|--------|--------|--------|--------|
| Training sets  | 27.36  | 22.97  | 29.29  | 20.38  |
| Test sets      | 25.50  | 23.27  | 28.92  | 22.31  |

IV. EXPERIMENTS

A. DATASETS

We evaluate our RKRL methods on two widely used knowledge graphs, Freebase and WordNet. Especially, we exploit benchmark datasets extracted from both datasets, FB15K and WN18. FB15K has 14,951 entities and 1,345 types of relations. WN18 has 40,943 entities and 18 types of relations. The detailed statistics of each datasets are reported in Table 3.

Additionally, we use the type information and type-constraints provided by [38] to incorporate the type information of triples. We filter the triples of relations which violate type-constraints or never appear in triples. Since WN18 does not provide the type information, we analyze and approximate the type information of observed triples in WN18 based on the LCWA.

Also, as shown in Table 4 and Table 5, we categorize relations and facts of FB15K into four types concerning the relation categories as defined in [3]. We checked that a given distribution of the triples is particularly unbalanced. The triples with N-to-N relations occupy most of FB15K. It is apparent that knowledge graph representation learning models should treat carefully this type of triples, which tend to build multi-hop paths.

TABLE 4. Distribution of the relations of WN18 (%).

| Categories     | 1-to-1 | 1-to-N | N-to-1 | N-to-N |
|----------------|--------|--------|--------|--------|
| Training sets  | 1.57   | 9.48   | 15.88  | 73.07  |
| Test sets      | 1.48   | 9.54   | 15.12  | 73.86  |

B. EXPERIMENTAL SETTINGS

We divide our RKRL models into two types in the experiments. RKRL (COM) indicates the version which only considers the composition of relation paths, including intermediate entities, while RKRL (COM + RPR) considers both the intermediate entities in relation paths and reliable knowledge graph path ranking.

Based on the datasets, we use mean rank and Hits@10 score as evaluation metrics. Mean rank denotes the average rank of correct entities or relations. Hits@10 score denotes the average proportion of correct entities or relations ranked within the top 10 elements. However, A corrupted triple may appear in a knowledge graph, which might be ranked higher than the test triple. Mean rank and Hits@10 thus ignore these corrupted triples which have appeared in the results. We call the original settings (possibility including corrupted triples) as Raw and the filtered one as Filter. In both settings, an efficient embedding model would achieve a lower mean rank and higher Hits@10. To examine the quality of knowledge representations of our RKRL models, we performed four evaluation tasks: entity prediction (link prediction), relation prediction, triple classification, and knowledge graph noise detection.

C. ENTITY PREDICTION

Entity prediction aims to predict one of the head and tail entities for a triple. We replace head or tail entities in the test sets with the entities in the \( T' \). Then a probabilistic score of each corrupted triple is calculated with the score function in the descending order. We compare our proposed methods with the baseline methods, including...
S. Seo et al.: RKRL

TABLE 7. Evaluation results on FB15K by mapping properties of relations on prediction (%).

| Tasks          | Predicting Heads (Hits@10) | Predicting Tails (Hits@10) |
|----------------|----------------------------|----------------------------|
|                | 1-to-1 | 1-to-N | N-to-1 | N-to-N | 1-to-1 | 1-to-N | N-to-1 | N-to-N |
| SME            | 30.9   | 69.6   | 19.9   | 38.6   | 28.2   | 13.1   | 76.0   | 41.8   |
| TransE         | 43.7   | 65.7   | 18.2   | 47.2   | 43.7   | 19.7   | 66.7   | 50.0   |
| TransH         | 66.7   | 81.7   | 30.2   | 57.4   | 63.7   | 30.1   | 83.2   | 60.8   |
| TransR         | 76.9   | 77.9   | 38.1   | 66.9   | 76.2   | 38.4   | 76.2   | 69.1   |
| RTransE        | 80.5   | 81.3   | 45.0   | 65.5   | 80.0   | 65.1   | 77.6   | 78.0   |
| PTransE (Add, 2-hop) | 91.0   | 92.8   | 60.9   | 83.8   | 91.2   | 74.0   | 88.9   | 86.4   |
| PTransE (Add, 3-hop) | 90.1   | 92.0   | 58.7   | 86.1   | 90.7   | 70.7   | 87.5   | 88.7   |
| RPE            | 92.5   | 96.6   | 63.7   | 87.9   | 92.5   | 79.1   | 95.1   | 90.8   |
| CKRL           | 65.6   | 76.5   | 35.3   | 50.7   | 50.8   | 30.6   | 75.8   | 58.0   |
| PaSKoGE        | 89.7   | 94.8   | 62.3   | 86.7   | 89.3   | 72.9   | 93.4   | 88.9   |
| RKRL (COM)     | 93.0   | 96.8   | 64.8   | 88.3   | 93.0   | 80.0   | 95.3   | 91.0   |
| RKRL (COM + RPR) | 93.6   | 97.0   | 65.6   | 89.5   | 93.5   | 81.2   | 96.0   | 91.5   |

D. RELATION PREDICTION

Entity prediction aims to predict the relations given a pair of entities $h$ and $t$. It replaces the relations in the test sets with the relations in the $T'$ and returns a list of candidate relations in descending order of score. Specifically, we adopt Hits@1 instead of Hits@10 for comparison following the procedure of [21], since Hits@10 for all methods exceeds 95.

The optimal configurations for relation prediction are similar to those of entity predictions: $\lambda = 0.001$ for SGD, embedding dimension $d = 50$, margin $\gamma = 1$, and the threshold for the path reliability $\eta = 0.05$, and $L_1$ as dissimilarity.

Table 8 shows the evaluation results of relation prediction. The results demonstrate that our methods outperform most of baseline methods including TransE, TransR, RTransE, and PTransE. Especially, RKRL (COM + RPR) achieves

TABLE 8. Evaluation results of relation prediction (%).

| Datasets | WN18 (Hits@1) | FB15K (Hits@1) |
|----------|---------------|----------------|
|          | Raw | Filter | Raw | Filter | Raw | Filter |
| TransE   | 66.1 | 66.1 | 65.1 | 64.3 | 67.1 | 67.1 |
| TransR   | 66.2 | 66.4 | 67.6 | 87.6 | 28.8 | 28.9 |
| RTransE  | 67.1 | 67.3 | 69.5 | 93.6 | 64.4 | 64.6 |
| PTransE (ADD, 2-hop) | 97.5 | 97.8 | 70.2 | 94.5 |
| PTransE (ADD, 3-hop) | 97.5 | 97.9 | 70.7 | 95.0 |
| RPE      | 66.3 | 67.0 | 67.8 | 88.0 | 79.2 | 79.2 |
| PaSKoGE  | 89.8 | 89.8 | 90.5 | 94.2 |
| RKRL (COM) | 97.5 | 97.9 | 70.7 | 95.0 |
| RKRL (COM + RPR) | 97.8 | 98.2 | 71.0 | 96.8 |

TABLE 9. Evaluation results of triple classification (%).

| Metric | Accuracy |
|--------|----------|
| TransE | 77.8     |
| TransH | 78.4     |
| TransR | 79.2     |
| PTransE (ADD, 2-hop) | 83.4 |
| PTransE (ADD, 3-hop) | 82.9 |
| RPE    | 89.8     |
| CKRL   | 83.5     |
| PaSKoGE| 90.5     |
| RKRL (COM) | 91.0 |
| RKRL (COM + RPR) | 92.2 |

D. RELATION PREDICTION

Entity prediction aims to predict the relations given a pair of entities $h$ and $t$. It replaces the relations in the test sets with the relations in the $T'$ and returns a list of candidate relations in descending order of score. Specifically, we adopt Hits@1 instead of Hits@10 for comparison following the procedure of [21], since Hits@10 for all methods exceeds 95.

The optimal configurations for relation prediction are similar to those of entity predictions: $\lambda = 0.001$ for SGD, embedding dimension $d = 50$, margin $\gamma = 1$, and the threshold for the path reliability $\eta = 0.05$, and $L_1$ as dissimilarity.

Table 9 shows the evaluation results of relation prediction. The results demonstrate that our methods outperform most of baseline methods including TransE, TransR, RTransE, and PTransE. Especially, RKRL (COM + RPR) achieves
an improvement by 1.2% compared to PaSKoGE in terms of filtered Hits@1 on FB15K. Also RKRL (COM + RPR) performs increases Hits@1 by 0.5% compared to PaSKoGE on WN18. Comparing the performance of RKRL (COM) and RKRL (COM + RPR), we found that the reliable knowledge graph path ranking algorithm plays a positive role as well in relation prediction.

E. TRIPLE CLASSIFICATION

Triple classification aims to judge whether the given triple \((h, r, t)\) is correct or not. In this binary classification, we use FB15K datasets. The classification process is as follows: for each triple in a knowledge graph, if the score of the triple \((h, r, t)\) is below a relation-specific threshold \(\delta\), the triple will be classified as positive, otherwise negative. The threshold \(\delta\) was determined by maximizing classification accuracies in all triples with the corresponding relation \(r\) on the validation dataset.

The optimal configurations of RKRL methods are identical to those of relation prediction: \(\lambda = 0.001\) for SGD, embedding dimension \(d = 50\), margin \(\gamma = 1\), and the threshold for the path reliability \(\eta = 0.05\), and \(L_1\) as dissimilarity.

Evaluation results on triple classification are reported in Table 10. We can see that RKRL methods show the best performance. Compared to CKRL, RPE, and PaSKoGE, RKRL (COM + RPR) improves performance by about 5%, 2%, and 1.5%, respectively. It suggests that the idea of composing intermediate entities is effective for obtaining better knowledge representations. Also, the results validate that the reliable knowledge graph path ranking algorithm can raise the determinability of knowledge graph representation learning models.

F. KNOWLEDGE GRAPH NOISE DETECTION

Knowledge graph noise detection aims to detect possible noises in knowledge graphs according to their triple scores. As described in [39], knowledge graph noises and conflicts derive from the misunderstanding between similar entities. For example, the noise BradPitt actIn TheRevenant is more likely to occur in real-world KG rather than AcademyAward actIn TheRevenant, in which the latter violates the type constraint.

We use three noisy datasets presented in [39], which are constructed based on FB15K with negative triples to be 10% (FB15K-N1), 20% (FB15K-N2), and 40% (FB15K-N3) of positive triples. FB15K-N1, FB15K-N2, and FB15K-N3 have 46,408, 93,782, and 187,925 negative triples, respectively. These three noisy datasets share the same validation and test sets with FB15K, and negative triples are fused into the training sets.

Similar to the triple classification, we consider the triples with higher scores are noises. The optimal configurations of RKRL methods are as follows: \(\lambda = 0.001\) for SGD, embedding dimension \(d = 50\), margin \(\gamma = 1\), and the threshold for the path reliability \(\eta = 0.05\), and \(L_1\) as dissimilarity.

Evaluation results on knowledge graph noise detection are reported in Fig. 3. The results demonstrate that our RKRL models achieve the best performances on all three noisy datasets. It suggests that our methods are suitable for finding the noises in a knowledge graph. In particular, compared with the best baseline method CKRL, RKRL (COM + RPR) obtains about 5% higher precision when the recall is 10%. We can also see that RKRL (COM + RPR) shows much better performance than RKRL (COM), which indicates that the reliable knowledge graph path ranking algorithm is well applied to the knowledge graph noise detection.

V. CONCLUSION

In this paper, we proposed a reliable path-based knowledge graph representation learning method, called RKRL, to better incorporate the semantics of relation paths within knowledge representations. We exploited the intermediate entities to learn more rich semantics of relation paths. To avoid unnecessary computation on unreliable paths and find semantically valid relation paths, we introduced the reliable knowledge graph path ranking method. Experimental results showed that our methods produce competitive results on several typical evaluation tasks for knowledge representations compared with the classical and state-of-the-art representation learning baselines.

REFERENCES

[1] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, “Learning structured embeddings of knowledge bases,” in Proc. AAAI, San Francisco, CA, USA, 2011, pp. 301–306.
[2] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, “Joint learning of words and meaning representations for open-text semantic parsing,” in Proc. AISTATS, 2012, pp. 127–135.

[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in Proc. NIPS, 2013, pp. 2879–2887.

[4] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, “A semantic matching energy function for learning with multi-relational data,” Mach. Learn., vol. 94, no. 2, pp. 233–259, Feb. 2014.

[5] K.-W. Chang, W.-T. Yih, B. Yang, and C. Meek, “Typed tensor decomposition of knowledge bases for relation extraction,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1568–1579.

[6] W. Chen, W. Xiong, X. Yan, and W. Y. Wang, “Variational knowledge graph reasoning,” in Proc. NAACL-HLT, 2018, pp. 1823–1832.

[7] L. Chen, Y. Zheng, and A. Li, “Path-specific knowledge graph embeddings,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 397–406.

[8] W. Zeng, Y. Lin, Z. Liu, and M. Sun, “Incorporating relation paths in knowledge graph embedding using subgraph feature extraction,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2015, pp. 1488–1498.

[9] K. Guu, J. Miller, and P. Liang, “Incorporating relationships with translations,” in Proc. Conf. Empirical Methods Natural Lang. Process. Nat. Proc., 2015, pp. 286–290.

[10] J. Artif. Intell. Res., vol. 55, pp. 715–742, Jul. 2018.

[11] S. Guo, Q. Wang, B. Wang, L. Wang, and L. Guo, “Semantically smooth knowledge graph embedding,” in Proc. 7th Int. Joint Conf. Natural Lang. Process., 2015, pp. 84–94.

[12] S. Guo, Q. Wang, B. Wang, L. Wang, and L. Guo, “SSE: Semantically smooth knowledge graph embeddings,” IEEE Trans. Knowl. Data Eng., vol. 29, no. 4, pp. 884–897, Apr. 2017.

[13] K. Guu, J. Miller, and P. Liang, “Traversing knowledge graphs in vector space,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2015, pp. 318–327.

[14] D. KrompaA, S. Baier, and V. Tresp, “Type-constrained representation learning of knowledge graphs,” in Proc. ISWC, 2015, pp. 640–655.

[15] N. Lao and W. C. Cohen, “Relational retrieval using a combination of path-constrained random walks,” Mach. Learn., vol. 81, no. 1, pp. 53–67, Oct. 2010.

[16] N. Lao, T. Mitchell, and W. C. Cohen, “Random walk inference and learning in a large scale knowledge base,” in Proc. EMNLP, 2011, pp. 529–539.

[17] X. Lin, Y. Liang, F. Gionchiglia, X. Feng, and R. Guan, “Relation path embedding in knowledge graphs,” Neural Comput. Appl., vol. 31, no. 9, pp. 5629–5639, Sep. 2019.

[18] C. Moon, P. Jones, and N. F. Samatova, “Learning Entity Type Embeddings for Knowledge Graph Completion,” in Proc. ACM Conf. Inf. Knowl. Manage. (CIKM), 2017, pp. 2181–2187.

[19] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and L. Liu, “Modeling relation paths for representation learning of knowledge bases,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2015, pp. 705–714.

[20] D. Q. Nguyen, K. Sirts, L. Qu, and M. Johnson, “STransE: A novel embedding model of entities and relationships in knowledge bases,” in Proc. NAACL-HLT, 2016, pp. 460–466.

[21] A. Neelakantan, B. Roth, and A. McCallum, “Compositional vector space models for knowledge base completion,” in Proc. Int. Joint Conf. Natural Lang. Process., 2015, pp. 156–166.

[22] Y. Jia, Y. Wang, Z. Jia, and X. Cheng, “Path-specific knowledge graph embedding,” Knowl.-Based Syst., vol. 151, pp. 37–44, Jul. 2018.

[23] M. Nickel, V. Tresp, and H. P. Kriegel, “A three-way model for collective learning on multi-relational data,” in Proc. ICML, 2011, pp. 809–816.

[24] M. Nickel, L. Rosasco, and T. A. Poggio, “Holographic embeddings of knowledge graphs,” in Proc. AAAI, 2016, pp. 1955–1961.
BYUNGKOOK OH received the B.S. degree in computer science from Yonsei University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. His research interests include knowledge graph embedding, natural language processing, and knowledge-based systems.

KYONG-HO LEE received the B.S., M.S., and Ph.D. degrees in computer science from Yonsei University, Seoul, South Korea, in 1995, 1997, and 2001, respectively. Previously, he has worked as a Guest Researcher with IT Laboratories, National Institute of Standards and Technology (NIST), Maryland. He was a Visiting Scholar with the University of California at Irvine, Irvine, in 2008. He is currently a Professor with the Department of Computer Science, Yonsei University. His research interests include services computing, cloud computing, and semantic web. He is also a member of the Editorial Board of the KSII Transactions on Internet and Information Systems, ICT Express, the International Journal of Web Science, the Journal of Information Processing Systems, and the Journal of Korea Multimedia Society.