EngineKGI: Closed-Loop Knowledge Graph Inference

Guanglin Niu, Bo Li, Yongfei Zhang, Shiliang Pu

1 Beihang University
2 Hikvision Research Institute
beihangngl@buaa.edu.cn

Abstract

Knowledge Graph (KG) inference is the vital technique to address the natural incompleteness of KGs. The existing KG inference approaches can be classified into rule learning-based and KG embedding-based models. However, these approaches cannot well balance accuracy, generalization, interpretability and efficiency, simultaneously. Besides, these models always rely on pure triples and neglect additional information. Therefore, both KG embedding (KGE) and rule learning KG inference approaches face challenges due to the sparse entities and the limited semantics. We propose a novel and effective closed-loop KG inference framework EngineKGI operating similarly as an engine based on these observations. EngineKGI combines KGE and rule learning to complement each other in a closed-loop pattern while taking advantage of semantics in paths and concepts. KGE module exploits paths to enhance the semantic association between entities and introduces rules for interpretability. A novel rule pruning mechanism is proposed in the rule learning module by leveraging paths as initial candidate rules and employing KG embeddings together with concepts for extracting more high-quality rules. Experimental results on four real-world datasets show that our model outperforms other baselines on link prediction tasks, demonstrating the effectiveness and superiority of our model on KG inference in a joint logic and data-driven fashion with a closed-loop mechanism.

1 Introduction

Knowledge graph (KG) has been developed rapidly in recent years. KGs store triple facts and some of them also have ontologies containing concepts of entities. Typical KGs include Freebase (Bollacker, Gottlob, and Flesca 2008), DBpedia (Lehmann et al. 2015) and NELL (Mitchell et al. 2018), which have proven to be incredibly effective for a variety of applications such as information extraction (Zhang et al. 2019a), dialogue system (Zhou et al. 2018), and question answering (Huang et al. 2019). However, the existing KGs are incomplete due to the limitations of human knowledge, web corpora, and extraction algorithms (Sadeghian et al. 2019). Thus, KG inference is a significant technique to complete KGs for better applications.

The existing KG inference approaches are usually classified into two main categories: (1) Rule learning-based models mine rules from KGs and employ these rules to predict new triples by deduction. In recent years, many rule learning tools specific to KG have been designed, such as AMIE+ (Galárraga et al. 2015), DRUM (Sadeghian et al. 2019) and AnyBurl (Meilicke et al. 2019). However, rule learning-based models suffer from low efficiency, especially for some large-scale KGs. Besides, the number of rules that can cover the inference is limited, so the generalization of rule learning-based models cannot be guaranteed. (2) KG embedding (KGE) models learn the embeddings of entities and relations to predict the missing triples with the well-designed score functions. Many KGE models such as TransE (Bordes et al. 2013), RotatE (Sun et al. 2019), HAKE (Zhang et al. 2020) and DualE (Cao et al. 2021) have proven their efficiency, benefiting from the low-dimensional numerical calculation rather than discrete graph search. Whereas, KGE models lack the interpretability that should be concerned in the general inference systems.

Some recent researches attempt to combine the advantages of rule learning-based and KGE-based models to complement each other. On the one hand, an idea is to introduce logic rules into KGE models, such as RUGE (Guo et al. 2018) and IterE (Zhang et al. 2019c). These approaches all convert the rules into formulas by t-norm based fuzzy logic to obtain newly labeled triples. However, these models cannot maintain the original and vital interpretability from symbolic rules. On the other hand, some rule learning-based models succeed in leveraging KG embeddings to extract rules via numerical calculation rather than discrete graph search, including RuLES (Ho et al. 2018), RLvLR (Omran, Wang, and Wang 2019) and DRUM (Sadeghian et al. 2019). Although the efficiency of mining rules is improved, the performance especially generalization of purely employing rules to implement KG inference is still limited.
In summary, all the previous KG inference models have two challenges: (1) Hard to balance performance: as mentioned above, all the existing three main streams of KG inference models cannot well balance accuracy, generalization, interpretability and efficiency at the same time. (2) The semantics implied in triples is limited: for the sparse entities that are associated with merely a few direct links, it will cause difficulty in learning effective embeddings for these entities with triple facts alone. Besides, both rule searching and pruning strategies of the existing rule learning models rely on triples without additional semantics such as paths and concepts, limiting the performance of rule learning.

To address the above challenges, we propose a closed-loop KG inference framework EngineKGI via combining the embedding-based rule learning and the rule-enhanced KGE, in which paths and concepts are utilized. Our model is named EngineKGI because it works like an engine as shown in Figure 1: (1) Intake Stroke. The closed-path rules are injected into the KGE module (analogous to intake) to guide the learning of KG embeddings, where the initial seed rules are mined by any rule learning tool, and the rule set would grow via our designed rule learning module from the first iteration. (2) Compression Stroke. The KGE module leverages the rules to compose paths between entity pairs for learning the low-dimensional embeddings (analogous to compression) of entities and relations, improving the interpretability and accuracy. (3) Expansion Stroke. The novel rule learning module outputs newly learned rules (analogous to exhaust) by the effective rule pruning strategy based on paths, relation embeddings and concepts. (4) Exhaust Stroke. Update the rule set (analogous to exhaust) by merging the previous rule set and the newly learned rules for boosting KGE and KG inference in the next iteration.

Our research makes the following three contributions:

- We design a novel and effective closed-loop KG inference framework EngineKGI that performs in a joint logic and data-driven fashion. The designed KGE and rule learning modules complement each other, balancing the accuracy, generalization, interpretability and efficiency.
- Rich additional information is exploited in the whole framework for injecting more valuable semantics. Path information generates candidate rules for improving the efficiency of the rule learning and provides more semantic associations between entities for the KGE module to handle the entity sparseness. Ontological concepts are leveraged in rule learning for a better pruning strategy.
- Experimental results on four datasets illustrate that our model EngineKGI outperforms various state-of-the-art KG inference approaches. The rule learning and link prediction evaluation over iterations verify the effectiveness of our model’s iteration pattern. Meanwhile, the case study illustrates the interpretability of our approach.

### 2 Related Work

Our work focuses on closed-loop KG inference via exploiting rich additional semantic information, including rules and paths as well as concepts. Thus, the closely relevant works include the following three lines of researches.

#### 2.1 Rule Learning-Based Models

According to the symbolic characteristics of KG, some rule learning techniques specific to KGs are proposed and used for KG inference for accuracy and interpretability, including AMIE+ (Galárraga et al. 2015), Anyburl (Melichec et al. 2019), DRUM (Sadeghian et al. 2019) and RLvrLR (Omran, Wang, and Wang 2019). AMIE+ (Galárraga et al. 2015) introduces optimized query writing techniques into traditional Inductive Logic Programming algorithms to generate horn rules and improves scalability, which is proven to be effective and convenient for learning rules from KGs. Both DRUM (Sadeghian et al. 2019) and RLvrLR (Omran, Wang, and Wang 2019) are the recent rule learning algorithms employing KG embeddings for improving efficiency and scalability. However, these rule learning-based models such as DRUM (Sadeghian et al. 2019) and Anyburl (Melichec et al. 2019) just infer tail but not head entities, and they lack generalization since the rules mined by these rule learning algorithms are hard to cover the inference of complex relations.

#### 2.2 KG Embedding Models

The typical KG embedding (KGE) model TransE (Bordes et al. 2013) regards the relations as translation operations from head to tail entities. Many translation-based variants are developed to modify TransE, including TransH (Wang et al. 2014), TransR (Lin et al. 2015b) and TransD (Ji et al. 2015). ComplEx (Trouillon et al. 2016) and RotatE (Sun et al. 2019) embeds the KG into a complex space while QuatE (Zhang et al. 2019b) and DualE (Cao et al. 2021) embeds relations into the quaternion space to model the symmetric and anti-symmetric relations. HAKE (Zhang et al. 2020) embeds entities into the polar coordinate system and models the semantic hierarchies of KGs. Graph neural networks are leveraged to encode the neighborhood of entities for updating entity and relation embeddings (Wang et al. 2021). RUGE (Guo et al. 2018) and IterE (Zhang et al. 2019c) both convert rules into formulas by t-norm fuzzy logic to infer newly labeled triples. Particularly, IterE iteratively conducts rule learning and KG embedding, but the significant distinctions between our model EngineKGI and IterE are: (1) Usage of rules: our model leverages rules to compose paths for learning KG embeddings while IterE uses rules to produce labeled triples. Meanwhile, we maintain the interpretability of the symbolic rules, while IterE does not. (2) Additional information: our model introduces paths and concepts into both rule learning and KG embedding while IterE simply depends on triples.

#### 2.3 Models with Paths or Concepts

In terms of the graph structure of KGs, paths establish extra multi-hop connections between entities. PathRNN (Nee-Lakantan, Roth, and McCallum 2015) encodes paths with recurrent neural network (RNN) to predict even unseen relations between entities. PTransE (Lin et al. 2015a) succeeds to extend TransE (Bordes et al. 2013) by measuring the plausibility not only among triple facts but also between relations and paths. Similar to PTransE, PaSkOE (Jia et al. 2018) modifies PTransE by an adaptive margin selection strategy.
Das et al. (Das et al. 2017) replaced entities with entity types for achieving better path representations. However, these path-based models learn path representations in a data-driven fashion, lacking interpretability and accuracy. Besides, a KG always consists of both concepts and instances. A concept represents the abstract features of an entity, which can enrich the semantics specific to entities (Hao et al. 2019; Wang, and Wang 2019), whereas, concepts are not fully utilized in these concept-enhanced models that neglect the concept-level correlations among triples.

3 Methodology

In this section, we firstly describe the problem formulation and notation of our work in section 3.1. Then, following the workflow of EngineKG as shown in Figure 2, we introduce the rule-enhanced KGE module in section 3.2 and the embedding-based rule learning module in section 3.3.

3.1 Problem Formulation and Notation

**Definition of Triple Facts.** Each triple fact \((h, r, t)\) in a KG is denoted as the head entity \(h\) and the tail entity \(t\) linked by the relation \(r\), where \(h\) and \(t\) are in the entity set \(E\) while \(r\) is in the relation set \(R\).

**Definition of Closed-Path Rule.** The closed-path (CP) rule is a fragment of the horn rule that we are interested in for our KGE module and the procedure of inference. A CP rule is of the form

\[
Head(x, y) \leftarrow B_1(x, z_1) \land B_2(z_1, z_2) \land \cdots \land B_n(z_{n-1}, y) \tag{1}
\]

where \(B_1(x, z_1), B_2(z_1, z_2), \ldots, B_n(z_{n-1}, y)\) are denoted as the atoms in the rule body \(Body(x, y)\), and \(Head(x, y)\) is the rule head. \(B_i\) and \(Head\) indicate relations. To measure the quality of the mined rules, the standard confidence (SC) and the head coverage (HC) are two predefined statistical measures to assess rules (Galarraga et al. 2015; Omran, Wang, and Wang 2019), which are defined as followings:

\[
Support = \#(e, e') : Body(e, e') \land Head(e, e') \tag{2}
\]

\[
SC = \frac{Support}{\#(e, e') : Body(e, e')} \tag{3}
\]

\[
HC = \frac{Support}{\#(e, e') : Head(e, e')} \tag{4}
\]

where \(\#(e, e')\) indicates the number of entity pairs \((e, e')\) that satisfy the condition on the right side of the colon. The standard confidence and the head coverage correspond to the standard accuracy and recall, respectively.

**Definition of Path.** A path between an entity pair \((h, t)\) is in the form of \([h \rightarrow r_1 \rightarrow e_1 \rightarrow \cdots \rightarrow r_n \rightarrow t]\) where \(r_i\) and \(e_i\) are the intermediate relation and entity, and the length of a path is the number of the intermediate relations.

**Definition of KG Inference.** KG inference namely link prediction focuses on predicting the missing entity in a triple since predicting entity is far more challenging than relation. Unlike the traditional KGE approaches, our closed-loop KG inference framework attempts to mine CP rules, learn KG embeddings simultaneously, and then complete the triples by the KG embeddings together with the learned rules.

3.2 Rule-Enhanced KGE Module

We aim to learn the entity and relation embeddings according to not only triple facts but also rules and paths. Firstly, we extract the paths by PCRA algorithm (Lin et al. 2015a). Apart from other path-finding approaches such as PRA (Lao, Mitchell, and Cohen 2011), the PCRA algorithm is selected in our work because it employs a path-constraint resource allocation strategy to extract paths from KGs and measure the reliability of each path. Notably, the reliability could be further exploited in the following KGE module. Particularly, we develop a joint logic and data-driven path representation mechanism to learn path embeddings.

**Logic-Driven Path Representation.** The CP rules could compose multi-hop paths into condensed ones with high confidence for enhancing their representation. For instance, considering a path [The Pursuit of Happiness \(\rightarrow\) CastActor \(\rightarrow\) Will Smith \(\rightarrow\) PersonLanguage \(\rightarrow\) English] in Figure 2 it is composed into a condensed path [The Pursuit of Happiness \(\rightarrow\) TVLanguage \(\rightarrow\) English] by the matched CP rule TVLanguage\((x, y) \leftarrow CastActor(x, z) \land PersonLanguage(z, y)\). Furthermore, the embedding of the relation TVLanguage could signify the embedding of the original multi-hop path.

**Data-Driven Path Representation.** For the scenario that the path cannot be further composed by rules such as the length-2 paths highlighted in Figure 2 we represent this path by adding all the relation embeddings along the path.

With the entity pair \((h, t)\) together with the linking path set \(\mathcal{P}\), the energy function for measuring the plausibility of the path-specific triple \((h, t, \mathcal{P})\) is designed as

\[
E_p(h, t, \mathcal{P}) = ||h + \mathbf{P} - t|| = \sum_{p \in \mathcal{P}} \sum_{p_i \in p} \frac{R(p_i|h, t)}{\sum_{p_i \in \mathcal{P}} R(p_i|h, t)} ||h + \mathbf{P}_i - t|| \tag{5}
\]
where \( h \) and \( t \) are the head and tail entity embeddings, respectively. \( p_i \) denotes the \( i \)-th path in the path set \( \mathcal{P} \) and \( p_r \) is the embedding of \( p_i \) achieved by the joint logic and data-driven path representation mechanism. \( R(p_i|h, t) \) indicates the reliability of path \( p_i \) between the given entity pair \((h, t)\) obtained by the PCRA algorithm. \( \mathbf{P} \) is the joint representation of the path set \( \mathcal{P} \) which is the sum of all the path embeddings weighted by the reliability of each path.

**Embedding Learning.** Along with the translation-based KGE models, the energy function for formalizing the plausibility of a triple fact \((h, r, t)\) is given as

\[
E_r(h, r, t) = \|h + r - t\| \tag{6}
\]

in which \( r \) is the embedding of the relation \( r \).

The aforementioned existing KGE techniques neglect the semantic association between relations. Remarkably, the length-1 rules model the causal correlations between two relations. As shown in Figure 3, the relation pair in the rule \( \text{CapitalOf}(x, y) \Leftarrow \text{LocatedIn}(x, y) \) should have higher similarity than other relations. Thus, we measure the association between relation pairs as

\[
E_r(r_1, r_2) = \|r_1 - r_2\| \tag{7}
\]

where \( r_1 \) and \( r_2 \) are the embeddings of relations \( r_1 \) and \( r_2 \). \( E_r(r_1, r_2) \) should be closer to a small value if \( r_1 \) and \( r_2 \) appear in a length-1 rule at the same time.

With the energy functions corresponding to the triple fact, path representation and the relation association, the joint loss function for training is designed as followings:

\[
L = \sum_{(h, r, t) \in \mathcal{T}} \left( L_t + \alpha_1 L_p + \alpha_2 L_r \right) \tag{8}
\]

\[
L_t = \sum_{(h, r, t' \in \mathcal{T}^r)} \left[ \gamma_1 + E_t(h, r, t) - E_t(h', r, t') \right]_+ \tag{9}
\]

\[
L_p = \sum_{(h, r, t') \in \mathcal{T}^r} \left[ \gamma_2 + E_p(h, r, t, P) - E_p(h', r', P') \right]_+ \tag{10}
\]

\[
L_r = \sum_{r \in S, r_0 \in S'} \sum_{r \in S'} \left[ \gamma_3 + E_r(r, r_0) - E_r(r, r_0) \right]_+ \tag{11}
\]

where \( L \) is the whole training loss consisting of three components: the triple-specific loss \( L_t \), the path-specific loss \( L_p \), and the relation correlation-specific loss \( L_r \). \( \alpha_1 \) and \( \alpha_2 \) are two weights for a trade-off of the influence of paths and relation association by rules. \( \gamma_1, \gamma_2 \) and \( \gamma_3 \) are three margins in each loss function. \( [x]_+ \) is the function returning the maximum value between 0 and \( x \). \( \mathcal{T} \) is the set of triples observed in the KG and \( \mathcal{T}^r \) is the set of negative samples obtained by random negative sampling. \( \mathcal{S} \) is the set of positive relations that is associated with relation \( r \) by length-1 rules and \( \mathcal{S}' \) is the set of negative relations beyond \( \mathcal{S} \) and relation \( r \).

We employ the mini-batch Stochastic Gradient Descent (SGD) algorithm to optimize the joint loss function for learning entity and relation embeddings. The entity and relation embeddings are initialized randomly and constrained to be unit vectors by the additional regularization term with L2 norm. With a collection of learned entity and relation embeddings, we can conduct KG inference in the way of information retrieval. Furthermore, relation embeddings can be utilized in the following rule learning module.

### 3.3 Embedding-Based Rule Learning

Remarkably, a path can naturally represent the body of a CP rule. Motivated by this observation, we firstly reuse the paths extracted in section 3.2 and regard these paths as candidate CP rules, which could improve the efficiency of the rule learning. On account of an entity pair \((h, t)\) connected by a path \( p = \{r_1(h, e_1), r_2(e_1, e_2), \ldots, r_n(e_{n-1}, t)\} \), we can deduce a CP rule as \( r(x, y) \Leftarrow r_1(x, z_1) \land r_2(z_1, z_2) \land \cdots \land r_n(z_{n-1}, y) \) if there is a relation \( r \) connecting the entity pair \((h, t)\), where \( x, y \) and \( z_i (i = 1, \ldots, n - 1) \) are the variables in the rule, and \( r, r_i \) are the relations.

We develop a novel and effective coarse-to-fine grained rule pruning strategy to evaluate and pick out the high-quality rules. At the coarse-grained evaluation, we propose an embedding-based score function to measure the plausibility of the candidate rules. The score function consists of two components: Embedding-based Semantic Relevance and Concept-based Co-occurrence, which respectively represent the global and local plausibility of a rule.

**Embedding-based Semantic Relevance.** Intuitively, a candidate rule is plausible if a path \( p \) corresponding to the rule body is similar to the relation \( r \) corresponding to the rule head. Meanwhile, at least an entity pair is connected by both the path \( p \) and the relation \( r \). It is noteworthy that we focus on length-2 paths and length-2 CP rules for the trade-off of efficiency and performance in both KG inference and rule learning. Based on the KG embeddings learned in our KGE module, we can measure the semantic relevance between the body (path) and the head (relation) of a candidate rule by the score function as

\[
E_{sr}(p, r) = \exp(-\|r - p\|) \tag{12}
\]

where \( p \) denotes the embedding of the path \( p \).

The embedding-based semantic relevance indicates a global evaluation of measuring the plausibility of a candidate CP rule. On the other hand, not all the paths that satisfy the global relevance can form a high-quality CP rule. Therefore, we propose a concept-based co-occurrence strategy to measure the local relevance of the shared neighbor arguments in a CP rule. For instance, on account of a CP rule \( \text{Nativity}(x, y) \Leftarrow \text{BornIn}(x, z) \land \text{LocatedIn}(z, y) \), the groundings of the atom \( \text{BornIn}(x, z) \) are all the triples containing the relation \( \text{BornIn} \). The concepts corresponding to the tail argument of relation \( \text{BornIn} \) and the head argument of relation \( \text{LocatedIn} \) should share the concept \( Location \). The neighbor arguments in a high-quality CP rule are expected to share as many same concepts as possible.

**Concept-based Co-occurrences.** Considering the concepts are far less than the entities, we encode each concept as a one-hot representation to maintain the precise semantic feature of each concept. The concept embedding of the head or tail argument of an atom can be respectively obtained by averaging all the one-hot representations of this argument’s concepts, which can be formalized as

\[
AC_h(r) = \frac{1}{||C_h(r)||} \sum_{c \in C_h(r)} OH(c) \tag{13}
\]

\[
AC_t(r) = \frac{1}{||C_t(r)||} \sum_{c \in C_t(r)} OH(c) \tag{14}
\]
where $AC_h(r)$ and $AC_t(r)$ are the concept embeddings in the head and tail position of arguments as to an atom containing the relation $r$. $C_h(r)$ and $C_t(r)$ are the concept sets respectively in the head and tail position of arguments, which are composed of the unique head and tail concepts corresponding to all the triples containing $r$. $OH(c)$ denotes the one-hot representation of the concept $c$.

Specific to a CP rule in the form of $r(x, y) \leq r_1(x, z_1) \land \cdots \land r_n(z_{n-1}, y)$, three types of co-occurrence score functions are designed according to the different positions of the overlapped arguments as followings:

- \[ E_{co}^h(r, r_1) = \text{sim}(AC_h(r), AC_h(r_1)) \]
- \[ E_{co}^l(r, r_n) = \text{sim}(AC_l(r), AC_l(r_n)) \]
- \[ E_{co}^{nt}(r, r_{i+1}) = \text{sim}(AC_l(r_i), AC_h(r_{i+1}))(i=1, \ldots, n-1) \]

where $E_{co}^h(r, r_1)$ denotes the co-occurrence similarity of the head arguments between the rule head and the first atom in the rule body, $E_{co}^l(r, r_n)$ indicates the co-occurrence similarity of the tail arguments between the rule head and the last atom in the rule body. $E_{co}^{nt}(r, r_{i+1})$ represents the co-occurrence similarity between the tail argument of the former atom with relation $r_i$ and the head argument of the latter atom with relation $r_{i+1}$ both in the rule body. $\text{sim}(p, q) = p \cdot q$ represent the cosine distance function for calculating the similarity of two vectors $p$ and $q$.

Then, the whole co-occurrence score function can be achieved by composing all the scores in Eqs. \[15\] \[17\] as

- \[ E_{co}(r, p) = E_{co}^h(r, r_1) + E_{co}^l(r, r_n) + \sum_{i=1}^{n-1} E_{co}^{nt}(r, r_{i+1}) \]

Consequently, the score functions associated with the Embedding-based Semantic Relevance and the Concept-based Co-occurrences can be combined as the overall score function for coarse-grained evaluation of candidate rules:

- \[ E_{cg} = E_{co}(r, p) + \beta E_{co}(r, p) \]

where $\beta$ is the weight of the co-occurrence score. We define the threshold of the coarse-grained score and filter out the candidate rules with the score below the threshold. Afterward, we employ two frequently-used quality criteria at the fine-grained evaluation stage, namely standard confidence and head coverage defined in Eqs. \[24\] to pick out the high-quality rules from the filtered candidate CP rules more precisely. In terms of the two quality criteria, we set the corresponding thresholds on standard confidence and head coverage. Only the candidate rules that satisfy the thresholds of both standard confidence and head coverage are selected as the high-quality rules for updating the rule set.

### 4 Experiments

#### 4.1 Experimental Setup

**Datasets.** To conduct the experiments of KG inference namely link prediction, four real-world datasets containing ontological concepts are employed, including FB15K (Bordes et al. 2013), FB15K237 (Toutanova and Chen 2015), NELL-995 (Xiong et al. 2017) and DBpedia-242. Particularly, DBpedia-242 is generated from the commonly-used KG DBpedia (Lehmann et al. 2015) to ensure each entity in the dataset has a concept. The statistics of the experimental datasets are listed in Table 1.

| Dataset         | #Relation | #Entity | #Concept | #Train  | #Valid | #Test  |
|-----------------|-----------|---------|----------|---------|--------|--------|
| FB15K           | 1,345     | 14,951  | 89       | 483,142 | 50,000 | 59,071 |
| FB15K237        | 237       | 14,505  | 89       | 272,115 | 17,535 | 20,466 |
| NELL-995        | 200       | 75,492  | 270      | 123,370 | 15,000 | 15,838 |
| DBpedia-242     | 298       | 99,744  | 242      | 592,654 | 35,851 | 30,000 |

**Baselines.** We compare our model EngineKGI with two categories of baselines:

- (1) The traditional KGE models depending on triple facts: TransE (Bordes et al. 2013), ComplEx (Trouillon et al. 2016), RotatE (Sun et al. 2019), QuatE (Zhang et al. 2019b), HAKE (Zhang et al. 2020) and DualE (Cao et al. 2021).
- (2) The previous KGE models using paths or rules: the path-based model PTransE (Lin et al. 2015a), the rule learning-based model RPIE (Niu et al. 2020b) and the model combining rules and KG embeddings iterE (Zhang et al. 2019c). Note that RLvLR (Omran, Wang, and Wang 2019) and DRUM (Sadeghian et al. 2019) are not utilized as baseline since they infer tail entities rather than head entities and most of the KGE approaches outperform them in both performance and efficiency. The evaluation results of the baseline models are obtained by employing the open-source codes on Github with the suggested hyper-parameters.

**Training Details.** We implement our model in C++ and on Intel i9-9900 CPU with memory of 64G. For the fair comparison, the embedding dimension of all the models is fixed at 100, the batch size is set to 1024 and the number of negative samples is set to 10. Specific to our model, during each iteration, the maximum training epoch is set to 1000, the standard confidence and the head coverage are respectively fixed to 0.7 and 0.1 for all the datasets. The entity and relation embeddings are initialized randomly. We employ grid search for selecting the best hyper-parameters on the validation dataset. The learning rate $l$ is tuned in \{0.001, 0.005, 0.01, 0.02, 0.05\}, the margins $\gamma_1, \gamma_2, \gamma_3$ are searched in \{1.0, 1.5, 3.0, 5.0\}, the weights $\alpha_1, \alpha_2, \beta$ are tuned among \{0.5, 1.0, 5.0\}, the score threshold $st$ for coarse-grained evaluation of rules is selected from \{1.0, 5.0, 10.0\}.

**Evaluation Metrics.** We test the performance following the evaluation setup in previous KGE models. Take head entity prediction for instance, we fill the missing head entity with each entity $e$ in the entity set $E$, and score a candidate triple $(e, r, t)$ according to the energy function as follows:

- \[ E_{o}(e, r, t, P) = E_{1}(e, r, t) + \alpha_{1}E_{o}(e, r, t, P) \]

...
Table 2: Link prediction results on four datasets. **Bold** numbers are the best results, and the second best is underlined.

| Models | MR | MRR | Hits@10 | Hits@3 | Hits@1 | MR | MRR | Hits@10 | Hits@3 | Hits@1 |
|--------|----|-----|---------|--------|--------|----|-----|---------|--------|--------|
| TransE | Bordes et al. 2013 | 117 | 0.534 | 0.775 | 0.646 | 0.386 | 228 | 0.289 | 0.478 | 0.326 | 0.193 |
| ComplEx | Trombition et al. 2016 | 197 | 0.346 | 0.593 | 0.405 | 0.221 | 507 | 0.236 | 0.406 | 0.263 | 0.150 |
| RotatE | Sun et al. 2019 | 39 | 0.612 | 0.816 | 0.698 | 0.488 | 168 | 0.317 | 0.553 | 0.375 | 0.231 |
| HAKE | Zhang et al. 2020 | 42 | 0.678 | 0.839 | 0.761 | 0.570 | 183 | 0.344 | 0.542 | 0.382 | 0.246 |
| DualE | Cao et al. 2021 | 43 | 0.759 | 0.882 | 0.820 | 0.681 | 202 | 0.332 | 0.522 | 0.367 | 0.238 |
| PTransE | Lan et al. 2015a | 52 | 0.679 | 0.834 | 0.819 | 0.693 | 773 | 0.312 | 0.495 | 0.344 | 0.222 |
| RPJE | Niu et al. 2020 | 40 | 0.811 | 0.808 | 0.832 | 0.762 | 207 | 0.443 | 0.579 | 0.479 | 0.374 |
| IterE | Zhang et al. 2019b | 85 | 0.577 | 0.807 | 0.663 | 0.451 | 463 | 0.210 | 0.355 | 0.227 | 0.139 |
| EngineKGI (Ours) | **20** | **0.854** | **0.933** | **0.885** | **0.810** | **118** | **0.555** | **0.707** | **0.590** | **0.479** |

Table 3: Link prediction results on FB15K and FB15K237 on various relation properties (Hits@10).

| Models | MR | MRR | Hits@10 | Hits@3 | Hits@1 | MR | MRR | Hits@10 | Hits@3 | Hits@1 |
|--------|----|-----|---------|--------|--------|----|-----|---------|--------|--------|
| TransE | Bordes et al. 2013 | 1996 | 0.256 | 0.539 | 0.395 | 0.075 | 8650 | 0.167 | 0.354 | 0.219 | 0.068 |
| ComplEx | Trombition et al. 2016 | 3839 | 0.196 | 0.387 | 0.230 | 0.104 | 11772 | 0.169 | 0.298 | 0.185 | 0.106 |
| RotatE | Sun et al. 2019 | 1323 | 0.308 | 0.594 | 0.422 | 0.143 | 9620 | 0.292 | 0.444 | 0.325 | 0.216 |
| HAKE | Zhang et al. 2020 | 1522 | 0.379 | 0.551 | 0.432 | 0.283 | 13211 | 0.245 | 0.370 | 0.283 | 0.175 |
| DualE | Cao et al. 2021 | 1363 | 0.360 | 0.592 | 0.439 | 0.232 | 11529 | 0.292 | 0.447 | 0.329 | 0.214 |
| PTransE | Lan et al. 2015a | 2170 | 0.436 | 0.510 | 0.469 | 0.393 | 6207 | 0.318 | 0.444 | 0.349 | 0.252 |
| RPJE | Niu et al. 2020 | 1770 | 0.521 | 0.576 | 0.542 | 0.487 | 6201 | 0.360 | 0.496 | 0.401 | 0.288 |
| IterE | Zhang et al. 2019b | 5016 | 0.190 | 0.326 | 0.215 | 0.129 | 12998 | 0.233 | 0.327 | 0.246 | 0.185 |
| EngineKGI (Ours) | **1275** | **0.523** | **0.647** | **0.551** | **0.501** | **5243** | **0.454** | **0.506** | **0.407** | **0.293** |

4.2 Results of Link Prediction

The evaluation results of link prediction are reported in Table 2. We analyze the results as follows: Firstly, our model EngineKGI significantly and consistently outperforms all the state-of-the-art baselines on all the datasets and all the metrics. Compared to the two types of best-performing models QuaE and RPJE, EngineKGI achieves performance gains of 30.9%/70.9%/28.6%/57.2% compared to RotatE and 24.3%/34.7%/11.2%/10.3% against RPJE on datasets FB15K/FB15K237/DBpedia-242/NELL-995. Particularly, on FB15K and FB15K237, the difference between the best performing baseline RPJE and our developed model is statistically significant under the paired at the 99% significance level. Secondly, our model achieves better performance than the traditional models that utilize triples alone, indicating that EngineKGI is capable of taking advantage of extra knowledge including rules and paths as well as concepts, which all benefit to improving the performance of the whole model. Thirdly, EngineKGI further beats IterE, illustrating the superiority of exploiting rules and paths for KG inference in a joint logic and data-driven fashion.

4.3 Evaluation on Various Relation Properties

The relations can be classified into four categories: One-to-One (1-1), One-to-Many (1-N), Many-to-One (N-1), and Many-to-Many (N-N). We select some well-performing models observed in Table 2 as the baselines in this section. Table 3 exhibits that EngineKGI achieves more performance gains on complex relations compared to other baselines. More interestingly, specific to the most challenging tasks (highlighted) namely predicting head entities on N-I relations and tail entities on I-N relations, our model consistently and significantly outperforms the outstanding baselines RotatE and RPJE by achieving the performance improvements: 38.7%/47.5% on FB15K and 238.2%/185.0% on FB15K237 compared to RotatE while 18.6%/7.75% on FB15K and 249.4%/102.5% on FB15K237 compared to RPJE. These results all demonstrate that our approach introducing valuable semantics derived from paths and generated rules benefit KG inference on complex relations.

4.4 Performance Evaluation Over Iterations

We evaluate the performance of our rule learning module on learning time and rules’ amount compared to the excellent rule learning tool AMIE+. For generating high-quality rules, our model takes 6.29s/2.26s/1.55s/10.50s in an iteration on FB15K237 compared to RotatE while 18.6%/7.75% on FB15K and 249.4%/102.5% on FB15K237 compared to RPJE. These results all demonstrate that our approach introducing valuable semantics derived from paths and generated rules benefit KG inference on complex relations.
initial iteration. Thus, we can discover that the quantity of rules generated in the first iteration and the third iteration is twice and three times the number of rules obtained by AMIE+, respectively.

More specifically, Figure 3 exhibits the number of rules and Figure 4 indicates the performance curves on the four datasets over iterations. Notably, the number of rules and the performance continue to grow as the iteration goes on and converge after three iterations on all the datasets. These results illustrate that: (1) Rule learning and KGE modules in our model indeed complement each other and benefit in not only producing more high-quality rules but also obtaining better KG embeddings. (2) More rules are effective to improve the performance of KG inference. (3) The iterative process will gradually converge along with the rule learning.

4.5 Ablation Study

To evaluate each contribution in our whole model EngineKGI, we observe the performance on FB15K237 as to the five different ablated settings: (1) Omitting paths (-Path). (2) Omitting rules (-Rule). (3) Omitting concepts (-Concept) by removing concept-based co-occurrences in rule learning. (4) Replacing rule mining tool AMIE+ with AnyBurl (Mellie et al. 2019) for obtaining the seed rules (+AnyBurl). (5) Employing only half of the seed rules and without iteration (-HalfRule). Figure 5 shows that the performance of our whole model is better than that of all the ablated models except for “+AnyBurl”, demonstrating that all the components in our designed model are effective and our model is free of any rule mining tool for obtaining the seed rules. Besides, removing paths and rules both have significant impacts on the performance, which suggests the paths and rules in our model both play a vital role in KG inference.

4.6 Case Study

As shown in Figure 6, although the head entity Jonathan and the candidate tail entity York are not linked by any direct relation in the KG, there is an explicit path between them. This path can be represented as the relation PersonBornInCity deduced by a matched CP rule. The path and the rule together boost the score of the correct candidate tail entity calculated by Eq. 20 and especially provide the interpretable explanation to help increase the user’s credibility of the predicted result.

5 Conclusion and Future Work

In this paper, we develop a novel closed-loop KG inference model EngineKGI by jointly learning rules and KG embeddings while taking advantage of additional information, including paths, rules and concepts. In the KG embedding module, both rules and paths are introduced to enhance the semantic associations and interpretability for learning the entity and relation embeddings. In the rule learning module, paths and KG embeddings together with entity concepts are leveraged in the designed rule pruning strategy to generate high-quality rules efficiently and effectively. Extensive experimental results on four datasets illustrate the superiority and effectiveness of our approach compared to other state-of-the-art models. In the future, we will investigate combining other semantics such as contextual descriptions of entities, and attempt to apply our model in dynamic KGs.
References

Bollacker, K.; Gottlob, G.; and Flesca, S. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In KDD, 1247–1250.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In NIPS, 2787–2795.

Cao, Z.; Xu, Q.; Yang, Z.; Cao, X.; and Huang, Q. 2021. Dual Quaternion Knowledge Graph Embeddings. In AAAI, 6894–6902.

Das, R.; Neelakantan, A.; Belanger, D.; and McCallum, A. 2017. Chains of Reasoning over Entities, Relations, and Text using Recurrent Neural Networks. In EACL, 132–141.

Galárraga, L.; Teflioudi, C.; Hose, K.; and Suchanek, F. 2015. Fast rule mining in ontological knowledge bases with AMIE+. The VLDB Journal, 24(6): 707–730.

Guo, S.; Wang, Q.; Wang, L.; Wang, B.; and Guo, L. 2018. Knowledge Graph Embedding with Iterative Guidance from Soft Rules. In AAAI, 4816–4823.

Hao, J.; Chen, M.; Yu, W.; Sun, Y.; and Wang, W. 2019. Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts. In KDD, 1709–1719.

Ho, V. T.; Stepanova, D.; Gad-Elrab, M. H.; Kharlamov, E.; and Weikum, G. 2018. Rule Learning from Knowledge Graphs Guided by Embedding Models. In ISWC 2018, 72–90.

Huang, X.; Zhang, J.; Li, D.; and Li, P. 2019. Knowledge Graph Embedding Based Question Answering. In WSDM, 105–113.

Ji, G.; He, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In ACL-IJCNLP, 687–696.

Jia, Y.; Wang, Y.; Jin, X.; and Cheng, X. 2018. Path-specific knowledge graph embedding. Knowledge-Based Systems, 151(2018): 37–44.

Lao, N.; Mitchell, T.; and Cohen, W. W. 2011. Random walk inference and learning in a large scale knowledge base. In EMNLP 2011, 529–539.

Lehmman, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; and Bizer, C. 2015. DBpedia-A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. Semantic Web, 6(2): 167–195.

Lin, Y.; Liu, Z.; Luan, H.; Sun, M.; Rao, S.; and Liu, S. 2015a. Modeling relation paths for representation learning of knowledge bases. In EMNLP, 705–714.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015b. Learning entity and relation embeddings for knowledge graph completion. In AAAI, 2181–2187.

Meilicke, C.; Chekol, M. W.; Ruffinelli, D.; and Stuckenschmidt, H. 2019. Anytime Bottom-Up Rule Learning for Knowledge Graph Completion. In IJCAI, 3137–3143.

Mitchell, T.; Cohen, W.; Hruschka, E.; and et al. 2018. Never-ending learning. Communications of the ACM, 61(5): 103–115.

Neelakantan, A.; Roth, B.; and McCallum, A. 2015. Compositional Vector Space Models for Knowledge Base Completion. In ACL, 156–166.

Niu, G.; Li, B.; Zhang, Y.; Pu, S.; and Li, J. 2020a. AutoETER: Automated Entity Type Representation for Knowledge Graph Embedding. In EMNLP Findings, 1172–1181.

Niu, G.; Zhang, Y.; Li, B.; Cui, P.; Liu, S.; Li, J.; and Zhang, X. 2020b. Rule-Guided Compositional Representation Learning on Knowledge Graphs. In AAAI, 2950–2958.

Omran, P. G.; Wang, K.; and Wang, Z. 2019. An embedding-based approach to rule learning in knowledge graphs. In IEEE Transactions on Knowledge and Data Engineering, 1–12.

Sadeghian, A.; Armandpour, M.; Ding, P.; and Wang, D. Z. 2019. DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs. In NIPS.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. RotateE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In ICLR.

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

Trouillon, T.; Welbl, J.; Riedel, S.; Eric Gaussier; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In ICML, 2071–2080.

Wang, S.; Wei, X.; Nogueira dos Santos, C. N.; Wang, Z.; Nallapati, R.; Arnold, A.; Xiang, B.; Yu, P. S.; and Cruz, I. F. 2021. Mixed-Curvature Multi-Relational Graph Neural Network for Knowledge Graph Completion. In Proceedings of the Web Conference 2021, 1761–1771.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In AAAI, 1112–1119.

Xie, R.; Liu, Z.; and Sun, M. 2016. Representation learning of knowledge graphs with hierarchical types. In IJCAI, 2965–2971.

Xie, Z.; Zhou, G.; Liu, J.; and Huang, J. X. 2020. ReInceptionE: Relation-Aware Inception Network with Joint Local-Global Structural Information for Knowledge Graph Embedding. In ACL, 5929–5939.

Xiong, W.; Hoang, T.; ; and Wang, W. Y. 2017. DeepPath: A reinforcement learning method for knowledge graph reasoning. In EMNLP, 564–573.

Zhang, N.; Deng, S.; Sun, Z.; Wang, G.; Chen, X.; Zhang, W.; and Chen, H. 2019a. Long-tail Relation Extraction via Knowledge Graph Embeddings and Graph Convolution Networks. In NAACL, 3016–3025.

Zhang, S.; Tay, Y.; Yao, L.; and Liu, Q. 2019b. Quaternion Knowledge Graph Embeddings. In NeurIPS, 2731–2741.

Zhang, W.; Paudel, B.; Wang, L.; Chen, J.; Zhu, H.; Zhang, W.; Bernstein, A.; and Chen, H. 2019c. Iteratively Learning Embeddings and Rules for Knowledge Graph Reasoning. In WWW, 2366–2377.
Zhang, Z.; Cai, J.; Zhang, Y.; and Wang, J. 2020. Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction. In AAAI, 3065–3072.

Zhou, H.; Young, T.; Huang, M.; Zhao, H.; and Zhu, X. 2018. Commonsense Knowledge Aware Conversation Generation with Graph Attention. In IJCAI, 4623–4629.